# 摘 要

VP9是Google推行的新一代开源视频编码标准，在保留了传统的混合视频编码框架的同时，采纳了大量新技术，让VP9在同等画质下码率比VP8降低50%，但也带来了巨大的编码复杂度。针对该问题，本文研究了VP9视频编码标准的帧内预测算法，期望降低其编码的复杂度，使该标准更适应实时性要求较高的业务。

本文针对VP9视频编码的帧内预测过程，提出了基于块划分的帧内预测快速算法以及基于模式选择的帧内预测快速算法。

前者是根据当前块的平坦度直接确定某些块的最优划分方式或者跳过某些不必要的划分方式；后者则是通过利用预测模式的方向相关性减少候选预测模式，并对预测过程中时间复杂度高的模块进行优化，减少其编码时间。

实验结果表明，所提出的两个算法使得VP9编码性能下降很小的同时，其帧内编码时间减少68%左右。

关键词：**VP9** 视频编码 帧内预测 快速算法

# **ABSTRACT**

VP9 is the next generation video coding standard that developed by Google. It still utilizes the traditional hybrid video coding framework and also adopts the newest video coding techniques in recent years. As a result, the coding performance of VP9 has a significant increment relative to the previous video coding standard VP8，and it achieves an average bit rate reduction of 50% compared with the VP8 coder while keeping the same perceptual video quality. However, the improvement of compression ratio sacrifices the coding speed at the same time. Based on the problem, this paper mainly focuses on the fast algorithms for intra prediction with expecting to reduce the coding time and thus makes VP9 standard more suitable for real time services.

Based on the key technologies of intra prediction in VP9, two fast algorithms of intra prediction are then proposed for VP9, they are the fast algorithm based on block partition and the fast algorithm based on mode selection.

The main idea of the former algorithm is to determine the optimal partition directly or skip some unnecessary partition based the smoothness of blocks. While the other one based on mode selection utilizes the texture similarity in direction of blocks and reduces the coding time of the most complex modules that make up intra prediction.

Experimental results show that the proposed fast algorithms can reduce the coding time of intra prediction by 68% in average with a negligible degradation on coding performance.

**Key words: VP9, Video coding, Intra prediction, Fast algorithms**

# 目 录

**第一章 绪论 1**

1.1论文研究背景及意义 1

1.2论文的主要内容与章节安排 2

**第二章 VP9视频编码技术 5**

2.1 VP9视频编码器特点 5

2.2 VP9视频编码器框架 6

2.3 编码结构 7

2.3.1 编码块 8

2.3.2 预测块 8

2.3.3变换块 9

2.5 VP9预测编码技术 9

2.5.1 VP9帧内预测技术 13

2.5.2 VP9帧间预测技术 10

2.6已有帧内预测快速算法研究 15

2.6.1基于块划分的快速算法研究 15

2.6.2帧内模式快速选择算法研究 17

**第三章 基于块划分的帧内预测快速算法 19**

3.1 引言 19

3.2 VP9帧内预测的编码块划分过程 19

3.3 VP9块划分快速算法 20

3.3.1编码块划分快速算法 20

3.3.2预测块划分快速算法 25

3.4算法结果与分析 27

3.5小结 28

**第四章 基于模式选择的VP9帧内预测快速算法 31**

4.1 引言 31

4.2 VP9帧内预测模式选择过程 31

4.3 帧内模式选择各模块的时间复杂度分析 32

4.4 帧内预测模式快速选择算法 33

4.4.1预测模式初选择 33

4.4.2预测模式再选择 35

4.4.3预测过程中的变换块尺寸选择 36

4.5实验结果与分析 37

4.6本章小结 38

**第五章 总结与展望 39**

**致谢 41**

**参考文献 43**

# 第一章 绪 论

## 1.1论文研究背景及意义

在互联网中，以视频内容作为媒介的流量已经占据了整个网络流量的80%以上[4]。特别是智能手机屏幕分辨率的快速提高，高分辨率、高质量的视频必须要更大的视频码流传输。目前高清电视和视频已经普及，因此人们对高清视频的要求和网络带宽容量存储之间的矛盾也日益突出。

大量的视频应用带来的多种多样的视频编码方法，为了能够让大范围的通用性地解码视频码流，国际标准组织也开始对视频编码建立国际标准。在这段历程中各国际标准组织和一些公司发布了许多编解码标准，其发展历史脉络如图1.1所示。



图1.1 视频编码技术发展历程

ITU-T主要关注视频通信领域，制定了包括H.263、H.264、H.265等标准，这些标准被广泛应用于网络视频传输业务，比如可视电话。MPEG（Moving Picture Experts Group，MPEG）动态图像专家组主要制定了MPEG系列视频编码标准。目前主流的视频编解码技术H.264包含专利，而且在硬件上使用需要取得授权，而Google则一直致力于发展开放和免费的网络视频编码标准，并且在2013年推出了新一代视频编码标准——VP9。

移动终端的出现极大地改变了我们认知生活和获得信息的方式，4G和WIFI的大面积普及也为人们获取高清视频奠定了基础。VP9是在2013年6月正式确定的编码标准，实际编码的效率与H.265/HEVC[2]相接近，比 H.264/AVC[3]优势很大。谷歌为了让VP9成为下一代的超高清视频编码主流标准之一，就在Chrome浏览器中和YouTube全面支持VP9视频编码技术。并且得到了如松下、三星、英特尔、ARM等硬件厂商的支持。

根据统计数据显示，目前访问YouTube网站的用户已经有120多亿人次，也就是说全球有约二分之一网民都在访问YouTube，因此VP9拥有庞大的用户，而且预计VP9硬件解码也将大量用于移动设备。VP9为了能够高校编码而采用了很多新的编码技术，其编码效率比VP8高了将近一倍，但同时也带来了极大的计算复杂度。但对于视频会议、视频电话和视频监控等实时应用，编码速度成为最重要的影响因素，所以研究如何提高编码速度也是相当重要和迫切的。在整个编码过程中，预测模块是最消耗时间的，所以预测模块是重要研究内容。本文主要针对帧内预测来研究VP9的快速算法，期望在保证视频质量下降很小的同时减少更多的编码时间。

## 1.2论文的主要内容与章节安排

视频是由一帧一帧的图像构成，相邻帧间一般区别很小，所以会存在很大的冗余，我们称之为时间冗余。当前帧中也会存在很大的相似性，我们成为空间冗余。因此我们可以利用相邻图像之间的相关性和一幅图像中相邻像素之间的相关性除去视频内容的大量冗余信息。视频压缩的基本过程是先通过帧间编码去除时域和空域冗余冗余信息，再使用熵编码对残差进行压缩，得到视频的编码比特流进行传输。

本文先对VP9的关键性技术进行研究，通过统计分析和参考来寻找降低其编码复杂度的优化算法。本文主要研究了帧内预测过程中的两个关键过程：块划分和帧内模式选择。针对这两个过程提出了对应的快速算法：基于块划分的帧内快速算法和基于模式选择的帧内快速算法。

首先，本文介绍了预测中块划分的详细过程，然后分析了原本算法的时间复杂度，根据统计结果确定块划分优化的整体思路：直接确定一些块的划分深度或者跳过一些影响不大的划分。在整个算法实现中，先分析了影响块划分的两个因素，分别是空间复杂度和量化参数。然后，统计分析空间复杂度和量化参数是如何影响编码块划分的。最后，根据实验和统计结果提出块划分的快速算法。从实验结果可以看出，该算法可以对具有不同纹理特征的不同序列，在各个量化参数下都能大幅度减少编码时间，并且视频质量几乎没有损失。

接着，介绍了帧内预测中模式选择的具体过程并分析了原有算法的复杂度，并利用分析结果来提出优化算法。通过大量测试和实验，得到影响帧内模式选择的各个因素，本文就其中的两个因素提出优化思路：利用预测模式角度相关性和减少帧内复杂度最高的变换单元。根据实验结果和影响因素，本文提出了基于模式角度的快速算法。算法的总体思路是利用角度相关性减少预测模式和优化变换块的复杂度。实验结果显示算法可以在保证视频质量几乎不变的情况下减少大量编码时间。

全文总共分为五章，各章的内容安排如下。

第一章：绪论，第一节主要介绍视频编码标准的历史、VP9产生的背景、VP9的优势和劣势，以及分析研究快速算法的必要性。第二节分别介绍论文的研究内容以及章节安排。

第二章：先是介绍了VP系列视频编码的发展历程和VP9视频编码器的特点，

接着介绍了VP9编码中的块划分技术和预测编码技术，最后总结分析了已经提出的帧内快速算法。

第三章：是针对块划分的帧内快速算法。第一、二节简要分析了研究帧内块划分优化算法的必要性以及研究过程。第三节针对编码块和预测块的划分过程，进行了大量的测试和统计，分别提出了相应优化算法。第四节是该算法的具体实现以及对算法性能做的测试和分析。

第四章：描述了VP9帧内预测中模式选择过程的快速选择算法。第一、二节主要对研究帧内模式选择优化算法的必要性和研究过程进行了分析。第三节描述了VP9中帧内预测中模式选择的相关因素。第四节针对帧内模式选择过程，结合上一节的分析结果，提出了优化算法并对算法性能做了测试和分析。第五节是该算法的具体实现以及对算法性能做的测试和分析。

第五章：总结了毕设期间的工作和研究结果，并对VP9研究方向的发展进行了分析。

# 第二章 VP9视频编码技术

VP9是由Google推出的免费视频编码标准，在同等画质下码率比H.264降低50%，并支持更大的编码块、更多的色彩空间。VP9同视频编码标准领域的其他标准，如H.264编码器功能块的组成基本一致，不同的是各个功能块的实现细节。VP9编码器由基本的几个模块组成，分别为块划分、帧内预测、帧间预测、变换量化、熵编码以及环路滤波。

## 2.1 VP9视频编码器特点

高清和超高清视频业务的快速发展，使得大分辨率图像压缩成为难题。大分辨率图像的一个特点是平坦的区域更大，而传统的视频编码通常都以宏块(MacroBlocks)作为编码的基本单位，这样就不能利用大分辨率图像的特性来获得最好的压缩效果。因此在VP8 16x16宏块的基础上，VP9采用了同HEVC相似大块划分方式——超级块，其中超级块最大为64x64。超级块采用四叉树的编码结构来划分出更小的子编码块。这样可以更好地复用图像像素，使得编码压缩效率提升50%。

VP8的帧内预测只对4x4块的采用10种候选模式，而对大块只采用4种候选模式，VP9对所有大小的预测块都支持10种预测模式：分别为DC、True Motion（简称TM）和8种角度预测，其中8种角度预测的方向分别为45°、63°、117°、90°、135°、153°、180°、207°。

VP9帧间预测支持3种类型的参考帧。每个帧间预测块可以选择1个或2个参考帧进行运动估计。VP9的运动估计支持1/8像素精度，相对于VP8或H.264/AVC的1/4像素精度提升50%，因此预测也会更加地准确。VP9支持3种类型的插值滤波器，分别为常规滤波器、锐利滤波器、平滑滤波器。在VP8采用6阶插值滤波器的基础上，VP9也将插值滤波器提升到8阶。

VP9支持不同大小变换块包括32x32、16x16、8x8、4x4四种变换尺寸。并且VP9去掉了VP8中的WHT变换，加入了一种新的变换类型ADST(Asymmetric Discrete Sine Transform)。在帧内预测时，如果边界只有一个的参考像素可用时，则使用ADST变换的效果要优于DCT变换。VP9的量化参数取值范围比VP8大了一倍，VP8的量化参数取值范围为0~127，VP9的量化参数范围为0~255。

VP8的布尔编码器是利用8个bit的概率进行编码的，但VP9是进行在每帧的头部存储码元概率的更新信息，所以能够对编码器中已有的概率进行部分更新。因此VP9有一定的自适应性，提高了压缩效率。

为了使编码器更加高效，VP9采用了更加先进的分段(segmentation)技术。一帧图像中的每个16x16的编码块都有一个段标识(segment\_id)来区分所属的段。位于同一个段的块在参考帧、量化参数、滤波强度等参数上采用相同的处理。比如对一帧中的背景区域利用较小的量化参数就可以获得很好的编码质量，在编码后续帧的过程中背景区域可以通过帧间预测来进行更精确地预测，从而可以节约出更多的比特对前景区域进行编码。

VP9对无损压缩和有损压缩都支持，并且支持多核多线程的并行，利用并行技术可以显著地提高编码速率。VP9中每帧的头部都有并行模式的标识。VP9也和HEVC一样引入了tile的概念，通过把一帧图像分为不同的tile，可以让每个tile独立编解码，这样可以更好地实现并行。

## 2.2 VP9视频编码器框架

如图2.1和图2.2所示，VP9和H.264的编解码器的功能模块的基本一致，主要在各功能模块的实现上有所不同。为了适应视频内容的实时变化，例如有时空间纹理复杂，有时平坦区域很大，VP9采用了对应的自适应技术。由于Google设计VP9的初衷是为了适应互联网的视频业务，但互联网稳定性差，容易阻塞或信号衰落， 为了对抗信号多变的影响，需要编码器需要有相应的自适应技术。

如图2.1所示，VP9编码器采用的是混合编码框架，框架包括了帧内预测、帧间预测、变换、量化、熵编码和环路滤波等模块。输入的原始视频帧首先被划分为互不重叠的块，然后以块为单位被VP9编码器处理。每个块采用帧内预测或帧间预测进行编码，并且它们都是参考重构后的像素来进行预测，得到预测像素值（图2.1中的P）。若是帧内预测，其预测值是由当前块周围的重构像素预测得到的。如果是帧间预测，其预测值P是由前面重构帧（如图2.1中的）中经过运动搜索和运动补偿得到的。VP9参考帧只有3个，但每个参考帧的性质都不相同。这样既减小了处理器的压力，也降低了编码复杂度，提升了编码速度。而且多个参考帧可以保证预测过程的预测精度。



图2.1 VP9视频编码器框图

图2.2 VP9视频解码器框图

预测后将原始块的参考像素和预测块对应像素相减得到残差块，再对残差块进行变换，去除频域上的冗余信息。变换块经过量化后得到DCT系数，对DCT系数熵编码，得到编码比特。然后将解码所需的一些头信息（如划分、模式、量化参数、跳过标志位等）和熵编码压缩后的信息组成要传输视频码流。

为了提供进一步预测需要的参考帧，编码器需要具有重构帧的模块。重构图像的模块与解码器的功能几乎一致。重构过程为残差经过反量化和反变换后与预测值进行相加得到未滤波的重构帧（如图2.2中的）。为了提高参考帧的质量，需要进行环路滤波以去除方块效应，滤波后的图像即为重构帧（如图2.2中的），重构帧将在后续编码图像中作为参考帧。

## 2.3 编码结构

VP9将帧处理单元定义了编码块、预测块和变换块。并且三个单元相互自由、独立，让编码、预测和变化各个处理单元也更加灵活。所以VP9的编码结构能够更好地处理具有不同纹理特征和运动变化的视频序列。

### 2.3.1 编码块

为了更灵活、更有效使地表现视频内容，VP9采用了树形结构单元，对亮度和色度分别进行树形编码。VP9最大的编码单元尺寸可以为64x64，并且通过四叉树（如图2.3所示）的编码方式可以将超级宏块进一步划分。其中每个超级宏块可以分为4个32x32编码块，32x32的编码块可以再分为4个16x16的编码块，最小可以划分的尺寸为8x8。通过四叉树结构递归循环可以遍历每一个编码单元和模式，有效地提升了编码压缩效率。

VP9的这种表示方法比VP8中的宏块划分方法更灵活，也有更多的优势。

1、编码块的大小可以大于传统的宏块大小(16×16)。对于平坦区域，使用较大的编码单元进行编码能减少使用的比特数，提高编码效率。

2、合理地选择64x64块的大小和最佳划分方式，编码器的编码结构能够根据不同的应用需求、图片大小、以及图片内容获得很大程度的优化。



图2.3 编码块的循环四叉树分层结构

### 2.3.2 预测块

预测块规定了编码块的所有预测模式，所有预测处理和信息都在预测块中进行和保存。帧内预测、帧间预测、运动矢量预测以及帧间预测参考帧索引都在预测块中进行。预测块有4种划分形式： 2N×2N、2N×N、N×2N、N×N。如图2.4所示。其中2N×N、N×2N不能继续往下划分。



图2.4 VP9预测块的划分方式

### 2.3.3变换块

变换块是完成变换和量化的基本单元，VP9的变换块可支持大小为32×32至4×4的编码变换，实现了编码结构的灵活独立。图2.5给出了当前编码块为64x64下各个预测块中变换块的分割结构的一个划分图例。根据预测残差的局部变化特性，变换块可以自适应地选择最优尺寸和模式。大变换块能够更好地集中能量，小变换块能够保存更多的图像细节。变换块的这种独立的划分选择方式，让变换后的残差信息得到充分地压缩，以进一步提高编码增益。



图2.5 变换块的划分示意图

## 2.5 VP9预测编码技术

预测编码是利用信号之间存在的相关性，利用前面一个或多个信号预测下一个信号进行，然后对原始值和预测值的差进行编码。如果预测比较准确，误差就会接近于0。在一定精度条件的要求下，就能够用较少的比特表示和传输，以达到压缩数据的目的。视频中的一幅图像内邻近像素之间有着较强的空间相关性，相邻图像之间也有很强的时间相关性。如今大部分视频编码采用帧内预测和帧间结合的预测方式，去除更多的视频空域和时域的相关性，提高编码效率。

预测编码是视频编码中的核心技术之一，本小节将介绍VP9中的帧间和帧内预测编码技术。

### 2.5.1 VP9帧间预测技术

因为视频序列帧间有很强的时间相关性，所以帧间预测是利用时间的相关性，使用已编码的帧来预测当前帧，以去除时域产生的冗余。

帧间预测的两个主要因素：参考帧和运动矢量。参考帧用于复制像素到预测帧，并根据复制块与当前处理块的偏差形成运动矢量的已编码帧。

1、参考帧

VP9的帧间预测最多支持3个参考帧：Last\_frame、Golden\_frame和Altref \_frame。

Last\_frame 为当前帧在时间轴上的前一帧。

Golden\_frame 是指时间上任意远的一帧，用来存储和重建编码视频背景内容的更新信息。Golden\_frame的特点是质量较高，量化参数较小。

Altref\_frame同Golden\_frame相似，量化参数较小，质量较高，但该类型帧只能被参考不能被显示。Altref\_frame参考帧有两种，第一种Altref\_frame是通过普通编码方式得到的，只是其QP较小，但质量高于普通帧。第二种Altref\_frame它是由多个未编码帧的编码块组合成的。

2、预测运动矢量(PMV)

帧间的PMV有五种，分别为NEARESTMV、NEARMV、NEWMV、ZEROMV和SPLITMV，其中SPLITMV只适用于小于8x8的块，其他 MV适用于大于等于8x8的块。

每一种PMV，根据参考帧类型及参考帧数目的不同，可以继续分为五种情况。

1. NEARESTMV和NEARMV

根据参考帧类型及参考帧数目的不同，分为五种情况：一个参考帧：Last\_frame、Golden\_frame或Altref\_frame；两个参考帧：（Last\_frame,Altref\_frame）或（Golden\_frame,Altref\_frame）。确定参考帧后，当前编码帧间块有8种空域候选块和1种时域候选块可以选择。编码块空域候选块的位置和顺序随当前块尺寸不同会有所不同。图2.6给出了不同预测块的空域候位置，图中每一个候选块的坐标均为当前编码块的相对坐标。其中ABCDEFGH为当前块的空域候选块，加上一个前一帧中与当前块位置相同的时域候选块，一共9个候选块，尺寸均为8x8。



图2.6 空域候选块位置

①在选取当前块的PMV时，先遍历候选块的参考帧，若候选块的参考帧中与当前块的参考帧相同，则候选块的MV会被放入参考帧列表，当有两个满足条件的MV，就不再遍历。如果遍历所有的候选块都没有找到两个满足条件的MV，则直接将前两个候选块的MV放入参考帧列表。

因为一个五种参考帧组合方式，并且每一种参考帧组合都需要找到一个NEARESTMV和一个NEARMV，所以最终可能会得到5个NEARESTMV和5个NEARMV。

1. NEWMV

NEWMV是由NEARESTMV和NEARMV得到的，但其不是PMV，而是编码块的真实MV。

①从NEARESTMV和NEARMV中选取一个失真较小的MV，当作 NEWMV运动搜索的起点。

②搜索代价最小的MV即为NEWMV。

③NEWMV模式的PMV为NEARESTMV，其MVD为NEWMV与NEARESTMV的差值。

1. ZEROMV

ZEROMV即PMV为0。

(4)SPLITMV



图2.7 8x8块的划分示意图

SPLITMV只适用于小于8x8的块， 8x8块划分方式如图2.7所示。

3、 帧间预测模式

VP9帧间预测模式主要由参考帧和运动矢量确定，总共25种。

1. NEARESTMV:

RD\_NEARESTMV,LAST\_FRAME,NONE

RD\_NEARESTMV,GOLDEN\_FRAME,NONE

RD\_NEARESTMV,ALTREF\_FRAME,NONE

RD\_NEARESTMV,LAST\_FRAME,ALTREF\_FRAME

RD\_NEARESTMV,GOLDEN\_FRAME,ALTREF\_FRAME

2) NEARMV

RD\_NEARMV,LAST\_FRAME,NONE

RD\_NEARMV,GOLDEN\_FRAME,NONE

RD\_NEARMV,ALTREF\_FRAME,NONE

RD\_NEARMV,LAST\_FRAME,ALTREF\_FRAME

RD\_NEARMV,GOLDEN\_FRAME,ALTREF\_FRAME

3) NEWMV

RD\_NEWMV,LAST\_FRAME,NONE

RD\_NEWMV,GOLDEN\_FRAME,NONE

RD\_NEWMV,ALTREF\_FRAME,NONE

RD\_NEWMV,LAST\_FRAME,ALTREF\_FRAME

RD\_NEWMV,GOLDEN\_FRAME,ALTREF\_FRAME

4) ZEROMV

RD\_ZEROMV,LAST\_FRAME,NONE

RD\_ZEROMV,GOLDEN\_FRAME,NONE

RD\_ZEROMV,ALTREF\_FRAME,NONE

RD\_ZEROMV,LAST\_FRAME,ALTREF\_FRAME

RD\_ZEROMV,GOLDEN\_FRAME,ALTREF\_FRAME

5) SPLITMV

RD\_SPLITMV,LAST\_FRAME,NONE

RD\_SPLITMV,GOLDEN\_FRAME,NONE

RD\_SPLITMV,ALTREF\_FRAME,NONE

RD\_SPLITMV,LAST\_FRAME,ALTREF\_FRAME

RD\_SPLITMV,GOLDEN\_FRAME,ALTREF\_FRAME

### 2.5.2 VP9帧内预测技术

帧内预测编码是指利用一帧图像中的空间相关性，使用当前图像已编码的像素预测当前帧的像素，以达到去除视频空域冗余的目的，再对预测残差进行熵编码压缩。

VP9支持13种尺寸的预测块每种尺寸的预测块都有10种预测模式，包括DC、TM和8种角度模式，所有模式有对应的模式编码如表2.1所示，图2.8(b)中的箭头方向表示了各个预测模式的方向。每种预测模式都使用相同的模板，如图2.8所示。其中表示参考像素，表示当前块像素的预测值。



图2.8 VP9帧内预测模板

表2.1 帧内预测10种模式编号

|  |  |
| --- | --- |
| 帧内模式编号 | 帧内模式名称 |
| 0 | DC\_PRED |
| 1 | V\_PRED |
| 2 | H\_PRED |
| 3 | D45\_PRED |
| 4 | D135\_PRED |
| 5 | D117\_PRED |
| 6 | D153\_PRED |
| 7 | D207\_PRED |
| 8 | D63\_PRED |
| 9 | TM\_PRED |

各尺寸块预测像素的生成方式都相同。下面以4x4的预测块为例介绍每种预测模式下预测像素的生成过程。4×4亮度块的上方、左侧以及左上方像素A～M为已编码的重构像素，a～p为待预测像素，利用如表2.1的10种预测模式和像素A～M进行预测计算。

图2.8 4×4亮度预测模板和方向

具体的十种模式预测方式如图2.9所示。从图2.8，4×4亮度预测模式图可以看出，VP9的10种亮度预测模式里，模式0、模式2～9预测模式同H.264标准中的9种4×4亮度预测模式一致。



图2.9 4×4亮度块预测模式

## 2.6已有帧内预测快速算法研究

目前已经有很多帧内预测快速算法，大致可以划为基于块划分的快速算法和基于模式选择的快速算法。

### 2.6.1已有的块划分快速算法

图像的内容大多数是平坦的，对于平坦的块采用大块划分可以减少传输比特，而细节复杂的区域采用小块更能保留图像特征。已有的快速算法中主要分为以下几类：1）通过分析图像块的内容，直接减少预测块的数目；3）利用前面预测块划分下得到的数据，提前终止预测块的四叉树划分；2）利用相邻块之间的相关性来预测当前块的划分方式。

1、基于空间复杂度的快速块划分方法

文献[24]中，是根据预测块的空间复杂度来确定帧内预测块的划分方式。块的空间复杂度由当前块的AC系数确定，计算公式如下：

 (2-1)

其中，Y(i，j)表示宏块中(i，j)位置处的亮度值。如果variance大于门限值就定义为高复杂度块，否则定义为低复杂度块。如果宏块被判断为高空间复杂度，则只选择4x4和8x8的划分；如果宏块被判断为低空间复杂度，则只选择16x16和8x8的划分。

2、利用相邻块的信息预测当前块的划分

文中把HEVC中编码单元的大小用划分深度表示，64x64块对应深度为0，8x8对应深度为3。因为自然图像中相邻块的划分具有很强的相关性，在文献[26]中利用当前块左侧、左上侧、上侧和右上侧共4个相邻块的划分深度来预测当前块的划分深度。预测公式如下：

 (2-2)

其中N等于4表示0~3共4个划分深度，表示深度为i的划分下的权重，表示深度值。小于等于0.5时，当前块的划分深度取0；大于 0.5并小于等于1.5时，当前块的划分深度取1；大于 1.5并小于等于2.5时，当前块的划分深度取2；大于 2.5，当前块的划分深度取3。

3、提前终止的快速预测块划分算法

文献[27]中定义了一个称为平均像素代价（Average Pixel Cost，简称APC）的量。如果APC小于定义的门限值，就终止当前块的划分，减少了预测块的RDO次数，使得编码时间平均减少了31.3%。

Kim等人在文献[29]中利用编码单元的率失真代价设置门限。如果率失真代价小于设置的门限，就不再继续划分；如果率失真代价大于设置的门限，就继续划分。在码率基本不变下，编码时间减少了24%

### 2.6.2已有的帧内模式快速选择算法

1、减少RDO候选预测模式的快速算法

这类方法的基本思想是通过分析图像内容选择若干可能性较大的模式或丢弃可能性较小的模式，从减少的模式中选择出最优模式进行RDO，从而减少了帧内预测的时间。此类方法主要分为以下四类：

1. 基于变换域的快速算法

这类算法是利用变换系数的一些特征排除一些可能性小的预测模式，从而减少帧内预测次数。文献[16]中，苏睿等人对待预测的4x4块进行整数DCT变换，利用变换域的6个AC系数估计出预测块的大致纹理方向，然后使用估计出的预测方向减少候选模式数目。

1. 基于纹理分析的快速算法

文献[9]中利用sobel算子计算预测块中每个像素的梯度方向和幅值，选择与梯度方向最接近的预测模式作为像素的梯度方向，再累加整个预测块中梯度方向相同的像素的幅值得到当前块的边缘方向直方图，最后选择直方图中梯度方向幅值最大的几个模式当作候选预测模式。

1. 提前终止的快速算法

在预测过程中，对大多数的预测块而言，如果在前面的预测模式中已经确定了最佳的预测模式，就无需进行后面的预测了。因此可以设置相关门限，提前终止预测过程。在文献[16]中，作者先对宏块的最佳预测模式进行统计分析，然后根据分析结果在模式选择过程中采用提前终止策略。

1. 利用简化率失真代价函数的模式选择快速算法

简化率失真代价函数的快速算法主要是通过分析率失真代价函数的特点，采用新的算法简化率失真代价函数的计算或者代替率失真代价函数，达到减小降低帧内编码过程复杂度的目的。在文献[20]中，作者利用用SATD代替率失真函数，并结合相邻块的信息进行最佳模式的选择，取得了很好的效果。在文献[22]中作者使用哈达玛变换的率失真代价来近似代替率失真代价，来简化模式选择的计算。

# 第三章 基于块划分的帧内预测快速算法

## 3.1 引言

由第二章得知VP9拥有更多的块划分方式和预测模式，这让帧内编码的压缩效率提高了很多，但同时造成编码复杂度大幅度增加。本章将从块划分角度进行VP9帧内算法的优化。

## 3.2 VP9编码块划分过程

VP9支持的最大编码单元为64x64，块划分过程就是确定超级宏块的划分方式。由于遍历整个过程需要从64x64一直到8x8，而且每个编码块还需要进一步确定出最优的预测块和变换块，所以整个编码划分过程复杂度很高。其过程如下：

编码块一共有4种划分形式：PARTITION\_NONE、PARTITION\_SPLIT、PARTITION\_HORZ（水平），PARTITION\_VERT（垂直）。其中PARTITION\_SPLIT中的每个不小于8x8的子块可以进一步进行四叉树划分，如图3.1所示。

对当前每一个CU遍历4种划分模式，首先进行PARTITION\_NONE，直接以当前CU为单位进行预测模式选择。接下来如果当前块的尺寸不小于8x8就进行PARTITION\_SPLIT，每个子块再分别遍历四种划分模式，这是一个逐层递归的过程。在每一层结束SPLIT后，做PARTITION\_HORZ和PARTITION\_VERT，每个长方形子块分别进行预测模式选择。最后比较4种划分模式下的RDCost，选择出最佳划分模式，将相应划分的模式信息和残差系数进行打包，传给熵编码模块。

根据以上描述可知，一个最大编码单元需要遍历所有的编码块才能最优划分方式。该算法虽然可以得到全局最优的划分方式，达到最佳的编码性能，但是编码复杂度太高。因此我们可以将上面过程适当优化，直接确定某些编码块的划分方式或跳过某些划分，达到减少编码时间的目的。



图3.1 编码块划分过程

## 3.3 VP9块划分快速算法

### 3.3.1编码块划分快速算法

由上一小节指导整个编码块的划分复杂度很高，编码块需要从64x64到4x4块中的任何一个尺寸进行遍历，最终找到最优的划分方式。因此我们需要对划分过程中的每个影响因素进行分析，并根据实验和分析结果确定最优的优化思路和算法。

从实验结果我们发现如果根据图像内容选择跳过某些编码块的划分或某一步划分那么可以减少整个块划分的编码时间。通过实验我们可以确定划分和图像内容的关系。实验测试序列为BaksetballPass，图3.2和图3.3分别是量化参数分别为200和100对应图像的最优划分方式。

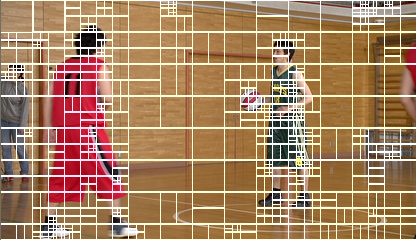


图3.2 QP=200时BasketballPass序列的图像划分方式

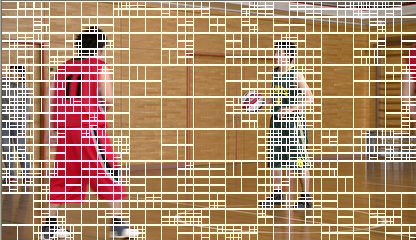


图3.3 QP= 100时BasketballPass序列的图像划分方式

从上面的实验结果可以发现，量化参数和图像内容的复杂情况都对图像的划分有所影响。

(a) 图像内容复杂度是确定编码块划分的决定性因素。通过图3.3我们可以发现平坦区域一般采用较大的划分，而图像内容复杂的区域划分都比较小。主要原因是平坦区域纹理一致性比较高，采用大块的划分可以准确地预测图像内容并且能够提高编码效率；而空间复杂度较高的区域需要更多的块去拟合纹理特征，这样才能保证视频质量。

(b) 量化参数是影响编码块划分另一个因素。由图3.3和3.2可以发现量化参数越大划分也越大，具体原因是因为量化参数大会使图像细节损失很多。所以大量化参数对视频图像内容的划分影响很大。

图像内容复杂度是确定编码块划分的决定性因素，所以需要通过一定的值来表示图像的空间复杂度。图像空间复杂度的表示方法有很多，比如利用灰度共生矩阵和通过灰度共生矩阵特征值来表示。本章中使用编码块中像素值的SSE来表示图像的空间复杂度。以HxW的矩形编码块为例，其图像的空间复杂度c可以由式3-1表示。

 (3-1)

式中表示图像中第i行、第j列的像素值，表示矩形编码块内像素值的平均值。

我们可以假设当前编码块的复杂度已经计算出来，并用C0表示。如果当前编码块没有划分到最大，就对当前编码块进行SPLIT，然后继续计算四个子编码块的空间复杂度，表示为C1、C2、C3和C4。表3.1是选择当前编码块NONE为最佳划分情况下的各个子编码块的空间复杂度C0、C1、C2、C3和C4。表3.2是编码块SPLIT为最佳划分情况下各编码块的空间复杂度，表示为C0、C1、C2、C3和C4。

表3.1 不划分情况下的空间复杂度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C0 | C1 | C2 | C3 | C4 |
| 3 | 3 | 3 | 3 | 3 |
| 3 | 2 | 3 | 2 | 2 |
| 4 | 3 | 2 | 3 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 3 | 4 | 2 | 3 | 2 |
| 3 | 3 | 2 | 3 | 2 |
| 3 | 2 | 3 | 2 | 3 |
| 3 | 2 | 3 | 2 | 2 |

表3.2 划分情况下的空间复杂度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C0 | C1 | C2 | C3 | C4 |
| 37 | 3 | 2 | 21 | 28 |
| 37 | 2 | 2 | 27 | 28 |
| 30 | 1 | 1 | 35 | 31 |
| 22 | 10 | 13 | 30 | 16 |
| 19 | 21 | 18 | 19 | 15 |
| 32 | 13 | 36 | 2 | 9 |
| 23 | 3 | 6 | 36 | 4 |
| 14 | 3 | 27 | 3 | 4 |

通过对比两个表格的数据可以发现，被划分的编码块中空间复杂度相较于不划分的编码块差别很大，复杂度较高与块划分有着很大的相关性。如何表示四个子块空间复杂度的差异是一个要解决的问题。衡量一个样本波动大小可以使用方差，方差越大，样本数据的波动就越大。因此可以用方差表示当前块空间复杂度。我们定义块空间复杂度的差异性为图像平坦度var，计算公式如3-2所示：

 (3-2)

其中表示亮度像素值，表示编码块的尺寸。

在此公式下，需要对平坦度确定门限。平坦度小于门限的定义为平坦，平坦度大于门限的定义为不平坦。由于方差和块大下、量化尺寸都有密切关系，因此主要从这两个方面考虑设定阈值。测试序列BasketballDrive\_1080P在QP=120，160，200时的平坦度分布。对比图3.4.和图3.5可以看出当QP=120时，64x64最佳划分为NONE（不划分）情况下，当前编码块的平坦度90%以上都在128以下；32x32最佳划分为当前编码块（NONE）情况下，当前编码块的平坦度90%以上都在192以下。对比图3.6和图3.7可以看出当QP=200时，64x64最佳划分为当前编码块（NONE）情况下，当前编码块的平坦度90%以上都在300以下；32x32最佳划分为当前编码块（NONE）情况下，当前编码块的平坦度90%以上都在400以下。 可以看出，随着块划分，var会越来越大；随QP增大，var也会增大。通过一系列的实验确定平坦度划分公式：

 (3-3)

其中bs表示当前块的尺寸，QP为量化参数，selfset为根据不同的分辨率序列设置初始值。

图 3.4 QP=120时最优划分方式为不划分时的平坦度分布

图 3.5 QP=120时最优划分方式为不划分时的平坦度分布

图 3.6 QP=200时最优划分方式为不划分时的平坦度分布

图 3.7 QP=200时最优划分方式为不划分时的平坦度分布

通过对对编码块划分的实验分析，提出一种对编码块划分的门限设置方法以进行划分地提前终止。

具体算法过程如下：

(1) 在当前编码块进行划分之前，首先进行空间复杂度的检测。计算当前编码块的平坦度，判断是否平坦。

(2) 如果平坦(var < par)，则不再继续划分，停留在此层进行预测。

(3) 如果不平坦，就继续进行划分，直到划分到8x8，但在每一层都要进行预测块的过程，并保存相关信息。

### 3.3.2预测块划分快速算法

由于VP9帧内预测块新增加两种非对称划分方式PARTITION\_HORZ和PARTITION\_VERT。为了统计这两种划分被选为当前预测块最佳划分的概率，选择Game\_car\_720和RiverBe1080两个分辨率不同的序列在不同量化参数下进行编码，统计预测块划分为PARTITION\_HORZ和PARTITION\_VERT的概率如表3.3所示，可以看出，当前预测块的最佳划分方式为PARTITION\_HORZ和PARTITION\_VERT的概率很小。

表3.3 非对称模式选为最佳预测模式的概率

|  |  |  |
| --- | --- | --- |
| 概率  序列 | | 最优划分为Nx2N和2NxN |
| Game\_car\_720 | 140 | 0.080 |
| 160 | 0.082 |
| 180 | 0.079 |
| 200 | 0.076 |
| RiverBed1080 | 140 | 0.050 |
| 160 | 0.042 |
| 180 | 0.043 |
| 200 | 0.041 |

因为非对称模式被选作最佳预测块的概率很小，所以可以考虑直接去掉非对称模式，表3.4是直接去掉非对称模式后，测试了另两个序列的性能变化。从实验结果能够发现，在关闭非对称模式下，PSNR下降很小，几乎能够忽略，同时码率增加也较小。由此得出结论，非对称模式的打开与关闭对编码整体性能影响不大。所以本文VP9帧内预测过程中预测块的快速方法是直接关闭帧内预测块中非对称模式。

表3.4 不做非对称模式编码性能的变化

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试序列 | QP | (%) | (dB) | (dB) | (dB) |
| Kimono\_1080P | 140 | 0.51% | -0.007 | -0.015 | -0.005 |
| 160 | 0.42% | -0.013 | -0.015 | -0.009 |
| 180 | 0.25% | -0.025 | -0.02 | -0.008 |
| 200 | -0.14% | -0.045 | -0.015 | -0.009 |
| ParkScene \_1080P | 140 | 0.82% | -0.021 | -0.01 | -0.002 |
| 160 | 0.79% | -0.029 | -0.016 | 0 |
| 180 | 0.67% | -0.04 | -0.02 | 0.014 |
| 200 | 0.41% | -0.05 | -0.023 | 0.028 |

## 3.4算法结果与分析

将本文关于块划分的快速算法添加到VP9帧内预测中，对测试序列进行编码并与原编码器的编码性能进行了比较实验结果如表3.5。在测试中采用的量化参数分别为140、160、180和200。测试结果包括5项，分别为本算法相对原算法编码，时间减少量，比特率的增量，和Y、U、V分量的PSNR的变化量、和。

表3.5 本文快速算法测试结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试序列 | QP | (%) | (%) | (dB) | (dB) | (dB) |
| BasketballDrive\_1080P | 140 | -76.0 | 2.53 | 0.013 | 0.023 | -0.028 |
| 160 | -74.7 | 2.12 | -0.003 | 0.019 | -0.052 |
| 180 | -74.7 | 1.46 | -0.025 | 0.021 | -0.063 |
| 200 | -74.0 | 0.68 | -0.051 | 0.006 | -0.073 |
| Kimono\_1080P | 140 | -68.9 | 0.30 | -0.025 | -0.03 | -0.012 |
| 160 | -68.4 | 0.06 | -0.033 | -0.02 | -0.016 |
| 180 | -68.3 | -0.25 | -0.049 | -0.027 | -0.011 |
| 200 | -67.3 | -0.83 | -0.066 | -0.02 | -0.004 |
| ParkScene\_1080P | 140 | -68.3 | 0.39 | -0.051 | -0.027 | -0.005 |
| 160 | -67.3 | 0.26 | -0.054 | -0.016 | 0.01 |
| 180 | -66.9 | -0.01 | -0.06 | -0.018 | 0.026 |
| 200 | -67.9 | -0.25 | -0.066 | -0.014 | 0.039 |
| BQTerrace\_  1080P | 140 | -62.4 | 1.20 | -0.017 | 0.008 | 0.035 |
| 160 | -62.3 | 1.40 | -0.017 | 0.009 | 0.036 |
| 180 | -61.8 | 1.45 | -0.026 | 0.02 | 0.03 |
| 200 | -62.6 | 1.59 | -0.032 | 0.029 | 0.095 |
| Cactus\_1080P | 140 | -67.3 | 1.27 | -0.038 | -0.013 | -0.008 |
| 160 | -66.6 | 1.27 | -0.042 | -0.016 | -0.018 |
| 180 | -65.7 | 1.03 | -0.055 | 0 | 0.004 |
| 200 | -65.5 | 0.71 | -0.066 | -0.012 | -0.016 |
| 平均 | — | -67.85 | 0.82 | -0.038 | -0.004 | -0.002 |

实验结果表明，本章的快速算法极大地减少了编码复杂度。编码时间平均上降低67.85%，并且编码性能略有下降，比特率平均上升小于0.82%，Y分量的PSNR下降在0.04左右。

另外选取PSNR下降最多的序列Cactus\_1080P，在QP为140时采用快速算法编码与原算法编码后的解码图像进行了对比，期望能够直观地观察到本算法的编码效果。虽然当QP为140时Cactus序列的PSNR下降最大，但两个解码后的图像基本没有差别，如图3.8所示。



(a) 原算法编码后的解码图像



(b) 快速算法编码后的解码图像

图3.8 解码图像对比

## 3.5本章小结

本章主要是对块划分过程提出了一种快速算法。首先分析了VP9中的编码块的划分过程；然后分析了编码块的空间复杂度，并确定了衡量当前编码块复杂度的标准；另外分析了只采用SPLIT一种划分下编码器的性能情况；最后利用前面分析与实验结果，提出了一种利用空间复杂度和去除对编码器影响性能很小的划分方式的快速选择算法。该算法性能测试结果表明：在不同量化参数下，对于不同复杂度的序列，都能够在视频质量（PSNR）下降很小的同时，减少了大部分的编码时间。

# 第四章 基于模式选择的VP9帧内预测快速算法

## 4.1 引言

VP9作为Google推出的新一代视频编码器，视频传输比VP8编码器更加高效，但在编码效率的提升的同时编码复杂度也大大增加。因此对于实时视频通信，VP9快速算法的研究意义重大。

本章先介绍VP9的帧内预测的模式选择过程，结合第二章总结他人的帧内模式选择快速算法，提出一种适用于VP9的帧内预测快速算法。

## 4.2 VP9帧内预测模式选择过程

帧内预测技术是利用当前预测块及其相邻像素的空间相关性来减少图像的空间冗余。VP8的亮度预测块尺寸为4x4和16x16，色度预测块的尺寸为8x8、16x16。亮度块的预测模式共有4种（水平预测、垂直预测、DC预测和TM预测）；4x4亮度块的预测模式共10种；8x8色度块的预测模式同16x16亮度块相同。VP9预测块尺寸共13种，从4x4、4x8、8x4到32x64、64x32、64x64。VP9采用灵活的预测尺寸来更好的压缩效果，如采用较大的块来预测图像内容平坦的区域；采用小块预测图像内容较复杂的区域。并让每一个预测块都进行十种模式预测，来更精确地预测图像内容。VP9采用率失真优化（RDO, Rate Distortion Optimization）[34,35]进行最优编码模式选择，通过遍历所有的预测模式，最后选择率失真代价最小的模式最为最佳帧内预测模式。具体的模式选择过程如下：

1. 利用当前预测块的参考像素生成预测像素；
2. 预测像素与原始像素相减得到残差块；
3. 对残差块进行变换、量化，通过估计代价模型压缩比特数R；

4）对量化系数进行反量化，由于失真是由于量化过程造成的，用量化前系数（变换系数）和反量化后的系数的SSE作为预测块的失真D；

5）反量化系数经过反变换后再与预测像素相加得到重构块，重构像素用作后续预测块的参考像素；

6）根据率失真函数计算该模式下的率失真代价，计算公式如下：

 (4-1)

7）选择率失真代价最小的模式作为最优预测模式。



图4.1 VP9帧内模式率失真代价计算

VP9帧内模式率失真代价的计算如图4.1所示，VP9有4x4~64x64共13种尺寸的预测块，每种预测块都要对10种模式进行RDO选择出最优模式。一个64x64编码块的率失真代价的计算次数10+2x20+40+2x80+160+2x320+640+2x1280+2560共6810次。如果计算率失真代价都要遍历6810次，那么将导致编码时间大大增加。由于在每一次预测过程中还会包括残差、变换、量化等操作，所以实际次数会更多，因此该处是优化的重点。4.3 帧内模式选择各模块的时间复杂度分析

在整个预测过程中，包括了许多模块，每一个模块都会有一定的复杂度为了能够更多的减少编码时间，需要清楚每一个环节和模块的时间复杂度，这样才能有选择地对花费时间多的部分进行改进或替代。为此统计了VP9帧内预测过程中各个模块的时间复杂度，测试序列为VP9的五个参考序列，本算法主要针对1080P序列。



图4.2 QP=140时32x32变换块



图4.3 QP=140时16x16的变换块

由图4.2和图4.3可以看出，在每一次模式选择中，占据RDO大部分时间的模块是变换、量化和反变换、反量化。变化量化的基本单位是变换块，并且变换块相对于预测块是独立，在每个预测块中变换块有可能需要继续遍历，所以要实现VP9的实时编码，需要对变换块进行优化以减少变换块的遍历次数。

## 4.4 帧内预测模式快速选择算法

根据4.2和4.3的分析，在预测模式的快速选择过程中可以从两个方面来减少编码时间，一个是减少预测模式。，另一个是减少变换块的选择。主要思路是利用模式方向相似性提前终止模式选择，减少变换块主要是在预测块下用适当的变换块，而不是全选择。本算法主要从VP9的主要实现过程着手，对RDO内部的复杂度高的模块进行优化以减少编码时间。

### 4.4.1非角度预测模式选择

在对VP9采取的10种帧内预测模式进行分析时发现，DC和TM有很多相似性，DC是由当前块的上方和左侧已编码像素的均值填充得到，而TM是由当前块中每一个像素的正上方、左上方和正左侧对应的重建像素的平均得到，两者都不具有表达纹理和方向的特性，而且TM模式使用较少，时间复杂度比DC大，可以在一定程度上舍弃TM模式。本小节在此基础上设计实验，测试去除TM模式后序列的性能

表4.1 去除TM模式后性能测试结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试序列 | QP | (%) | (%) | (dB) | (dB) | (dB) |
| Basketball-Drive\_  1080P | 140 | -11.9 | 0.55 | -0.008 | 0.01 | -0.003 |
| 160 | -11.8 | 0.41 | -0.013 | -0.009 | -0.01 |
| 180 | -9.3 | 0.27 | -0.022 | 0.022 | -0.008 |
| 200 | -9.2 | -0.04 | -0.034 | -0.027 | -0.014 |
| Kimono\_  1080P | 140 | -12.1 | -0.006 | -0.011 | -0.012 | -0.012 |
| 160 | -11.1 | -0.007 | -0.009 | -0.011 | -0.011 |
| 180 | -11.3 | -0.014 | -0.015 | -0.01 | -0.01 |
| 200 | -9.2 | -0.026 | -0.012 | -0.002 | -0.002 |
| ParkScene\_1080P | 140 | -10.5 | -0.019 | -0.001 | -0.001 | -0.002 |
| 160 | -9.8 | -0.019 | -0.008 | -0.008 | -0.002 |
| 180 | -8.1 | -0.023 | -0.006 | -0.006 | 0.009 |
| 200 | -10.1 | -0.024 | -0.007 | -0.007 | 0.013 |
| BQTerrac-e\_1080P | 140 | -12.0 | 0.1 | -0.042 | 0.006 | 0.008 |
| 160 | -9.6 | -0.03 | -0.046 | 0.009 | 0.008 |
| 180 | -8.2 | -0.13 | -0.048 | 0.014 | 0.0 |
| 200 | -11.1 | -0.41 | -0.052 | 0.008 | 0.035 |
| Cactus\_  1080P | 140 | -11.7 | 0.1 | -0.021 | -0.006 | -0.002 |
| 160 | -10.5 | -0.01 | -0.025 | 0 | 0 |
| 180 | -9.4 | -0.17 | -0.031 | 0.002 | -0.002 |
| 200 | -8.1 | -0.45 | -0.035 | -0.005 | -0.013 |
| 平均 | — | -10.25 | -0.06 | -0.025 | -0.0022 | -0.0006 |

从上面的实验结果可以看出，在去除TM模式后编码时间下降了10%左右，并且码率下降了0.06%，PSNR下降了很少。理论上编码时间减少10%，因此实验结果符合预期且对于除去TM模式是可行的，对大多数序列是性能损失很少，但编码时间节约了10%。

### 4.4.2角度预测模式选择

在上一小节针对具体10种模式进行了分析，对非角度预测进行了初步选择。VP9的10种帧内预测模式中，有8种模式是具有方向性的，且8种角度模式具有一定的方向相关性，若当前最优模式为D45\_PRED则下一个最优模式应在45°周围寻找，其他模式跳过，这样可以有效地减少编码时间。

具体算法步骤伪代码：

If current\_mode == D117\_PRED &&

best\_intra\_mode != V\_PRED &&

best\_intra\_mode != D135\_PRED)

Skip current\_mode;

If (current\_mode == D63\_PRED &&

best\_intra\_mode != V\_PRED &&

best\_intra\_mode != D45\_PRED)

Skip current\_mode;

If (current\_mode == D207\_PRED &&

best\_intra\_mode != H\_PRED &&

best\_intra\_mode != D45\_PRED)

Skip current\_mode;

If (current\_mode == D153\_PRED &&

best\_intra\_mode != H\_PRED &&

best\_intra\_mode != D135\_PRED)

Skip current\_mode;

表4.2 利用方向相关性性能测试结果

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 测试序列 | | QP | | (%) | | (%) | | (dB) | | (dB) | | (dB) | |
| Basketbal-lDrive\_1080P | | 140 | | -25.0 | | 0.6 | | 0.002 | | -0.003 | | -0.003 | |
| 160 | | -26.4 | | 0.70 | | 0.002 | | -0.021 | | -0.01 | |
| 180 | | -14.8 | | 0.72 | | -0.003 | | -0.016 | | -0.008 | |
| 200 | | -31.6 | | 0.76 | | -0.009 | | -0.052 | | -0.014 | |
| ParkScene\_1080P | | 140 | | -32.1 | | 0.82 | | -0.027 | | 0.0 | | -0.031 | |
| 160 | | -29.7 | | 0.75 | | -0.015 | | -0.036 | | -0.019 | |
| 180 | | -25.5 | | 0.54 | | -0.021 | | -0.033 | | -0.018 | |
| 200 | | -35.5 | | 0.16 | | -0.022 | | -0.028 | | -0.004 | |
| 平均 | | — | | -30.65 | | 0.57 | | -0.021 | | -0.024 | | -0.018 | |

从上面的实验结果可以看出，在利用方向相关性后编码时间下降了30%左右，并且码率下降了0.57%，PSNR几乎不变。

### 4.4.3预测过程中变换尺寸的选择

在预测块进行预测之后，为了能够尽可能多的保留图像的纹理信息，VP9在预测块中进行了变换块的最优选择，其方式是遍历所有小于当前预测块大小的变换块。根据4.2小节的帧内预测过程的分析，一个64x64的编码块要进行的RDO总次数为10+2x20+40+2x80+160+2x320+640+2x1280+2560共6810次。而变换却要在这6810次中再进行全遍历，这样极大地增加了编码的复杂度。而且变化块中的操作是RDO过程中最耗时的模块，所以对变换块进行优化是十分必要的。

近年来图片比较倾向于大分辨率，VP9中超级宏块也是针对大分辨率提出的，在大分辨率中，若被划分的块是Superblock，其一般都是背景区，纹理表现很平坦，对于它如果采用全遍历选择变换块，编码性能提升不大。对于纹理丰富的区域，一般会采用小划分，变换块也不会增加很多。因此，本文将采用与预测块相同尺寸的变换块，并针对该算法进行了性能测试。

表4.3本文算法性能测试结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 测试序列 | | QP | (%) | (%) | (dB) | (dB) | (dB) |
| Cactus\_  1080P | | 140 | -53.3 | 0.74 | 0.008 | -0.002 | -0.011 |
| 160 | -52.2 | 0.85 | 0.008 | -0.002 | -0.017 |
| 180 | -52.5 | 0.74 | 0.003 | 0.001 | -0.027 |
| 200 | -52.7 | 0.24 | -0.02 | -0.01 | -0.026 |
| Kimono\_1080P | | 140 | -52.4 | 0.21 | 0.005 | 0.01 | 0.007 |
| 160 | -53.2 | 0.19 | 0.007 | 0.004 | 0.003 |
| 180 | -53.6 | 0.20 | 0.011 | 0.014 | 0.016 |
| 200 | -54.6 | 0.32 | 0.017 | 0.021 | -0.006 |
| ParkScen-e\_1080P | | 140 | -52.4 | 0.39 | 0.003 | -0.004 | -0.002 |
| 160 | -53.2 | 0.45 | 0.004 | 0 | -0.01 |
| 180 | -54.1 | 0.31 | -0.003 | -0.005 | 0.019 |
| 200 | -56.3 | 0.06 | -0.004 | -0.003 | -0.029 |
| 平均 | — | -53.38 | 0.39 | 0.0048 | 0.0025 | -0.01 |

本实验结果显示，在设置预测块尺寸等于变换块尺寸的情况下，编码时间减少了53%，码率只上升了0.39%，PSNR几乎没有下降，可以看出该方法能够很好的提高编码性能

## 4.5实验结果与分析

将上述快速算法添加到VP9帧内预测中，对测试序列进行编码并与原编码器的编码结果进行比较。本测试中采用的量化参数分别为140、160、180和200。测试结果包括5项，分别为本算法相对原算法编码时间的减少量，比特率的增量和PSNR的变化量、和。

基于模式选择的帧内快速选择算法总体算法如图4.4



图4.4 帧内模式选择的快速算法

表4.4 本文快速算法测试结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试序列 | QP | (%) | (%) | (dB) | (dB) | (dB) |
| Kimono\_  1080P | 140 | -68.7 | 0.26 | -0.017 | -0.012 | -0.012 |
| 160 | -69.7 | -0.03 | -0.029 | -0.011 | -0.011 |
| 180 | -69.7 | -0.50 | -0.047 | -0.01 | -0.01 |
| 200 | -70.0 | -1.02 | -0.066 | -0.002 | -0.002 |
| ParkScene\_1080P | 140 | -69.5 | 0.53 | -0.029 | -0.036 | -0.02 |
| 160 | -68.6 | 0.39 | -0.034 | -0.032 | -0.025 |
| 180 | -69.2 | -0.19 | -0.048 | -0.036 | 0.018 |
| 200 | -70.2 | -0.98 | -0.06 | -0.028 | 0.019 |
| BQTerrace\_1080P | 140 | -68.5 | 1.75 | -0.013 | -0.049 | -0.056 |
| 160 | -67.5 | 1.99 | -0.015 | -0.035 | -0.061 |
| 180 | -67.7 | 2.15 | -0.023 | -0.027 | -0.089 |
| 200 | -68.8 | 2.13 | -0.029 | -0.043 | -0.089 |
| Cactus\_  1080P | 140 | -67.9 | 1.36 | -0.019 | -0.02 | -0.036 |
| 160 | -68.5 | 1.34 | -0.027 | -0.024 | -0.059 |
| 18 | -68.2 | 1.06 | -0.039 | -0.032 | -0.068 |
| 200 | -68.7 | 0.44 | -0.053 | -0.042 | -0.075 |
| 平均 | — | -68.85 | 0.67 | -0.034 | -0.035 | -0.045 |

实验结果表明，本文的快速算法可以较大幅度的降低帧内编码的复杂度，使编码时间平均上降低68.85%。于此同时，编码性能几乎没有改变，比特率平均上升小于1%，PSNR下降在0.035左右。本算法达到了预期效果。

## 4.6 本章小结

本文经过试验统计和分析其他帧内预测的算法，提出了一种适用于高分辨率的VP9帧内预测快速算法。算法的核心在于减少模式选择的次数和优化RDO内部计算过程。通过利用角度预测模式的方向相关性和非角度模式的相似性，排除了部分预测模式。另外结合帧内预测各个模块的复杂度，减少高复杂度模块的计算次数。实验结果表明，本章的快速算法在视频质量下降很小的情况下（PSNR）大幅度降低了编码复杂度，减少编码时间。

# 第五章 总结与展望

随着移动设备的迅速发展，高清和超高清视频信息量成倍数的增长，对视频通信的实时性要求越来越高。VP9在VP8的基础上应用很多新技术，使VP9的编码效率有了很大的提高。但是编码效率提升的同时，也增加了VP9的编码复杂度，导致其编码时间大量增加。本文就该问题，从帧内预测方面提出了两种适用于VP9的快速算法，期望VP9编码标准能够适用于实时业务，所以对以后VP9编码器的推广具重大意义。

本文先分析了VP9视频编码的特点和基本框架，然互具体分析了编码结构和预测编码技术，接着主要研究了和帧内预测过程相关的两个关键模块：块划分和模式选择。最后根据分析这两个模块的影响因素和利用这两个因素进行试验的结果，提出了VP9帧内预测的优化算法：基于块划分的帧内快速算法和基于模式选择的帧内快速算法。

在基于块划分的帧内快速算法中，首先确定了最优块划分的影响因素：空间复杂度和编码量化参数。接着确定了空间复杂度的表示方法，并利用空间复杂度进行了试验分析：平坦区域选择大划分，复杂区域选择小划分。在对量化参数的分析中，主要利用量化参数来自适应地调整平坦度门限。实验结果表明，该算法能够在大幅度减少编码时间的同时，使得编码效率下降很少。

在基于模式选择的帧内快速算法中，本文通过利用预测模式的方向相似性减少帧内预测的候选模式和优化了变换块的计算过程。该算法排除了大部分预测模式和减少了大量RDO的计算量，降低了编码复杂度，减少了65%左右的编码时间。

本文所做工作主要是在VS2010的实验环境下，实验结果都是在服务器Dell PowerEdge R720上得到的。虽然本文的研究取得了一些成果，但是由于时间所限，本文提出的快速算法仍有进一步改进的余地。

对未来的展望：

在本人所提出的算法中涉及到门限的设置，该门限是根据大量实验数据来设置的，不具有自适应性，所以如何让门限自适应也要进一步的研究。 本文提出的两种快速算法都是相互独立的，理论上可以结合使用，但可能降低编码效率，这些问题还需要进一步的确认和研究。

# 致 谢

论文完成之际，向给我关心和帮助的老师、同学、朋友和家人表示衷心的感谢。

首先，特别感谢我的研究生导师杨付正教授。在这半年多的毕设阶段，杨老师不仅在科研学术上给予了我耐心的教诲和指导，在生活态度、人生观和世界方面也深深影响了我。杨老师渊博的学识、严谨的科研作风、高尚的个人修养一直是我努力地方向和目标。正是由于他的严格要求和耐心指导，我才可以顺利的完成本科毕业设计。

非常感谢杨老师对我论文的指导和帮助，让我能够顺利按时的完成本科毕业设计。在毕业设计期间遇到的每个困难，杨老师都会和我认真讨论并热心地帮助我解决问题。在解决毕业设计遇到的各种问题时，不仅锻炼了我解决问题的能力，也提高了我思考问题的方式，极大地开阔了眼界，使我的思考和解决问的方式都发生了很大的改变。另外，通过一个学期的毕业设计，我也学会了坚持和不怕挫折的品质，这将是我人生中重要的财富。

感谢贺竟博士对我的帮助。感谢师兄师姐戚兴春、王一、李娜以及何艳坤、张新、吴丽珍、王鹏、徐子强、王佳慧、冯俊凯等师哥师姐们对我的关心和帮助。感谢我的舍友余宏捷、赵阳浩、边宇凡、朱明涛、路家俊、王文甲、周凌冬、付光耀、冯收、杨飞、唐强，他们令我度过了难忘而美好的本科生生活。深深感谢我的家人，感谢他们一直以来对我的支持和理解。他们的关爱和鼓励，让我能够在本科四年中能够安心的学习，并且快乐地度过我四年的本科生活，

最后向参加论文审阅和答辩的专家和老师们表示感谢。感谢你们能在百忙之中抽出宝贵时间来参加我们的答辩，并提出珍贵的意见。

# 参考文献

[1] Gary J.Sullivan, Jens-Rainer Ohm, Woo-Jin Han, et al. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology. 2012.

[2] Thomas Wiegand, Gary J.Sullivan, Ajay Luthra, et al. Overview of the H.264/AVC Video Coding Standard. IEEE Transactions on Circuits and Systems for Video Technology. 2011.

[3] 何海东, 董全武, 陈宏. HEVC & VP9在网络直播点播中的应用探索(上). 广播与电视技术. 2014, 41(8). 171-172.

[4] 刘小鹤. 开放的VP9谷歌新一代互联网高清视频编码. 微型计算机. 2014

[5] 郭奕星. 下一代数字视频技术应用前瞻. 电信快报. 2014

[6] 毕厚杰. 新一代视频压缩编码标准—H.264/AVC. 人民邮电出版社. 2005.

[7] CCITT/SG XV. Codecs for videoconferencing using primary digital group transmission. in Recommendation H.120, CCITT (currently ITU-T), Geneva, 1989.

[8] Wei Jiang, Hanjie Ma, and Yaowu Chen. Gradient Based Fast Mode Decision Algorithm for Intra Prediction in HEVC. Signal Image Technology and Internet Based Systems（SITIS）,2012 Eighth International Conference on. 2012, 225-229.

[9] Zhang M, Zhao C, and Xu J. An adaptive fast intra mode decision in HEVC. Proceedings of International Conference on Image Processing (ICIP). 2012: 221-224.

[10] Jiang W, Ma H, and Chen Y. Gradient based fast mode decision algorithm for intra prediction in HEVC. Processdings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet). 2012: 1836-1840.

[11] Da S, Thaisa L, Agostini L V, et al. Fast HEVC intra prediction mode decision based on EDGE direction information. Proceedings of the 20th European Signal Processing Conference (EUSIPCO). 2012: 1214-1248.

[12] F Pan, X Lin. Fast Mode Decision for intra Predicition. Circuits and Systems for Video Technology. 2005, 7(8): 813-822.

[13] J.F Wang, J.C Wand. A Novel Fast Algorithm for Intra Mode Decision in H.264/AVC Encoder. ISCAS. 2006: 3498-3501.

[14] Feng Pan, Xiao Lin, and Susanto Rahardja. Fast Mode Decision Algorithm for Intra Prediction in H.264/AVC Video Coding. IEEE Transaction on Circuits and Systems for Video Technology. 2005.

[15] 苏睿, 刘贵忠, 张彤宇, 等. 利用变换域信息快速实现H.264帧内预测编码的新算法. 电子与信息学报. 2007, 29(1).161-164.

[16] 余成伟, 陆建华, 郑君里.H.264编码模式选择快速算法研究. 清华大学学报:自然科学版. 2007(10).

[17] B.J.Meng, Au.O.C, W.Wong, et al. Efficient Intra-predicition mode selection for 4x4 blocks in H.264. Proceedings of the 2003 IEEE International Conference on Multimedia and Expo (ICME). 2003.

[18] Zhao Liang, Zhang Li, Ma Siwei, et al. Fast mode decision algorithm for intra prediction in HEVC. IEEE Visual Communications and Image Processing(VCIP). 2011.

[19] Jeahwan Kim, Jungyoup Yang, Hoyoung Lee, et al. Fast intra mode decision of HEVC based on hierarchical structure. 2011 8th International Conference on Information, Communications and Signal Processing(ICICS). 2011.

[20] Motra A.S, Gupta A, Shukla M, et al. Fast intra mode decision for HEVC video encoder. 2012 20th International Conference on Software Telecommunication and Computer Networks (SoftCOM). 2012.

[21] M.G.Sarwer, Q.M.J.Wu. Efficient Two Step Edge based Partial Distortion Search for Fast Block Motion Estimation. IEEE Transactions on Consumer Electronics.2009. 2154-2162.

[22] Kim Y, Jun D, Jung S-H, et al. A fast intra-prediction method in HEVC using rate-distortion estimation based on hadamard transform. ETRI JOURNAL. 2013, 25(2). 270-280.

[23] Yi-Hsin Huang, Tao-Sheng Ou, and Homer H.chen. Fast Decision of Block Size, Predicition Mode, and Intra Block for H.264 Intra Prediction. Circuits and Systems for Video Technology, IEEE Transactions on. 2010, 20(8). 1122-1132.

[24] Guifen Tian, and S.Goto. Content adaptive prediction unit size decision algorithm for HEVC intra coding. Picture Coding Symposium (PCS). 2012. 405-408.

[25] Liquan Shen, Zhaoyang Zhang, and Ping An. Fast CU size decision and mode decision algorithm for HEVC intra coding. Consumer Electronics, IEEE Transactions on. 2013,59(1). 207-213.

[26] Ting Y-C, Chang T-S. Fast intra prediction algorithm with transform domain edge detection for HEVC. Preceedings of IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS). 2012: 144-147.

[27] K.Saurty, P.C.Catherine and K.M.S.Soyjaudah. Early CU size determination in HEVC intra prediction using Average Pixel Cost. Digital Information and Technology and it’s Applications (DICTAP). 2014. 247-252.

[28] Jongho Kim, Yoonsik Choe, Yong-Goo Kim. Fast coding unit size decision algorithm for intra coding in HEVC. Proceedings of IEEE International Conference on Consumer Electronics. 2012: 637-638.

[29] Gang He, Dajiang Zhou, and S.Goto. Transform-based fast mode and depth decision algorithm for HEVC intra prediction. 2013 IEEE 10th International Conference on ASIS (ASICON). 2013.

[30] Zhenxing Chen, and S.Goto. A QP and Partition-Size Statistic Based Fuzzy Algorithm for Fast Inter & Intra Mode Decision in Video Coding. Image and Signal Processing, 2009. CISP’09. 2nd International Congress on. 2009.