

中图分类号: TN919.81

单位代号: 10280

密 级: 公开

学 号: 18721542

上海大学



硕士学位论文

SHANGHAI UNIVERSITY
MASTER'S DISSERTATION

题 目	H.265 无损帧内编码算法优化 及硬件实现
--------	---------------------------

作 者 林庆毫

学科专业 信号与信息处理

导 师 林敏

完成日期 二〇二一年五月

上海大学

本论文经答辩委员会全体委员审查，确
认符合上海大学硕士学位论文质量要求。

答辩委员会签名：

主席：

委员：

导 师：

答辩日期：

原创性声明

本人声明：所呈交的论文是本人在导师指导下进行的研究工作。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已发表或撰写过的研究成果。参与同一工作的其他同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

签名：_____ 日期：_____

本论文使用授权说明

本人完全了解上海大学有关保留、使用学位论文的规定。即：学校有权保留论文及送交论文复印件，允许论文被查阅和借阅；学校可以公布论文的全部或部分内容。

(保密的论文在解密后应遵守此规定)

签名：_____ 导师签名：_____ 日期：_____

上海大学工学硕士学位论文

H.265 无损帧内编码算法优化及硬件实现

作 者: 林庆毫

导 师: 林敏

学科专业: 信号与信息处理

通信与信息工程学院

上海大学

2021 年 5 月

A Dissertation Submitted to Shanghai University for the
Degree of Master in Engineering

Algorithm Optimization and Hardware Implementation of H.265 Lossless Intra Frame Coding

Candidate: Qinghao Lin

Supervisor: Min Lin

Major: Signal and information processing

School of Communication and Information Engineering

Shanghai University

May, 2021

摘 要

H.265 视频编码标准自 2013 年发布以来, 凭借其高压缩率、低失真度的优秀表现, 被广泛应用在各类产业中。但在高精度需求的场景中, 不产生失真的无损编码仍扮演着重要角色, 例如指纹图像、航天器遥感图像存储以及疫情影响下的医学图像传输、远程桌面等。本课题通过对 H.265 无损帧内编码的深入研究, 归纳出其在帧内预测和编码块划分流程中存在的缺陷, 针对性地提出优化算法。本论文的主要创新点如下:

1) 针对帧内预测在远离参考点的区域预测准确度低的缺陷, 提出 L 形迭代预测算法, 使所有待预测点与参考点的距离缩短到 1 单位, 大幅提高帧内预测准确度; 2) 针对四叉树结构下的编码块划分无法适应小片区域复杂纹理的缺陷, 提出 L 形分块算法, 使编码块划分更加灵活, 同时与 L 形迭代预测有机融合, 提出联合算法, 进一步提高编码效率; 3) 针对帧内预测残差仍具有特殊空间冗余的缺陷, 提出残差中值边缘检测算法, 对残差进行二次预测, 有效降低其空间冗余, 提高编码效率。

除了上述算法创新, 课题还开展了硬件实现与验证方面的工作。1) 在 H.265 硬件编码器上实现上述改进算法, 通过仿真收集测试数据, 对比软硬件优化效果; 2) 利用 FPGA 搭建硬件编码系统原型验证平台, 通过该平台可对提出的各类改进算法进行硬件级测试验证。

基于联合专家组制定的通用测试环境和标准测试序列对本课题所提算法进行测试。实验结果表明, 与 H.265 标准测试模型相比, 上述 3 个算法分别得到了 9.31%, 2.34%, 7.04% 的码率优化, 联合算法得到了 10.21% 的码率优化, 证明所提算法性能良好。在 FPGA 验证中, 得到了与软件测试相当的优化效果, 证明所提算法硬件实现可行, 有较高的应用价值。

关键词: 无损视频编码; H.265; 帧内预测; 编码块划分; 硬件实现

ABSTRACT

Since the release of H.265 video coding standard in 2013, with its excellent performance of high coding efficiency and low distortion, it has been widely used in many industries. However, the lossless video coding still plays an important role in some high-precision fields, such as fingerprint image storage, satellite remote sensing image storage and the medical image transmission and remote desktop under the impact of COVID-19, since it is distortion-free. Through the in-depth study of H.265 lossless intra frame coding, this paper summarizes the defects in intra prediction and coding unit partitioning, and proposes the optimization algorithms. The innovations in the dissertation as follows:

1) Aiming at the low accuracy of intra prediction in the region far away from the reference samples, an L-shape-based Iterative Prediction algorithm is proposed, which shortens the distance between predicted samples and reference samples to 1 unit, thereby greatly improves the accuracy of intra prediction; 2) In view of the defect that the coding unit partitioning under quadtree structure can not adapt to the rich texture of small area, this paper proposes an L-shape-based partitioning algorithm, which makes the coding unit partition more flexible. At the same time, it can integrate with L-shape-based Iterative Prediction and become a joint algorithm to further improve the coding efficiency; 3) In view of the special spatial redundancy of intra prediction residuals, Residuals Median Edge Detect algorithm is proposed to predict the residuals again, which can effectively reduce the special spatial redundancy and improve the coding efficiency.

In addition to the above algorithm innovations, we also carried out the work of hardware implementation and verification. 1) The improved algorithm is implemented on the basis of H.265 hardware encoder, and the test data is collected through behavioural simulations to compare the gain of software and hardware; 2) FPGA is used

to build the prototype verification platform of hardware video coding system, through which proposed optimization algorithms can be verified and tested at the hardware level.

Proposed algorithms are tested using the standard test sequences under common test condition specified by joint expert group. Experimental results show that, compared with H.265 test model, the above three algorithms get 9.31%, 2.34% and 7.04% bit-rate saving respectively, and the joint algorithm gets 10.21% bit-rate saving, which proves that the proposed algorithms are significantly optimized. In FPGA verification, the optimization effect is equivalent to that of software testing, which proves that the hardware implementation of the proposed algorithms are feasible and worthy of application.

Keywords: Lossless video coding; H.265; Intra frame prediction; Coding unit partition; Hardware implementation

目 录

第一章 绪论	1
1.1 课题研究背景与意义	1
1.2 国内外研究现状	1
1.3 本课题的主要工作	3
1.4 本文章节安排	4
第二章 视频编码原理与关键技术	5
2.1 视频编码技术发展历史	5
2.2 预测编码技术	6
2.2.1 帧内预测	8
2.2.2 帧间预测	9
2.3 变换与量化技术	9
2.3.1 变换编码	10
2.3.2 量化	11
2.4 环路后处理技术	13
2.4.1 去方块滤波	13
2.4.2 样点自适应补偿	14
2.5 熵编码技术	15
2.5.1 变长编码在视频编码中的应用	15
2.5.2 算术编码在视频编码中的应用	16
2.5.3 模式依赖的编码顺序	17
2.6 率-失真优化技术	19
2.7 软、硬件视频编码项目	20
2.7.1 软件视频编码项目	20
2.7.2 硬件视频编码项目	21
2.8 本章小结	21

第三章 帧内无损编码优化算法	23
3.1 帧内无损编码的可优化方向	23
3.2 帧内无损编码的预测过程优化	24
3.2.1 帧内预测过程分析	24
3.2.2 L 形迭代预测算法	27
3.2.3 算法性能测试与分析	29
3.3 帧内无损编码的编码块划分优化	31
3.3.1 编码块划分决策过程分析	33
3.3.2 L 形分块算法	34
3.3.3 算法性能测试与分析	35
3.4 帧内无损编码的待编码系数再处理	35
3.4.1 帧内预测残差分析	35
3.4.2 残差中值边缘检测算法	38
3.4.3 算法性能测试与分析	40
3.5 联合算法及性能测试	45
3.6 本章小结	49
第四章 H.265 帧内无损编码及优化算法硬件实现	50
4.1 硬件系统框架	50
4.2 关键模块的硬件实现	51
4.3 行为级仿真与测试	52
4.4 本章小结	54
第五章 H.265 帧内无损编码 FPGA 原型验证平台	55
5.1 FPGA 原型验证方案	55
5.2 FPGA 验证平台关键模块	56
5.2.1 FPGA 器件	56
5.2.2 视频源及预处理	57
5.2.3 与上位机通信接口	58
5.2.4 编码模块	59

5.2.5 上位机解码模块	59
5.3 FPGA 原型验证结果	60
5.4 本章小结	62
第六章 总结与展望	63
6.1 课题总结	63
6.2 工作展望	63
插图索引	65
表格索引	67
参考文献	68
作者在攻读硕士学位期间发表的论文与研究成果	73
致 谢	74

第一章 绪论

1.1 课题研究背景与意义

H.264/AVC(Advanced Video Coding)、H.265/HEVC(High Efficiency Video Coding)、H.266/VVC(Versatile Video Coding)是由国际电信联盟与国际标准化组织联合专家组制定的一系列视频压缩编码标准^[1-3],是数字视频编码技术 40 年的学术研究和 30 年的标准化的成果,代表着自 2003 年至今各时期最先进的视频编码技术。帧内编码是 H.26X 系列标准中的重要组成部分,其利用图像的空间相关性进行数据压缩,视频的关键帧也只使用帧内编码处理。加之近年来,在传感器分辨率不断提高、存储价格不断下降的催化下,自动驾驶视觉、云游戏、后期制作、影像存档等需要使用无损编码的应用加速发展^[4]。同时,在某些高精度需求的场景中无损编码仍是不可或缺的存在,例如来之不易的卫星遥感图像的编码、涉及安全保障的指纹图像存储、全球疫情影响下的医学影像传输、远程桌面共享等。因而探索无损帧内编码的优化算法有很高的研究和应用价值。

本课题的另一项重点研究内容是 H.265 编码器及优化算法的硬件实现。实时视频编码在监控、直播等场景中应用广泛,实时编码一般有 2 类实现方案: 1) 利用数字信号处理器 (Digital Signal Processor, DSP) 实现; 2) 使用专用视频编码硬件实现,例如各类手机处理器中集成的视频编解码单元。由于 H.265 实时编码对硬件的性能和功耗需求都极高^[5],使用 DSP 实现难度极大,实现后编码能力低。专用视频编码器经优化后可得到良好的性能与功耗的折衷,因此研究专用 H.265 编码器的硬件实现有很高的应用价值。

1.2 国内外研究现状

H.26X 系列编码标准可以通过简单地跳过变换、量化、去块滤波、自适应样点补偿等可能引入失真的步骤,实现无损帧内编码^[6]。但也因此使得待编码系数具有较高的能量,为后续的熵编码带来极大的压力。为优化 H.26X 系列标准无

损帧内编码的性能,国内外学者进行了大量的研究。Kamisli^[7]提出了一种不会引入失真的整数离散正弦变换,可在不增加数据动态范围的情况下将样点值映射到变换域,并且可用于无损编码,一定程度上弥补了帧内无损编码时跳过变换带来的码率损失;Zhang 等^[8]分析了亮度通道与色差通道之间的关系,得出编码块中亮度和色差分量之间存在线性相关性的结论,据此设计了3个模型,使用亮度通道对色差通道进行帧内预测,一定程度上降低了需要编码的数据量;残差差分脉冲编码调制 (Residual Differential Pulse Code Modulation, RDPCM) 是 H.265 屏幕图像编码扩展标准 (HEVC Screen Content Coding Extension, HEVC-SCC)^[9]的一部分,通过对残差进行再处理的方式进一步去除残差间的相关性,从而提高了编码效率^[10];区块差分脉冲编码调制 (Block Differential Pulse Code Modulation, BDPCM)^[11]是 H.266 标准的一部分,通过在水平或垂直方向上使用临近像素迭代预测的方式提高了预测准确性;Zhou 等^[11-15]将 DPCM 的思想扩展到了任意预测角度,提出了基于样本点的帧内角度预测 (Sample-based Angular Intra-Prediction, SAP) 及其多种衍生方案,使用逐点进行预测-重建的方案代替整块处理的原始方案,进一步提高了预测准确性;Sanchez 等^[16]通过分析残差块中数据的分布特征,提出了将待编码数据的“重心”进行搬移的新颖方案,其针对不同的分布特征设计了3组映射规则,整体降低了残差的能量;元辉等^[17]提出若在编码时发现当前预测单元各预测模式的结果高度相似,则不必编码模式,而是将所有预测模式的均值作为预测结果,一定程度上减少了需要编码的模式信息;Li 等^[18]发现使用最靠近预测单元的单行单列样点作为参考点的方式有时无法得到最佳预测结果,反而部分远距离参考点更有利于进行预测,因而提出通过将参考点拓展到多行的方式提高帧内预测的准确性。除此之外,还有将深度学习^[19]、神经网络^[20]、遗传算法^[21]与图像编解码相结合进行的研究,与本课题及传统的编解码研究出入过大,按下不表。

上述研究中提出的算法对帧内编码的性能均有不同程度的优化,但仍存在以下待改进的问题:[7]设计的整数变换方法局限性很大,且没有量化的配合很难降低待编码系数的能量,因此带来的码率优化较小;[9, 16]设计的残差分段映射方案没有考虑残差独特的空间相关性,仍有优化空间;[8, 11-15, 17, 18]均是

设法增加或优化预测模式来提高帧内预测的准确性，然而各代 H.26X 标准中的帧内预测模式不断增多，在最新的 H.266 中已经达到 67 种^[22]，很难再挖掘出优化空间，可以预见今后这类改进带来的优化会越来越小；最后，上述大部分方案具有一定的应用范围的局限性，很难在各世代的 H.26X 标准中通用。

1.3 本课题的主要工作

本课题的主要目的是提高 H.265 无损帧内编码效率，即在保证视频经过编解码后无失真的情况下，提高编码后的压缩率。为此，本文对 H.265 的核心编码过程进行了详尽的分析，提炼出 3 个可优化的方向并进行了软、硬件实现，通过测试验证达到了预期的性能指标。本课题的主要工作内容概述如下：

1. 整体上介绍视频编码的基本原理，解释帧内/帧间预测、变换量化、熵编码、环路滤波和率-失真优化 5 个核心手段在视频编码过程中发挥的作用；
2. 进一步分析帧内预测过程及原理，针对其部分待预测像素距离参考点距离过远的缺陷，提出了一种 L 形迭代预测 (L-shape-based Iterative Prediction, LIP) 算法，将待预测像素与参考点的距离缩短到 1，极大地提高了帧内预测的准确性，从而提高编码效率；
3. 进一步分析帧内分块决策过程，配合上述 LIP 算法，提出了一种不同于 H.265 传统的基于四叉树结构的分块方法，即 L 形分块 (L-shape-based Block Partitioning, LBP) 算法。LIP 与 LBP 互相配合，进一步提高编码效率；
4. 进一步分析系数的编码过程，利用待编码系数中存在的特殊的空间结构性，提出了一种残差中值边缘检测 (Residuals Median Edge Detect, RMED) 算法，通过对预测残差进行二次处理降低待编码系数的整体能量，进一步提高编码效率；
5. 在 H.265 硬件编码器的基础上，实现上述改进算法，同时利用 FPGA(Field Programmable Gate Array) 搭建硬件编码系统原型验证平台，对提出的改进算法进行测试验证。

1.4 本文章节安排

本文由 6 个章节组成，具体安排如下：

本章为绪论，概述课题研究背景与意义，介绍本课题相关的国内外研究状况，并简要阐述本课题的工作。

第 2 章首先从整体上介绍视频编码技术的发展历史，其中主要介绍 H.265，亦会涉及到其前身 H.264 及 2020 年正式发布的 H.266；同时从帧内/帧间预测、变换量化、熵编码、环路滤波和率-失真优化 5 个视频编码的核心手段展开，简单介绍技术要点；最后介绍部分学术界和工业界常用的软、硬件视频编解码相关项目，本课题就是以其中部分项目为基础开展的。

第 3 章是本文的核心部分之一，该章详细地探讨了帧内无损编码优化的可行性。通过对帧内预测过程、帧内分块决策过程和系数编码过程的进一步分析，提出了 4 个优化算法，分别涉及 1) 帧内预测过程的优化；2) 帧内分块过程的优化；3) 待编码系数的再处理。最后对 4 种优化算法进行单独、联合测试，给出优化算法的性能测试结果。

第 4 章开始介绍 H.265 硬件编码器的实现，详细描述了 H.265 硬件编码器的实现框架和部分关键模块的实现方案，并给出行为级仿真结果。

第 5 章介绍完整的 H.265 硬件视频编码系统的 FPGA 原型验证平台，详细描述平台验证方案，对各模块的划分及承担的任务进行说明，最后给出硬件原型验证结果并进行分析。

第 6 章对本课题的全部工作进行总结，并针对下一步的研究工作作出展望。

第二章 视频编码原理与关键技术

本章由简要的视频编码技术发展史开始，引出视频编码中最为核心的 5 项技术：预测编码、变换量化 (Transform and Quantization, TransQuant)、环路后处理、熵编码以及率-失真优化 (Rate-Distortion Optimization, RDO)，为后文分析视频编码的可优化方向以及分析所提出算法的可行性打下基础。在早期视频编码标准中并未包含上述所有核心技术，但随着各类应用对视频编码效率的需求日渐严苛，该 5 项核心技术被证明对编码效率提高起着核心作用，成为了缺一不可的存在。2020 年刚确定最终标准的 H.266^[1] 的发展趋势亦是在该框架下再进行部分精细优化。

在正文之前，首先申明以下用词：1) 统一使用视频“编码”而非视频“压缩”。在视频编解码领域，“编码”与“压缩”表达相同的含义，都指经过预测、变换量化、熵编码等操作达到减小数据冗余、压缩视频数据量的目的。编码是手段，压缩是目的，按照习惯统一使用视频编码这一表述，类似地，使用编码标准、编码方案、编码效率等表述；2) 为了直接体现出编码标准的发展历程，不使用高级视频编码 (Advanced Video Coding, AVC)、高性能视频编码 (High Efficiency Video Coding, HEVC) 和多功能视频编码 (Versatile Video Coding, VVC) 的表述，统一使用 H.264、H.265 和 H.266，两类表述在含义上并无差异；3) 由联合专家组维护的各标准参考模型随着版本迭代可能存在不同的命名，例如 H.266 的参考模型在早期曾称为 JEM(Joint Exploration Model)，本文中统一使用以下命名：用于 H.264 的 JM(Joint Test Model) 模型^[23]、用于 H.265 的 HM(HEVC Test Model) 模型^[24] 以及用于 H.266 的 VTM(VVC Test Model) 模型^[25]。

2.1 视频编码技术发展历史

应用广泛的 H.265 与确定最终标准不久的 H.266 视频编码标准，是数字视频编码技术 40 年的学术研究和 30 年的标准化的成果，代表着各时期最先进的视频编码技术^[26]。在视频编码技术发展的早期 (H.264 制定以前)，国际上

存在 2 套应用广泛的标准，分别是国际电信联盟-电信标准化部门 (International Telecommunication Union-Telecommunication Standardization Sector, ITU-T) 制定的 H.26X 系列，与国际标准化组织 (International Organization for Standardization, ISO) / 国际电工委员会 (International Electrotechnical Commission, IEC) 制定的 MPEG 系列。前者广泛应用于网络视频通信，后者主要应用于视频存储和广播流媒体。H.264 以后的编码标准则是由 ITU-T 派出的视频编码专家组 (Video Coding Experts Group, VCEG) 和 ISO/IEC 派出的动态图像专家组 (Moving Pictures Experts Group, MPEG) 共同征集提案制定的。该联合小组在 H.264 时期称为联合视频组 (Joint Video Team, JVT)；在 H.265 时期称为视频编码联合合作组 (Joint Collaboration Team on Video Coding, JVT-VC)；在 H.266 早期称为联合视频探索组 (Joint Video Exploration Team, JVET)，现在称为联合专家组 (Joint Video Experts Team, JVET)。

H.26X 标准自问世至今在整体框架上并未变动，如图 2.1 所示，包含帧内/帧间预测、变换量化、熵编码、环路后处理以及率-失真优化模块^[27]。但各代标准在上述 5 大模块中均加入了新的编码工具，使编码效率呈现出每代提高约 50% 的趋势^[1, 2]。例如帧内预测模式自 H.264 的 9 种发展到了 H.266 的 67 种；变换量化过程经过拆分重组减小了计算复杂度；环路后处理方面从 H.265 开始引入了新的自适应补偿机制，大幅提高了图像质量。

2.2 预测编码技术

预测编码是 H.26X 编码标准中的重要组成部分之一。视频信号表现出很强的空间相关性和时间相关性，即在空间上相邻像素点之间的亮度或色差值相近，在时间上同一区域相邻两帧之间的像素点亮度或色差值相近^[28]。使用帧内和帧间预测技术可以准确地对像素值进行预测，进而编码预测值与原始像素值的残差，极大地减少视频信号的时空冗余。预测编码的基本流程如图 2.2 所示。对于待预测像素 $x(n)$ ，首先利用已重建的像素结合特定的预测模式得到预测值 $p(n)$ ，然后计算预测残差 $e(n)$ ，最后对 $e(n)$ 进行变换、量化、熵编码，同时使用去量化、逆变换后的重建值 $e'(n)$ 与预测值 $p(n)$ 重建出后续待预测像素的参考值 $x'(n)$ 。

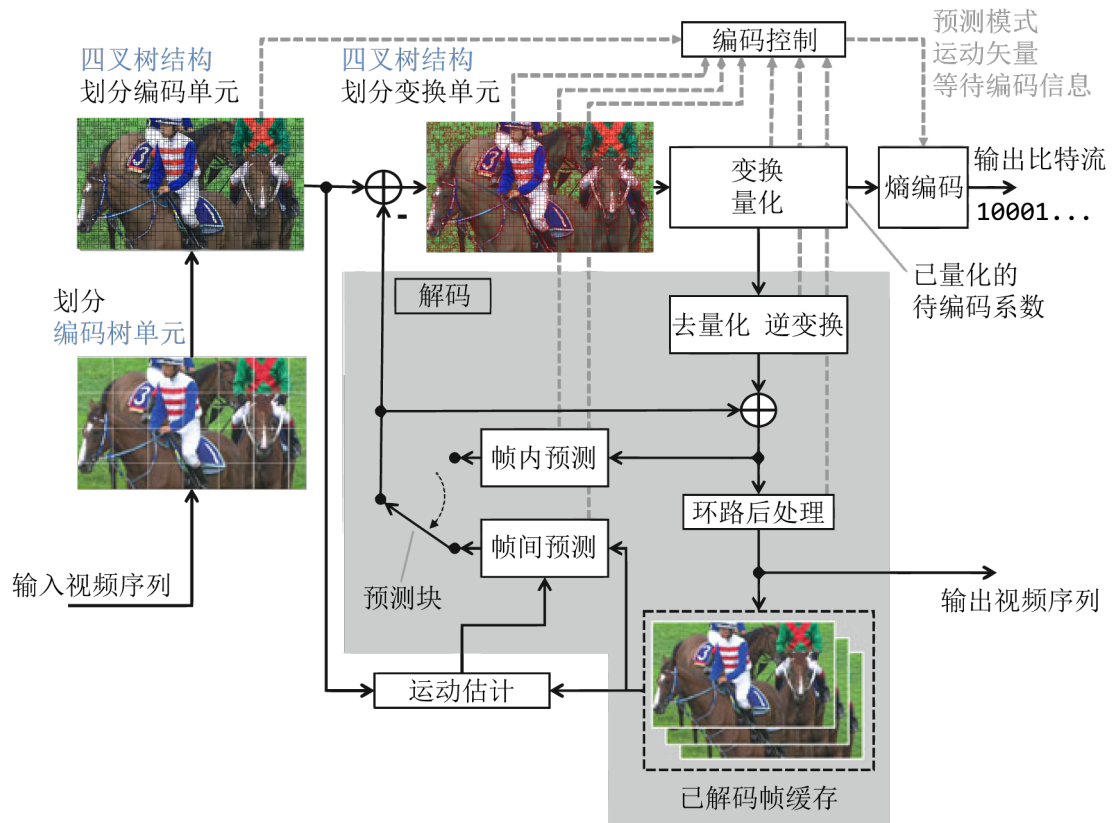


图 2.1 H.26X 标准视频编码框架

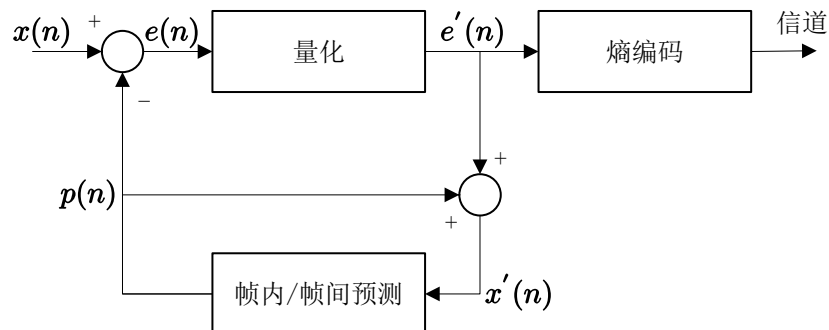


图 2.2 H.26X 预测编码的基本流程

根据参考点种类的不同，可将预测编码分为以下两类，在后文将进一步介绍：

1. 帧内预测，即利用待预测点空间上临近的已重建像素点进行预测的方法；
2. 帧间预测，即利用待预测点时间上临近的几帧（可以是之前也可以是之后）在相同位置附近的已重建像素点进行预测的方法。

2.2.1 帧内预测

帧内预测指利用视频、图像的空间相关性，使用已编码重建的像素点预测当前像素，预测完成后的像素经过重建又作为后续待预测像素的参考点^[29]。设待预测点为 $f(x, y)$ ， (x, y) 为待预测点坐标，该点的预测值由已重建的参考点 $\tilde{f}(k, l)$ 进行预测给出：

$$\hat{f}(x, y) = \sum_{(k, l) \in Z} a_{k, l} \tilde{f}(k, l) \quad (2.1)$$

其中 $a_{k, l}$ 表示特定预测方式，多表现为投影与插值的组合， Z 为参考点所在区域。后续变换量化、熵编码的对象均为预测值与原始值的误差，即预测残差 $e(x, y)$ ：

$$e(x, y) = f(x, y) - \hat{f}(x, y) \quad (2.2)$$

H.26X 系列标准中制定了多种帧内预测模式，包括平滑预测(DC 模式、Planar 模式)和方向预测两大类。随着编码标准的发展，预测模式的数量不断增多，H.264 仅使用 9 种模式^[3]，H.265 增加到 35 种^[2]，H.266 达到了 67 种^[1]，显然模式越多越有可能得到精确的预测值，最后体现为更高的编码效率^[30]。标准中，任意一种预测模式都是以相邻块的边界上的重建像素为参考点进行的，该方案使得每个预测单元 (Prediction Unit, PU) 均能自适应地选择最佳的预测模式。如图 2.3 所示，左侧衣物上存在明显的方向性条纹，适合使用方向预测模式，而右侧的衣物呈现出平坦的特性，适合使用 DC 或 Planar 模式。关于帧内预测的更详细的分析将在 3.2.1 节进行。



图 2.3 选择不同的帧内预测模式

2.2.2 帧间预测

帧间预测指利用视频的时间相关性，使用临近帧（可以是之前也可以是之后）相同位置附近的像素点预测当前像素。以像素点为单位进行帧间预测显然是不可行的，因此与帧内预测类似，帧间预测也以块为单位，如图 2.4 所示。其预测策略是针对每个待预测块，从已编码的其他帧中寻找与其匹配的块作为参考块，计算参考块到待预测块的位移信息。该过程即为运动估计，其中得到的位移信息称为运动矢量，预测块与参考块的差值称为运动预测残差。

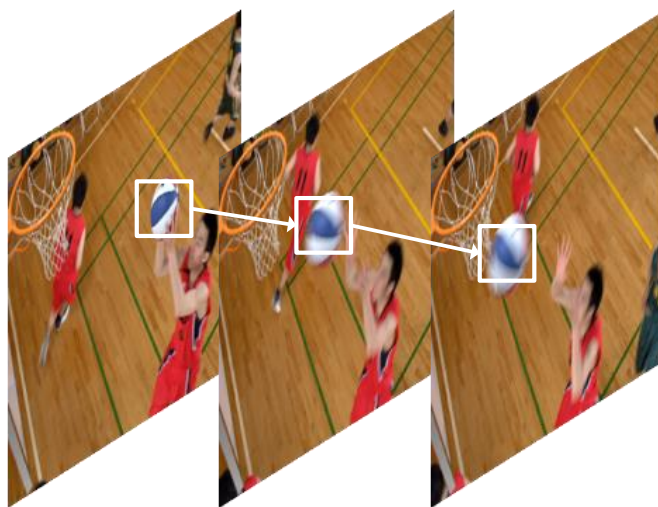


图 2.4 H.26X 帧间预测

帧间预测中的核心技术是运动搜索算法，运动搜索是 H.26X 编码中最耗时但也是影响编码效率最大的一个过程。H.26X 标准中并未指定搜索方法而只在软件模型中给出了参考算法，例如 HM 模型中给出了全搜索算法和 TZSearch 算法^[31]。全搜索算法指对于每一个待预测块，遍历搜索窗口中所有可能的运动矢量取其中运动预测残差最小的一个；TZSearch 算法属于快速算法的一种，其按照固定的菱形搜索模板或正方形搜索模板以 2 的整数次幂递增步长进行运动搜索，大幅减少搜索量的同时保持了较好的搜索性能（不容易陷入局部最优）。

2.3 变换与量化技术

H.26X 编码标准中，通过帧内和帧间预测获得了相对原始像素值能量大幅降低的预测残差，且预测编码并未引入失真。后续的变化与量化进一步地对残

差进行处理，使能量集中得到稀疏的待编码数据，同时也是编码过程中真正引入失真的步骤。

2.3.1 变换编码

变换编码技术指将预测残差（某些特殊情况下是原始像素值）从空间域通过离散余弦变换 (Discrete Cosine Transform, DCT) 或离散正弦变换 (Discrete Sine Transform, DST) 转换到频域，以变换域系数来表示数据。本节以 DCT 为例简单介绍 H.26X 标准中变换编码的应用场景及效果。

DCT 是一种在频域对数字信号进行分析的工具，在数学上共有 8 种形式^[32]，图像、音视频领域中常用的是 II 类 DCT^[33]：

$$X(k) = \sqrt{\frac{2}{N}} \varepsilon_k \sum_{n=0}^{N-1} x(n) \cos \left[\frac{k(2n+1)\pi}{2N} \right], \quad k = 0, 1, \dots, N-1$$

$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 1, & o.w \end{cases} \quad (2.3)$$

当 $k = 0$ 时式中不再有余弦项，即 $X(0) \propto$ 信号 $x(n)$ 的均值，此时称 $X(0)$ 为信号 $x(n)$ 的直流 (Direct Current, DC) 分量；反之，当 $k > 0$ 时， k 越大式中的余弦信号频率越高，此时 $X(k)$ 反映了信号 $x(n)$ 在不同频率分量上的分布多寡，称为信号 $x(n)$ 的交流 (Alternate Current, AC) 分量。

以 H.265 标准测试序列 “BasketballPass” 中第一帧的前 8 个数据 $x(n)$ 为例计算其对应的变换域系数 $X(k)$ ：

$$x(n) : 127, 129, 130, 128, 125, 122, 123, 124$$

当 $k = 0$ 时：

$$X(0) = \frac{1}{2\sqrt{2}}(127 + 129 + 130 + 128 + 125 + 122 + 123 + 124) \cos(0) \approx 356$$

同理根据式 2.3 计算所有的 $X(k)$ 得到：

$$X(k) : 356, 6, -1, -4, 0, 0, 0, 0$$

从上述示例中可以看出，当原始信号 $x(n)$ 的数值分布比较均匀即各数值之间相关性强时，其变换域的系数 $X(k)$ 将呈现出数值集中分布在 DC 和低频部分的稀疏状态。显然，自然图像中的像素幅值或其预测残差都很容易表现出强相关性，因此将待编码数据转换到变换域十分有利于后续的熵编码。另外，上述示例是 II 类 DCT 的一维形式，实际上图像、视频编码中使用的是其二维形式：

$$X(k, l) = C(k)C(l) \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \cos \left[\frac{(2m+1)k\pi}{2N} \right] \cos \left[\frac{(2n+1)l\pi}{2N} \right],$$

$$C(k) = C(l) = \begin{cases} \sqrt{\frac{1}{N}}, & k, l = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq k, l \leq N-1 \end{cases} \quad (2.4)$$

变换编码技术在 H.26X 中存在较多研究和优化内容不再赘述，但以下几点值得一提：

1. DCT 去相关性的性能随着变换大小的增大而增强，但增强的比率迅速减缓，因此考虑到计算复杂度和编码性能的折衷，较常采用的是 4×4 与 8×8 DCT，很少用到 16×16 及更大的变换；
2. 同样出于计算复杂度的考虑，从 H.264 开始标准中的变换一般拆分为两部分进行，即与浮点运算无关的整数变换部分及分离出的比例缩放系数^[34]，后者将与量化合并为一次计算，提高了计算效率；
3. 根据统计结果，对 4×4 大小的亮度块预测残差使用 DST 能提高约 0.8% 的编码效率而计算复杂度不变^[35]，因此从 H.265 开始将 DST 纳入考虑；
4. 哈达玛变换与 DCT 处理后的变换残差绝对值之和 (Sum of Absolute Transformed Difference, SATD) 相近，但哈达玛变换计算复杂度极低，因此 H.265 将哈达玛变换与 SATD 作为快速预测模式选取的工具。

2.3.2 量化

从 H.264 将变换拆分为整数变换和比例缩放两部分以后，H.26X 标准中产生失真的原因集中到了量化上（图 2.5）。H.26X 标准中仅规定了量化过程的解码方法，将编码端量化方式的选择策略交给编码器自行设计和优化，H.265 中可选的量化方法有传统标量量化、自适应量化 (Adaptive Quantization) 和率-失真优

化量化 (Rate-Distortion Optimization Quantization, RDOQ)，本节以标量量化为例介绍量化的过程和作用。

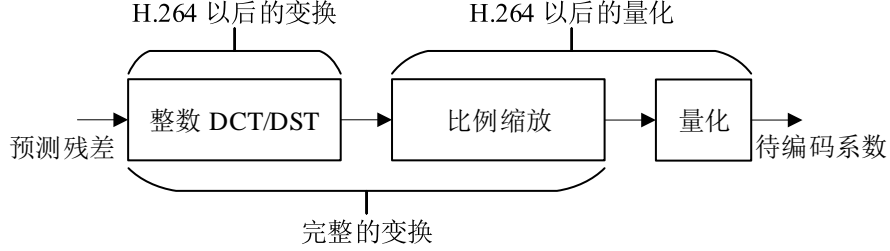


图 2.5 H.26X 中变换量化过程的拆分与合并

由于经过变换后的数据动态范围很大，需要通过量化将数据的取值范围压缩才能得到理想的编码效率。经过压缩必定会产生数据精度损失，因此量化是音视频编码中产生失真的根本原因，也是控制编码结果码率的最直接因素。标量量化的过程如下式：

$$l_i = \left\lfloor \frac{c_i}{Q_{step}} + f \right\rfloor \quad (2.5)$$

其中 c_i 表示 2.3.1 节所述的变换域系数， Q_{step} 为量化步长， l_i 为量化器的输出， f 用于控制舍入。H.265 中可选 52 个量化步长，量化步长小代表数据精度损失小即失真小但同时编码效率低。需要注意量化并非简单地对输入的变换域系数平等地做缩放。基于人类视觉系统 (Human Visual System, HVS) 研究中发现的人眼对亮度、低频分量敏感，对色差、高频分量较不敏感的生物特性^[36]，各类图像视频编码方案中均存在量化矩阵的概念。即对不同频率分量的变换域系数做相应的缩放，例如对低频分量做更轻微的量化以保证良好的视觉效果。值得一提的是相对于学术界关注峰值信噪比 (Peak Signal-to-Noise Ratio, PSNR)、结构相似度 (Structural Similarity Index, SSIM)^[37] 等客观图像质量评价指标，工业界更关心的是平均主观得分 (Mean Opinion Score, MOS)，即实验者根据自身的观看体验给出的图像视频质量分数，工业界中甚至有关于“让人感觉舒服，错以为图像质量好的失真”的研究，此为另外的研究方向不再展开。总之量化是控制视频质量的关键步骤，量化系数的选择策略尤为重要。

2.4 环路后处理技术

H.26X 标准中的预测、变换和量化等编码工具均是基于块结构设计的，相邻块之间由于预测模式突变、量化值突变和不属于同一个变换单元 (Transform Unit, TU) 等多方面原因，产生了不可避免的数值不连续性。人眼对此类不连续性较敏感，导致视频编码后的 MOS 大打折扣。对此 H.26X 对重构像素进行后处理作为应对，因该处理处于预测-变换-量化-重建环路中而称为环路后处理（图 2.1），包含去方块滤波 (Deblocking Filter) 和 H.265 中新增的样点自适应补偿 (Sample Adaptive Offset, SAO)。

2.4.1 去方块滤波

去方块滤波技术从 H.264 开始引入，其作用是消除相邻编码块间的不连续性，一方面可以直接优化视频的视觉效果提高 MOS，另一方面经过去方块滤波的数据更适合作为后续编码单元 (Coding Unit, CU) 的参考点，使预测值更加精准，间接提高了 PSNR 和 SSIM 等客观指标^[38]。需要注意的是并非所有的编码块边缘都需要进行去方块滤波，图像中纹理信息丰富的编码块之间必然存在较强的不连续性，此时进行去方块滤波不仅无法提升视觉质量还会导致图像中的细节丢失。因此去方块滤波的基本原则是对数值平滑的编码块之间存在的不连续性做强滤波，对本身数值波动较大的编码块之间存在的不连续性不做滤波或只做弱滤波，如图 2.6(a)(b) 所示。因此，去方块滤波在具体实现中分为了滤波决策与执行滤波两个步骤，执行滤波也分为亮度分量强滤波、亮度分量弱滤波和色差分量滤波 3 类，具体不再展开。

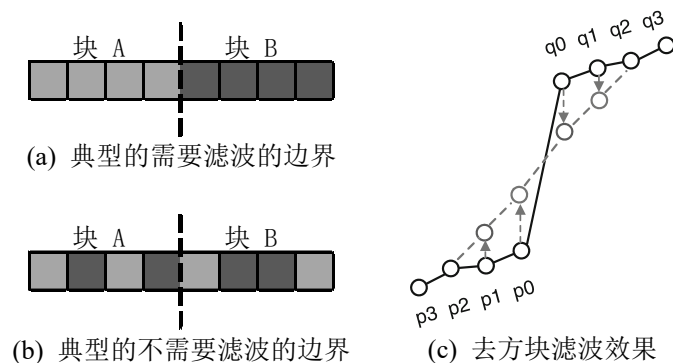


图 2.6 去方块滤波应用场景及效果

2.4.2 样点自适应补偿

SAO 是 H.265 开始引入的环路后处理技术。H.26X 标准中为了得到稀疏的待编码系数进行了量化，量化的对象是变换域系数，而量化不可避免地会产生数据失真，变换域系数的数据失真通过逆变换回到空间域则表现为重建值在原始值的附近上下波动^[39]，如图 2.7 所示。

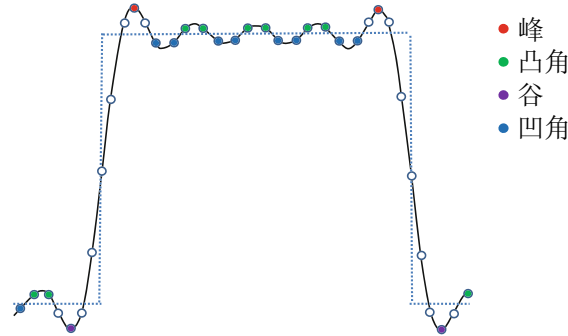


图 2.7 量化失真引起的振铃效应

这种波动在数学上称为吉布斯现象，在图像视频编码领域更形象地称为振铃效应。振铃效应的成因是变换域的高频分量系数产生的失真过大，因此“振铃”在图像细节丰富的区域尤为明显。为了减轻振铃效应带来的图像质量损失，H.265 标准应用了一种十分直接的处理方案：对重建数据进行后处理，在“振铃”的“谷”添加正补偿，在“振铃”的“峰”添加负补偿。在具体的补偿方案中又分为边界补偿 (Edge Offset, EO) 和边带补偿 (Band Offset, BO)，本节仅以 EO 为例介绍 SAO 的处理过程。

边界补偿采用对每个像素点单独进行分类和补偿的策略，出于性能与复杂度的折衷，分类过程仅涉及待处理像素及其相邻的 2 个像素。H.265 中需要通过 SAO 进行补偿的像素点共有 4 类，如图 2.7 与 2.8 所示，图中 c 表示当前待处理像素， a 和 b 为其相邻的 2 个像素。

对于 1 类“谷”和 2 类“凹角”进行正补偿，将使补偿点附近更加平滑；对于 3 类“凸角”和 4 类“峰”进行负补偿，同样使补偿点附近平滑。对于不属于上述 4 类的像素点将归类为 0 类不进行补偿。在预先定义了补偿的正负性后，编码器仅须记录绝对补偿值，解码器便可通过同样的分类策略判断补偿的正负性并进行补偿。因此 SAO 在保证了仅传输极少量数据的情况下对图像视频质量

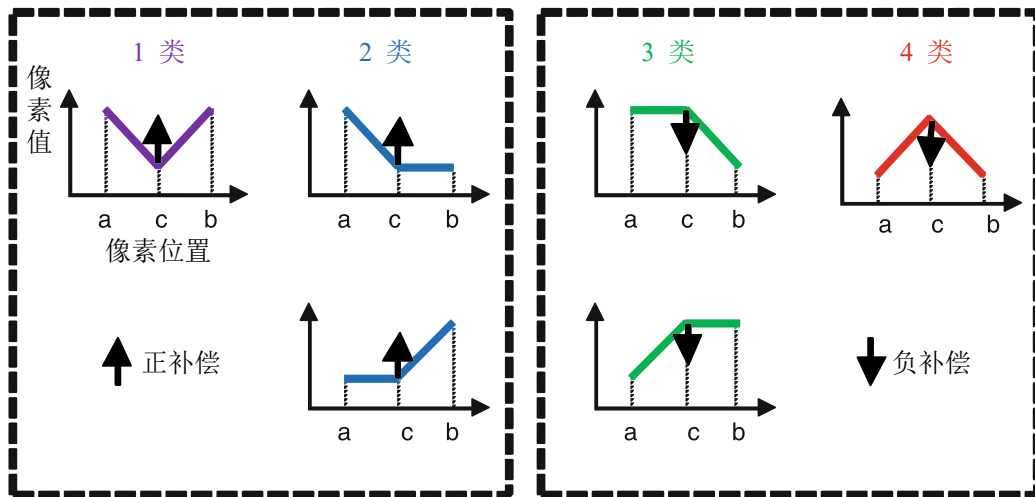


图 2.8 SAO 边界补偿分类

做出了有效的优化。

2.5 熵编码技术

熵编码是 H.26X 编码标准的最后一步，其将各类待编码的数据转换为具有特定语法语义的二进制比特流（见图 2.1）。因其依据信息熵理论进行优化而称熵编码，熵编码可进一步利用待编码数据的统计特性消除冗余，是提高视频编码效率的重要工具^[40]。

2.5.1 变长编码在视频编码中的应用

对不同的输入符号使用不同长度的码字进行编码称为变长编码。为了得到更高的编码效率，需要将短码字分配给出现频率高的符号，反之将长码字分配给几乎不出现的符号以使平均码字长度最短。值得一提的是，变长编码中各码字必须符合“任意一个码字的尾缀不能是其他任意码字的前缀”的要求，否则变长码将不可译（在不添加码字间分隔符的情况下），符合该要求的一组码字一定是“唯一可译码”。

经典的哈夫曼码是一种最佳变长码，然而在解哈夫曼码时解码器必须依靠编码端提供的哈夫曼树结构，因此增加了编码、传输的数据量，另外解哈夫曼码时需要一次性读入大量码流在树中搜索匹配码字，复杂度过高。因此早期的图像编码中使用了固定结构的范式哈夫曼编码^[41]，如 JPEG。另一类方案是使用

有固定规则的编码方法，如 H.26X 中使用的指数哥伦布码 (Exponential Golomb Code)。指数哥伦布码的构建过程如下，其中 k 表示阶数：

1. 将待编码数值表示为 2 进制，分离出尾部 k 位，剩余部分加 1；
2. 计算剩余部分有几位，减去 1 后得到的结果为码字前缀 0 的个数；
3. 将 1 中分离出的尾部 k 位重新补回。

表 2.1 给出了 0 阶和 1 阶的部分指数哥伦布码字，较小的数值被分配到了更短的码字。在 H.265 中指数哥伦布码多用于各类参数信息的编码，如视频参数集 (Video Parameter Set, VPS) 和序列参数集 (Sequence Parameter Set, SPS) 等均使用 0 阶指数哥伦布码进行编码。

表 2.1 k 阶指数哥伦布码

阶数	码字	编码范围
$k=0$	1	0
	0 1 x_0	1...2
	0 0 1 $x_1 x_0$	3...6
	0 0 0 1 $x_2 x_1 x_0$	7...14

$k=1$	1 x_0	0...1
	0 1 $x_1 x_0$	2...5
	0 0 1 $x_2 x_1 x_0$	6...13
	0 0 0 1 $x_3 x_2 x_1 x_0$	14...29

2.5.2 算术编码在视频编码中的应用

尽管变长编码提高了编码效率，但对于每个输入符号使用变长编码至少都要 1 比特表示。算术编码解除了这种限制，其本质是对整个输入序列分配 1 个码字，平均来看可能实现用少于 1 比特的数据量来表示一个符号。算术编码的过程简述如下：

1. 统计输入序列中所有符号出现的概率；

2. 根据编码符号对应的概率将区间 $[0, 1)$ 进行分配, 区间的大小正比于概率, 某符号分配到区间 $[L, H)$;
3. 从初始区间 $[0, 1)$ 开始, 设置:

$$low = 0, \quad high = 1$$

4. 按顺序读入待编码符号, 根据其分配到的 $[L, H)$ 对区间进行更新:

$$low = low + (high - low) * L$$

$$high = low + (high - low) * H$$

5. 最后得到区间 $[low, high)$, 选择下界 low 作为该序列的算术编码码字。

在上述过程开始前确定了各符号的概率进行编码的称为静态算术编码, 显然这在视频编码中是无法实现的。因此 H.26X 在进行算数编码的过程中对信源符号的概率不断建模、更新, 该技术发展到 H.265 成为了成熟的基于上下文的自适应二进制算术编码 (Context-Based Adaptive Binary Arithmetic Coding, CABAC)。其核心是编码过程中维护了一个设计精良的上下文模型, 每个符号的编码均与以前的编码结果有关, 根据输入待编码符号流的统计特性自适应地分配码字。H.265 中对待编码系数有关的大量语法元素采用的都是 CABAC, 如变换跳过标志、编码块标志以及帧内/帧间标志等。

2.5.3 模式依赖的编码顺序

量化后的待编码系数普遍具有幅值小且稀疏的特性, 有效地利用这一特性可以使 CABAC 的上下文建模更加高效。H.26X 标准中的编码工具均是基于块设计的, 因此在最后一步的熵编码之前, 如何将二维分布的待编码数据按特定的顺序扫描送入熵编码器在标准中也进行了规定。扫描的顺序需要进一步考虑数值的空间分布特征, 将幅值相近的数据集中在一起扫描有利于上下文建模, 从而提高编码效率。

H.265 中规定扫描的过程分为 4×4 子块间的扫描与 4×4 子块内的扫描, 块间与块内扫描递归进行。具体的扫描顺序又分为水平扫描、垂直扫描和对角扫描, 制定该 3 类扫描顺序的根据是预测模式与系数分布特征间的联系: 一个采

用（接近）垂直预测模式得到的待编码块，其能量很可能集中在前几行，因此水平扫描对（接近）垂直预测模式的块有利，同理垂直扫描对水平预测模式的块有利^[42]，对此 3.2.1 节将做进一步分析。图 2.9 给出了具体的 3 类扫描顺序示例。

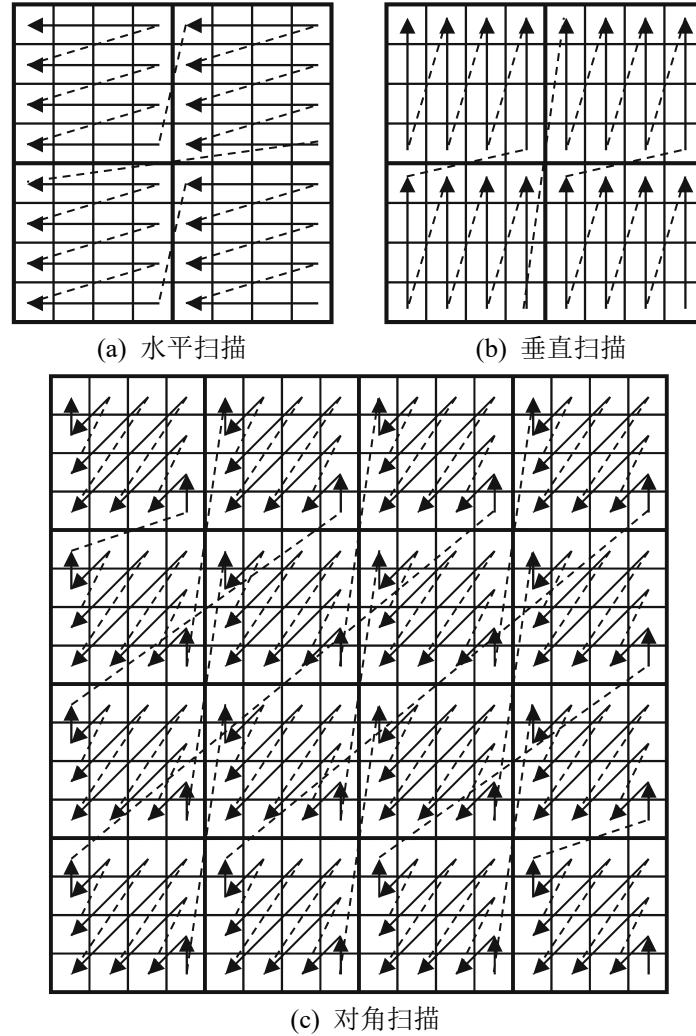


图 2.9 H.265 待编码系数扫描顺序

与传统图像编码如 JPEG 不同，H.265 的扫描始终从最后一个系数开始到 DC 系数结束，与 Zig-Zag 扫描恰好相反。值得一提的是上述扫描的单位固定为 4×4 系数块，不论该块在预测、变换时是何种尺寸，这一限定有利于具体实现时的模块化操作。

2.6 率-失真优化技术

视频编码标准中的大部分内容是指定各类语法和语义，只要是符合标准语法的码流都能称为符合某视频编码标准。例如在使用 H.265 进行编码时，若将最小 PU 限定为 32×32 (HM 默认配置是 4×4)，编码效率与失真度可以预见甚至不如简单的 M-JPEG(Motion Joint Photographic Experts Group)。同时，为了适应不同的场景需求，视频编码标准中存在大量可配置参数用于调控编码效率、失真度和编码复杂度，因此，如何在标准的编码框架下得到不同场景下最优的编码效率与失真度的平衡，是编码标准中未规定的算法研究的核心内容，寻找折衷点的过程被称为率-失真优化^[43]。

由于 H.26X 标准中未规定率-失真优化的具体技术细节，因此编码器可自定义、优化参数选择的过程。率-失真优化方法可在图像组 (Group of Pictures, GOP) 层、片 (Slice) 层、编码树单元 (Coding Tree Unit, CTU) 层、CU 层和 PU 层应用。

- GOP 的率-失真优化

由于帧间预测的存在，各帧图像的编码效率和失真度存在相互依赖的关系，因此要得到最优的一组编码配置参数只能通过穷举的方法。GOP 层的穷举复杂度最高无法在实际编码中应用，因此 GOP 层的率-失真优化简化为独立确定各帧图像的最优编码配置参数。

- Slice 的率-失真优化

Slice 是 H.265 中相对独立的编码单元，一个 Slice 包含多个相邻的 CTU，帧内/帧间预测时不能使用当前 Slice 外的参考点。与 GOP 类似，当前 Slice 级别即各 CTU 之间的率-失真相关性还未被系统地考虑，仍按照独立确定各 CTU 的编码参数进行。

- CTU 的率-失真优化

CTU 是 H.265 中基本的编码单元，CTU 一般会往下切分为多个 CU，CU 可切分为多个使用不同预测策略的 PU，PU 可选择适合自身的预测模式。由此 CTU 的率-失真优化分为 CU/PU 切分和 PU 的模式选择。类似地为了降低复杂度，当前 CU 的优化结束后不再考虑其对后续 CU 的影响。

- CU 的率-失真优化

CU 层的优化任务是确定 PU 的预测模式和变换单元 (Transform Unit, TU) 的分割。目前 CU 级别的穷举复杂度可被接受, 因此部分编码器在 CU 的优化过程中遍历候选集找出率-失真代价最小的一组参数。

- PU 的率-失真优化

PU 层的优化任务是确定具体的预测模式。以帧内为例, H.265 提供了 35 种可能的预测模式, 可通过遍历的方法计算所有率-失真代价得到最优的预测模式。部分编码器如 HM 应用了哈达玛变换来加速这一过程 (见 2.3.1 节)。

2.7 软、硬件视频编码项目

视频编解码这一涉及到成像系统、生物信息、数学、统计学、计算机以及集成电路的庞大学科, 如果没有一个系统的项目让各学科的研究人员参与进来, 而是由单一的某一组织闭门造车是无法得到长足发展的。本小节总结了目前常用的软、硬件视频编码项目, 本课题就是基于其中部分项目开展的。

2.7.1 软件视频编码项目

- JM^[23]

JM 是制定 H.264 标准的 JVT 负责维护的 H.264 编解码参考软件。JM 是学术界常用的编码器, 其实现了 H.264 的所有特性, 内部没有使用多线程或进行汇编层级的优化, 因此其编解码速度较慢无法实现实时编码, 多用于研究和测试标准性能。

- x264

x264 是由 VideoLAN 维护的 H.264 编码器。x264 是工业界最常用的 H.264 编码器, 其内部集成了丰富的快速、并行算法, 使得编解码效率得到了极大提升, 被应用在各类音视频处理框架中。目前最完善的多媒体处理软件 FFmpeg 在 H.264 部分就内嵌了 x264 用于编码。

- HM^[24]

HM 是制定 H.265 标准的 JCT-VC 负责维护的 H.265 编解码参考软件。HM

是学术界在研究 H.265 和下一代视频编码标准时常用的编码器，其实现了 H.265 的所有特性（包括部分未写入标准的会议提案），内部没有使用多线程或进行汇编层级的优化，因此其编解码速度较慢无法实现实时编码，多用于研究和测试标准性能。

- VTM^[25]

VTM 是制定 H.266 标准的 JVET 负责维护的 H.266 编解码参考软件，随着 H.266 编码最终标准的确定仍在快速迭代中，在 H.266 标准探索的初期曾称为 JEM。VTM 是联合小组在制定下一代视频编码标准 H.266 时使用的编码器，用于测试所征集提案的性能。

- VVenC^[44]

VVenC(Fraunhofer Versatile Video Encoder)是在考虑到 VTM 过于庞大冗杂的编码工具后，根据多种复杂度优化提案在编码性能和编码时间之间取折衷的 H.266 编码器。维护者来自 VTM 开发的主要参与方、H.266 标准制定的重要参与者 Fraunhofer HHI。

2.7.2 硬件视频编码项目

- OpenASIC Encoder RTL IP

OpenASIC H.264/H.265 Encoder RTL IP 是由复旦大学微电子学院视频图像处理科研团队发布的 H.264/H.265 RTL IP，由芯片社区 OpenASIC 共同维护。其中 H.264 Encoder RTL IP 实现了 H.264 标准编码流程，最大支持 $1920 \times 1088 @ 30\text{fps}$ 实时编码，已被部分公司、研究所研究使用；H.265 Encoder RTL IP 发布了 2 个版本，实现了帧内/帧间预测、熵编码、环路后处理以及基础的码率控制，满足 $0.13\mu\text{m}$ 工艺 400MHz 无时序违例综合，最大支持 $4\text{K} @ 30\text{fps}$ 实时编码。

2.8 本章小结

本章简述了视频编码技术和标准的发展史，介绍了视频编码中的 5 项核心技术：帧内/帧间预测、变换量化、环路后处理、熵编码以及率-失真优化，为后

文探索编码性能优化的可行性打下基础。下一章开始正式介绍本课题进行的算法优化及硬件实现。

第三章 帧内无损编码优化算法

本章从帧内无损编码的可优化方向的分析开始，引出本研究课题中提出的 4 个优化算法：1) L 形迭代预测算法 L-IP；2) L 形分块算法 L-BP；3) 残差中值边缘检测算法 R-MED。4) 联合算法 L-BPIP；在具体描述所提出算法之前均先详细地分析了标准规定或 HM 参考软件中的实现方案，以与所提算法形成对比。最后在对各算法性能单独测试的基础上，给出联合算法及其性能测试，以证明所提算法的有效性。

3.1 帧内无损编码的可优化方向

针对 H.26X 标准，可考虑优化多种编码工具使最终的编码效率提高，当将优化对象聚焦到帧内无损编码时，涉及到的可优化模块有：帧内预测模块、熵编码模块以及贯穿编码过程的率-失真优化模块。本课题中提出的 3 个优化算法是在深入分析帧内预测过程及率-失真优化过程后提出的。

对帧内预测过程进行分析。H.26X 的帧内预测是基于块结构进行的，在 H.265 中最小的预测单元大小为 4×4 ，但针对视频图像中纹理丰富的区域 4×4 大小仍显得有些不足。出于计算复杂度的考虑，H.26X 标准暂时难以跳出块预测的框架，因此寻找更为精准的预测方式是一个可行的优化方向。块结构的预测准确度仍不够高的原因在于部分被预测点距离参考点的空间距离过大，基于此本课题中提出了 L 形迭代预测算法，在保持整体帧内预测流程仍是块结构的情况下将所有待预测点与参考点的距离缩短到了 1 单位。

对率-失真优化中的编码块划分过程进行分析。为了同时在图像中的平缓区域和细节丰富区域得到良好的编码效率，H.26X 在 RDO 过程中会进行搜索，自适应地确定编码块的尺寸，即在平缓区域用较大的块进行编码，在细节丰富的区域划分出多而小的编码块进行编码。在 H.265 中，帧内预测部分采用基于四叉树的循环分层结构，即对于一个 $2N \times 2N$ 大小的 CU，PU 可选择的模式只有两种：划分为 4 个 $N \times N$ 或保留 $2N \times 2N$ 。仅能 2 选 1 的划分模式难以满足图像视

频中复杂多变的纹理的编码需求，因此提供更多可选的划分模式是一个可行的优化方向。基于此本课题中提出了一种非对称的 L 形分块算法，使 RDO 过程可以搜索更多的分块可能性。

对无损模式下帧内预测的残差进行分析。H.26X 中基础的无损编码是通过简单地跳过变换、量化和环路后处理等可能引入数值失真的步骤实现的^[6]。由于缺少了变换这一将数值能量集中的步骤，熵编码面临的压力剧增，既耗时又无法得到理想的编码效率。但也可以注意到此时待编码数据不再是变换域系数而是预测残差，因此其仍具有空间域的物理意义。通过进一步分析，发现无损模式下的待编码系数具有不同于自然图像的特殊的空间结构相关性，表现为含有丰富的边缘特征，因此利用待编码系数的特性，设法对其做二次处理使更有利于熵编码是一个可行的优化方向。基于此本课题中提出了残差中值边缘检测算法 R-MED，对无损模式下的待编码系数进行再处理得到一组能量大幅降低的新待编码系数，以降低熵编码的压力提高整体编码效率。

后文将先后对上述 3 个可优化方向做进一步的分析，并详细描述、测试所提出的算法。

3.2 帧内无损编码的预测过程优化

H.26X 的帧内无损编码方案中，预测过程的准确度直接决定了最终待编码系数的能量高低，因此设法优化预测过程使预测残差尽量降低是一个有效的优化方案。帧内预测以块为单位的特性导致其预测结果在距离参考点远的地方误差大，本节将针对该缺陷进行分析和优化。

3.2.1 帧内预测过程分析

以 H.265 中亮度分量的预测策略为例进行分析。每个帧内亮度预测单元均可选择 35 种预测模式进行处理，包括 33 种角度模式和 2 种平滑模式（DC 模式、Planar 模式）。角度模式进行预测的核心思想是图像中距离相近的像素点在数值上也大概率相近，这种数值相近的规律有可能发生在任意方向上，因此 H.265 规定了 33 个角度以适应图像中不同方向的纹理，如图 3.1 所示。显然可选择的

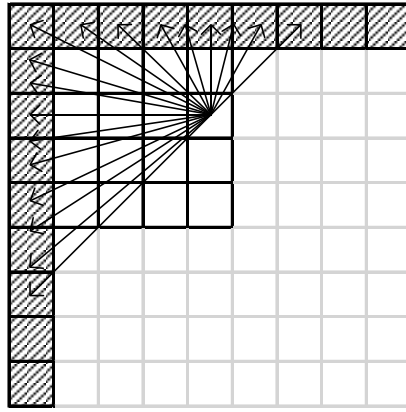


图 3.1 H.265 帧内预测角度模式

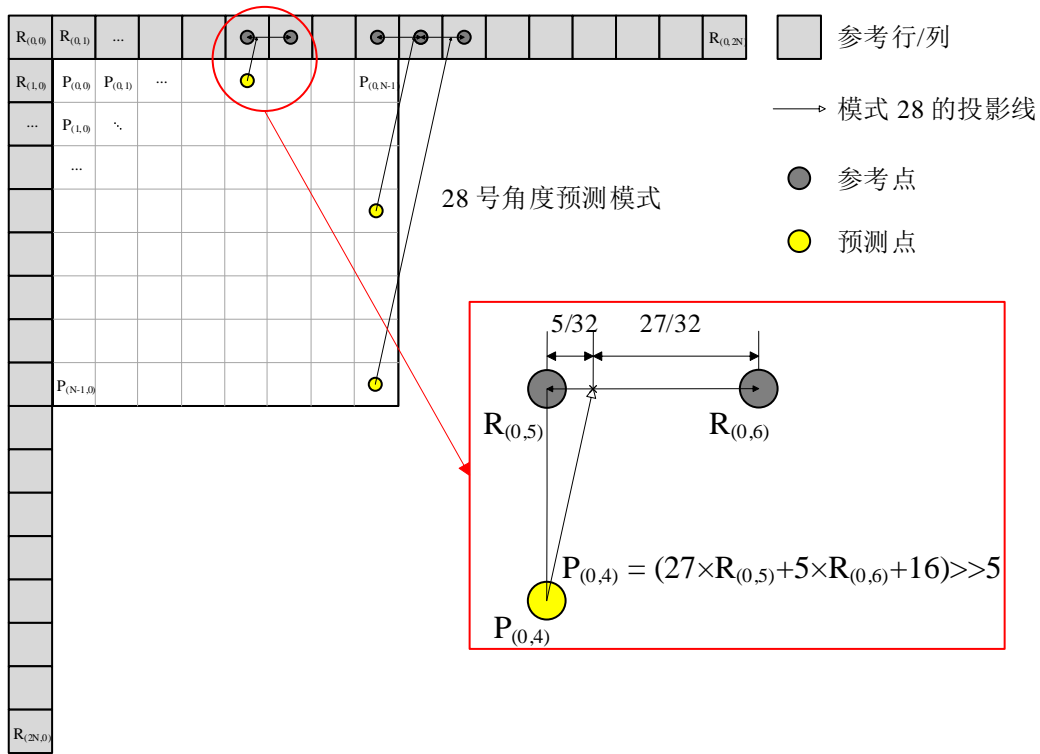


图 3.2 H.265 角度预测 1/32 精度投影

方向越多越有可能得到准确的预测值，例如 H.266 中将可选的方向增加到了 65 种。角度预测的过程可以直观地理解为：使用待预测块左侧一列与上侧一行的参考点进行插值与投影。值得一提的是确定投影角度、插值时的权重分配均是以 1/32 像素精度进行的。坐标 (x, y) 处的预测值 $P_{(x,y)}$ 的计算方式如下式：

$$P_{(x,y)} = ((32 - \omega)R_{(0,i)} + \omega R_{(0,i+1)} + 16) \gg 5 \quad (3.1)$$

其中 $R_{(0,i)}$ 与 $R_{(0,i+1)}$ 表示从待预测点投影到参考行、列时投影点左右的参考点数值, ω 用于控制权重分配, 精度为 $1/32$, 投影点距离 $R_{(0,i+1)}$ 越近 ω 越大。图 3.2 展示了 28 号角度预测模式下的投影与计算。

图 3.3 展示了一个 8×8 大小的 PU 经过 H.265 的 35 种帧内预测方式后分别得到的预测结果。参考点取值波动极大, 说明该待预测块附近很可能是纹理丰富的区域, 因此从不同角度进行投影后得到的预测结果也呈现出丰富的边缘信息。

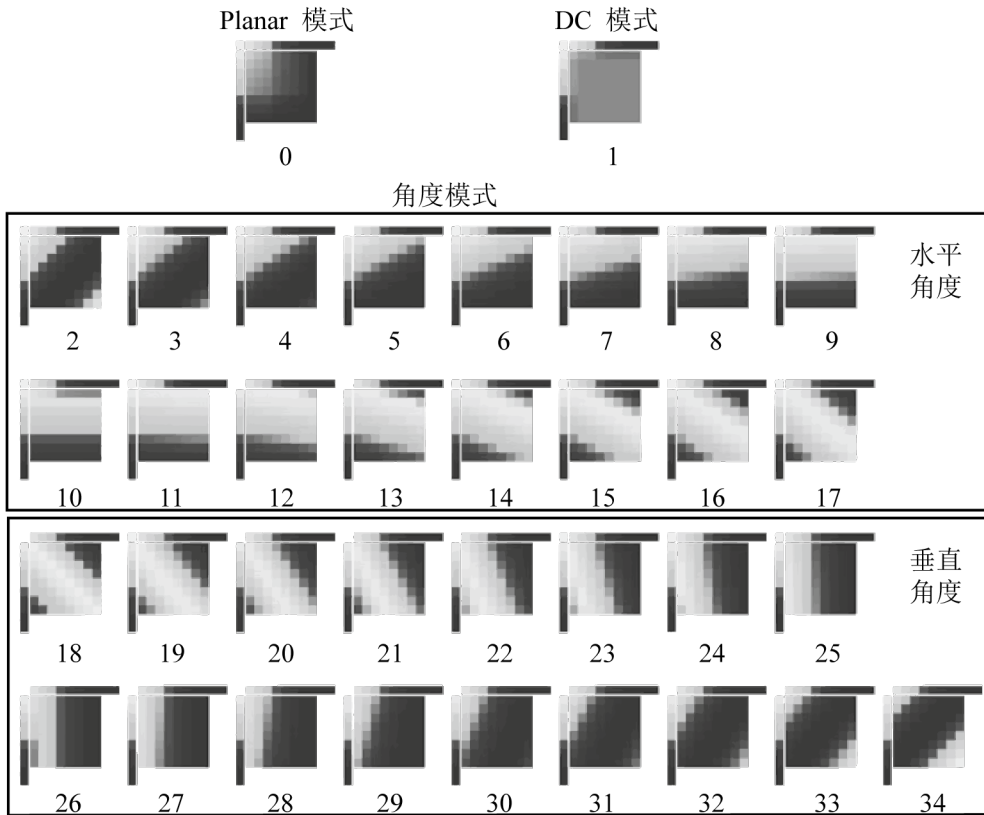


图 3.3 H.265 帧内预测结果示例

最后, 帧内预测过程还包含缺失参考点的填补方法、参考点的滤波策略和模式信息的编码方案等重要细节^[45], 但与将要提出的优化算法并无直接关系, 暂不详述。在充分理解 H.26X 帧内预测的物理意义及预测过程后, 下一节将分析其存在的缺陷并提出改进算法。

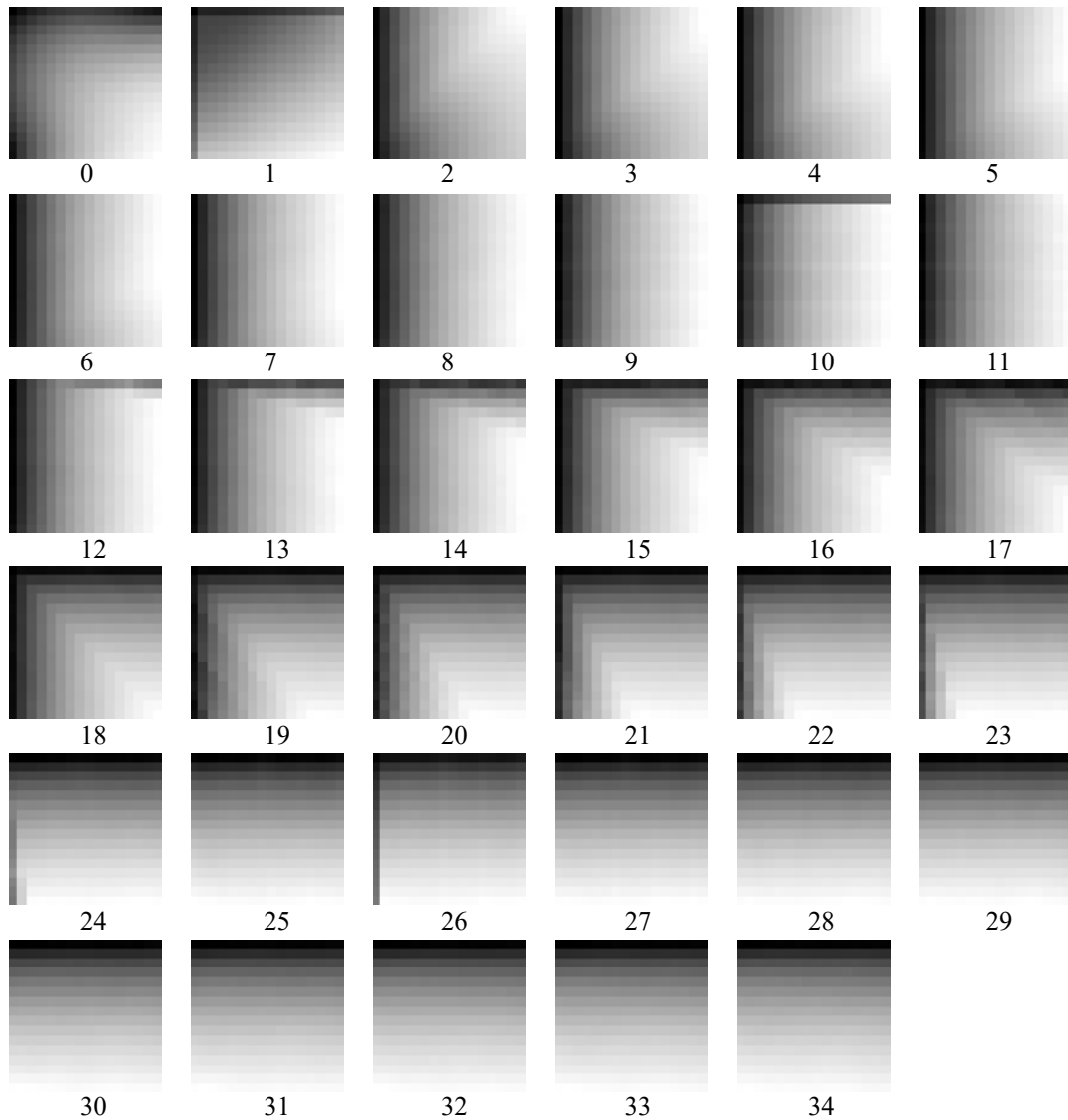


图 3.4 H.265 帧内预测所有模式残差分布可视图

3.2.2 L 形迭代预测算法

显然，距离参考行、列越远的待预测点越有可能得到误差大的预测值，这是自然图像朴素的特性决定的。出于研究的严谨性，可通过统计大量预测单元的残差分布来验证这一结论^[46]。图 3.4 是 16×16 大小的预测单元，在所有 35 种帧内预测模式下产生的残差分布可视图，统计的对象来自 JCT-VC 推荐的 H.265 测试序列^[47] 中的 Traffic, Kimono, BasketballDrill, FourPeople 4 个序列的第一帧。图中颜色越深表示残差幅值越小即预测准确，反之颜色越淡表示预测越不准确。

分析图 3.4 的残差分布可得 2 条结论：1) 预测值的不准确性随着被预测点与参考点的距离增大而提升。换句话说，靠近参考点的待预测点能被准确地预测，

较远区域的待预测点得到的预测效果差。这一现象是空间相关性随着预测距离增大而衰减造成的；2) 不同的预测模式下，靠近参考点的边界区域的预测准确度变化很大。观察图可以发现，模式 2~9 中左侧边界的预测准确度远高于上侧边界，这是由于模式 2~9 的预测过程中用到更多的左侧参考点，这些参考点十分靠近左边界。相反地，模式 27~34 的预测过程中用到更多的上侧参考点，因此表现为上侧边界的预测准确度高。

基于上述分析，为了对抗帧内预测中远离参考点的区域预测效果差的缺陷，本课题提出了 L 形迭代预测算法 L-IP。L-IP 的核心思想是将二维待预测块分割为一系列一维 L 形（┐形）区域，各 L 形区域自左上角开始迭代地进行帧内预测，每次预测的参考点均为临近的上一次迭代的 L 形区域。对于一个 $N \times N$ 大小的预测单元，算法将其分割为 $N-1$ 个 L 形区域和 1 个像素点（处于右下角的像素点），如图 3.5(a) 所示。第一个 L 形区域 L1 与 H.265 标准的帧内预测一致，使用预测块外的左侧和上侧参考点（L0）进行预测，其余包含 m 个像素点的 L 形区域均使用包含 $m+2$ 个像素点的上一次迭代后的重建区域作为参考点进行预测。除去参考点位置不同，剩余的预测过程几乎与 H.26X 一致，同样使用投影与插值实现预测：

$$P_{(x,y)}^{L_j} = ((32 - \omega)R_{(0,i)}^{L_{j-1}} + \omega R_{(0,i+1)}^{L_{j-1}} + 16) \gg 5 \quad (3.2)$$

式中使用上标 L_j 表示样点所处的 L 形区域，其余物理量与式 3.1 意义一致。

L-IP 算法使待预测点与参考点的距离缩短到了 1 单位，使预测精度大幅上升。同时带来的一个优化是预测过程中不可用的需要填充的参考点（图 3.5(a) 中红色点）更加精确，距离可用点仅 1 单位，而在 H.26X 的块结构预测框架中，一个 $N \times N$ 大小的预测单元需要填充的不可用参考点可能距离可用点远达 N 单位。另外，观察图 3.2 与图 3.5(b) 可发现在相邻的 2 个角度预测模式间切换时，待预测点（黄色圆点）对应的参考点（灰色圆点）的权重分配会根据 Δd 产生变化，当待预测点远离参考点时，相邻 2 个角度模式的切换将产生较大的 $\Delta d'$ 致使权重分配变得十分粗犷，不利于在纹理丰富的区域生成变化精细的预测值。在 L-IP 算法方案中，这个缺陷由于待预测点与参考点的距离缩短到 1 单位而不再存在。

L-IP 算法中有 2 个细节需要说明：1) 尽管我们可以严格地按照式 3.2 从 L1

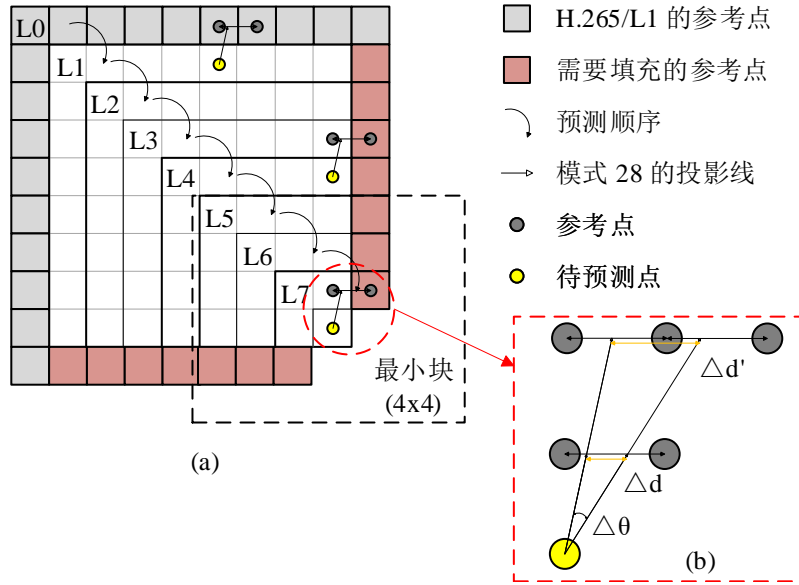


图 3.5 L-IP 算法

迭代预测到右下角的最后一个像素点，但我们在测试过程中发现，L-IP 算法需要对预测单元内的每一个 L 区域记录一个预测模式信息，在 L 形区域足够小的情况下，提高极少量待预测点的预测准确度带来的收益甚至无法抵消记录额外模式信息的开销。经过大量的实验统计，我们将 4×4 大小的待预测块规定为最小块，以图 3.5 为例，L1~L4 应用 L-IP 算法，右下角的 4×4 块则应用 H.26X 的基于块结构的预测；2) 由于需要对每个 L 形区域记录一个预测模式信息，使用 6bits 的码字编码 35 种预测模式之一显得代价过大，又因为待预测点与参考点的距离缩短，使并不需要如此密集的角度模式。基于上述 2 点原因，L-IP 中每个 L 形区域仅在 8 个预测模式中进行选择，一方面减少了模式信息的数据量，另一方面一定程度上加快了编码速度。另外，应用了 L-IP 算法并非完全舍弃 H.26X 基于块结构的预测。对于每个 PU，均分别应用 L-IP 算法预测和基于块结构的预测，选择率-失真代价较小的一方，换句话说，L-IP 与 H.26X 原始算法共同加入了编码的 RDO 过程。

3.2.3 算法性能测试与分析

L-IP 提出的初衷是提高远距离待预测点的预测准确度，为了验证预测准确度的改进，对 H.265 基于块结构的预测残差和 L-IP 算法得到的预测残差进行统计。统计的对象来自 JCT-VC 推荐的 H.265 测试序列中的 BasketballDrive(ClassB),

Kimono(ClassB), PartyScene(ClassC), Johnny(ClassE), SlideEditing(ClassF)。统计结果如表 3.1 所示。统计结果显示, 在 4×4 大小的预测单元中预测残差的平均幅值降低了 20.43%, 在 8×8 PU 中这一数值是 22.14%, 在 16×16 PU 中是 31.36%, 由于 32×32 大小的 PU 数量过少, 不具有统计意义, 不在表中列出。

表 3.1 H.265 与 L-IP 各尺寸 PU 平均残差幅值统计

序列类别	序列	4×4 PU		8×8 PU		16×16 PU	
		H.265	L-IP	H.265	L-IP	H.265	L-IP
ClassB	BasketballDrive	30.01	26.54	130.94	116.08	536.56	401.07
	Kimono	27.39	20.58	100.63	86.52	371.53	291.05
ClassC	PartyScene	109.97	98.62	463.98	316.04	1067.07	860.59
ClassE	Johnny	27.85	17.23	82.54	60.19	230.74	164.25
ClassF	SlideEditing	163.6	135.32	337.67	248.71	1464.68	560.35
残差平均幅值降低比率		20.43%		22.14%		31.36%	

H.26X 传统的基于块结构的预测中, 每个预测单元仅使用 1 种预测模式, 这使得图像中纹理丰富的区域必定被划分为多而小的预测单元。L-IP 算法下, 大尺寸预测单元的每个 L 形区域均有一个独立的预测模式, 因此应用 L-IP 后将使得 PU 划分结果中出现更多的大尺寸 PU。这一结论亦被实验数据验证, 如表 3.2 所示。表中数据的含义是: 一帧中有 x% 的像素点是在 $N\times N$ 大小的预测单元中进行预测得到的。根据统计结果, 16×16 与 32×32 的大尺寸预测单元分别增加了 18.67% 和 33.04%, 4×4 与 8×8 的小尺寸预测单元则分别减少了 47.42% 和 4.29%。

最后, 应用 L-IP 算法进行实际编码测试, 与 H.265 标准参考软件 HM-16.21^[24] 编码结果对比, 统计出编码效率提升的幅度。测试的对象为 JCT-VC 推荐的标准测试序列, 包括 5 类自然图像: ClassA(2560×1600), ClassB(1080p), ClassC(WVGA), ClassD(WQVGA), ClassE(720p) 和 1 类屏幕图像: ClassF。测试严格按照 H.265 通用测试环境 (Common Test Condition, CTC)^[47] 进行配置, 由于所提 L-IP 算法是为帧内无损编码设计的, 因此仅在全帧内 (All Intra, AI) 的无损配置下进行测试。统计结果如表 3.3 所示。

表 3.2 H.265 与 L-IP 各尺寸 PU 占比统计

序列	4×4 PU/%		8×8 PU/%		16×16 PU/%		32×32 PU/%	
	L-IP	H.265	L-IP	H.265	L-IP	H.265	L-IP	H.265
PeopleOnStreet	19.27	90.24	5.58	8.24	28.21	1.45	46.94	0.07
Traffic	14.93	87.80	4.22	11.46	29.50	0.70	51.35	0.05
BasketballDrive	48.44	63.73	22.78	26.49	16.30	9.36	12.48	0.43
BQTerrace	42.09	72.62	11.98	15.54	19.79	7.61	26.14	4.22
PU 数量变化率	-47.42		-4.29		18.67		33.04	

使用码率降低的百分比来衡量算法性能，表中负值表示使用 L-IP 算法编码得到的结果码率对比 H.265 标准降低。使用编码时间衡量算法复杂度，表中的编码时间、解码时间百分比数值由下式计算得到：

$$\Delta T = \frac{T_{proposed}}{T_{anchor}} \times 100\% \quad (3.3)$$

$T_{proposed}$ 表示所提算法的编码或解码所用时间， T_{anchor} 表示 HM 参考软件编码或解码用时。ClassA~E 中，序列 FourPeople 得到了最大的 12.51% 的编码效率提升，均值为 8.10%；ClassF 中，序列 SlideShow 获得了最大的 22.26% 的编码效率提升，均值为 14.76%。算法复杂度方面，L-IP 使编码时间增加了 25%，解码时间基本不变，证明 L-IP 算法性能良好，具有较高的应用价值。

3.3 帧内无损编码的编码块划分优化

H.26X 的帧内分块方案中，基于四叉树结构的分块算法决定了一个预测单元仅可在保留 $2N \times 2N$ 结构与划分为 4 个 $N \times N$ 结构这 2 个选项中进行选择。尽管这能有效地做到在不同纹理丰富程度的区域自适应选择分块大小，但过于粗暴的 2 选 1 导致每划分一次 $N \times N$ 结构都产生 4 组需要独立编码的模式信息、分块信息及其他辅助信息，一定程度上降低了编码效率^[48]。本节将针对该缺陷进行分析与优化。

表 3.3 L-IP 算法性能测试

序列类别	序列	分辨率	码率	编码时间
ClassA	PeopleOnStreet	2560×1600	-11.29%	125.98%
	Traffic	2560×1600	-11.46%	124.76%
	BasketballDrive	1920×1080	-4.20%	126.57%
	BQTerrace	1920×1080	-8.57%	125.85%
ClassB	Cactus	1920×1080	-3.74%	126.96%
	Kimono	1920×1080	-7.71%	127.41%
	ParkScene	832×480	-6.58%	127.37%
	BasketballDrill	832×480	-5.51%	126.32%
ClassC	BQMall	832×480	-6.17%	126.48%
	PartyScene	832×480	-5.58%	125.11%
	RaceHorsesC	832×480	-8.10%	124.38%
	BasketballPass	416×240	-12.36%	124.54%
ClassD	BlowingBubbles	416×240	-5.47%	125.01%
	BQSquare	416×240	-5.27%	124.80%
	RaceHorses	416×240	-8.51%	125.34%
	FourPeople	1280×720	-12.51%	124.64%
ClassE	Johnny	1280×720	-10.97%	124.51%
	KristenAndSara	1280×720	-11.86%	123.96%
	BasketballDrillText	832×480	-6.07%	124.67%
ClassF	ChinaSpeed	1024×768	-17.15%	121.53%
	SlideEditing	1280×720	-13.57%	123.07%
	SlideShow	1280×720	-22.26%	126.19%
ClassA~E 均值		-	-8.10%	125.56%
ClassF 均值		-	-14.76%	123.87%
ClassA~F 均值		-	-9.31%	125.25%
解码时间			103.21%	

3.3.1 编码块划分决策过程分析

H.265 标准将 H.264 中 16×16 的宏块扩展到了 64×64 ，并且使用编码树单元 (Coding Tree Unit, CTU) 和 CU 取代了 H.264 中宏块的概念，同时应用基于四叉树的分块方案使分块的大小根据图像中纹理丰富与否进行自适应调整。例如，在图像中平滑或仅含简单纹理的区域，以大块的形式进行预测和编码，节省了大量预测模式与分块模式信息的开销，从而提高编码效率，但也一定程度上降低了预测的准确性。而在纹理丰富的区域将大块划分为大量小尺寸的预测块，使预测残差减小，但同时会产生大量需要额外编码的模式信息和分块标志信息。因此，我们需要在预测准确性和编码产生的数据量之间寻找折衷，这也是 H.265 中 RDO 的重要一环。HM 参考软件中的 RD 代价 J_{mode} 计算方式如式 3.4。

$$J_{mode} = B_{mode} + \lambda_{mode} \cdot SSE \quad (3.4)$$

其中 B_{mode} 表示编码当前单元产生的数据量，重建块的误差平方和 SSE 用于衡量失真，有时也用绝对误差和或 PSNR 进行衡量， λ_{mode} 为拉格朗日因子。在无损编码中重建块不再存在失真，RD 代价的计算简化为：

$$J_{mode} = B_{mode} \quad (3.5)$$

在编码过程中，RDO 用于搜寻最佳的预测模式与最佳的分块模式。

编码时，一帧图像首先被切分为若干大小相等的 CTU，RDO 将从最小的 4×4 分块开始向上搜索，以 RD 代价为依据确定最终一个 CTU 的分块模式。图 3.6(a) 展示了一个 CTU 的分块结果，每个方框代表一个独立的预测单元，对于每一个单元都将编码 1 个分块标志指示当前块是否往下切分一次。从图中可观察到在基于四叉树结构的分块算法下，由于右下角区域存在极小部分纹理丰富的区域，CTU 经过 RDO 被切分为 19 个独立进行预测的小块，因此编码该 CTU 需要处理的数据有： 64×64 个预测残差；19 个预测模式信息；13 个分块标志。尽管左、上侧 3 个较大块可能使用的是相同的预测模式，也必须编码 3 次预测模式信息，一定程度上降低了编码效率。在充分理解 H.26X 帧内分块过程存在的缺陷后，下一节将提出本课题的改进算法。

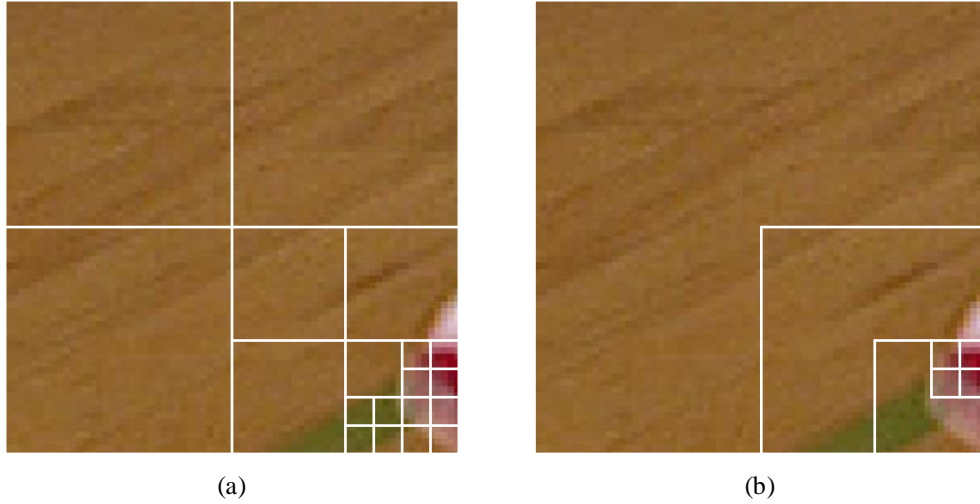


图 3.6 基于四叉树的分块算法与 L-BP 算法

3.3.2 L 形分块算法

根据上节所分析，在 H.265 传统的基于四叉树结构的分块算法下，一旦当前 $2N \times 2N$ 大小的编码单元存在小区域的丰富纹理，该单元大概率将根据 RD 代价判断切分为 4 个 $N \times N$ 大小的子单元分别进行预测和编码，从而增加了 3 个需要额外编码的预测模式信息。并且，如果应用了 3.2.2 节所述的 L-IP 算法，一个 $N \times N$ 大小的预测单元需要记录 $N-3$ 个预测模式信息，需要额外编码的预测模式将进一步增加。考虑到这一情况，结合 L-IP 的预测结构，本课题提出了一种 L 形分块算法 L-BP。L-BP 可以保证编码单元仅包含小区域丰富纹理时，使编码单元仍保持大块的结构进行预测和编码，以减少预测模式信息的增加量，最大化 L-IP 算法带来的增益。

L-BP 算法的核心思想是将一个需要向下切分的编码单元视为 2 部分组成：一个 L 形区域和一个块状区域，而非四叉树结构下大小均等的 4 部分。其中 L 形区域是由四叉树结构下的 3 个块状区域组合而成，根据 3 个块状不同位置的组合，将产生 4 种可能性：「L」，图 3.6(b) 展示了 L-BP 的分块结果，其中出现了两种 L 形分块模式（「L」）。对于剩下的块状区域，将继续判断是否进行 L-BP 分块，直到分割到最小单元。整体的分块决策流程与 H.265 一致，从底层向上搜索。值得一提的是提出 L-BP 算法的同时并未抛弃 H.265 基于四叉树的分块算法，换句话说，L-BP 提供了 4 种新的可能的分块模式，分块决策时从 H.265 的 2 选 1 增加到了 6 选 1，使分块更加灵活，以适应不同纹理丰富程度的区域。

图 3.7(a)(b) 对比了一个 CTU 在 H.265 基于四叉树结构的分块算法和加入了 L-BP 算法后的分块结果。图 3.7(c) 展示了加入了 L-BP 算法后的整体分块决策流程，在每一个层级均提供了 6 个分块模式进行选择。

3.3.3 算法性能测试与分析

最后，应用 L-BP 算法进行实际编码测试，与 H.265 标准参考软件 HM-16.21^[24] 编码结果对比，统计出编码效率提升的幅度。测试的对象和环境与 3.2.3 节所述相同，不再赘述。统计结果如表 3.4 所示。

使用码率降低的百分比来衡量算法性能，表中负值表示使用 L-BP 算法编码得到的结果码率对比 H.265 标准降低。ClassA~E 中，序列 KristenAndSara 得到了最大的 4.40% 的编码效率提升，均值为 2.04%；ClassF 中，序列 SlideShow 获得了最大的 5.43% 的编码效率提升，均值为 3.70%。使用编码时间衡量算法复杂度，L-BP 算法几乎没有影响编码、解码时间。统计数据证明 L-BP 算法性能好，具有较高的应用价值。

3.4 帧内无损编码的待编码系数再处理

H.26X 系列标准可以通过简单地跳过变换、量化、去块滤波、自适应样点补偿等可能引入失真的步骤，实现帧内无损压缩。但也因此使得编码器去相关性差，待编码系数仍具有较高的能量，为后续的熵编码带来了极大的压力。本节将针对该缺陷进行分析与优化。

3.4.1 帧内预测残差分析

无损编码方案中跳过了量化步骤，此时待编码数据不再是变换域系数而是预测残差，因此其仍具有空间域的物理意义。为了直观地验证这一结论，尝试将残差绘制出来观察。图 3.8 展示了在 H.265 标准下经过帧内预测后的残差图像（数值整体平移了 128 以绘制负值）^[49, 50]。

出于多种原因，尽管帧内预测的准确性不断提高（H.264 仅使用 9 种预测模式，H.265 增加到 35 种，H.266 达到了 67 种），残差中仍然可能保留丰富的边缘

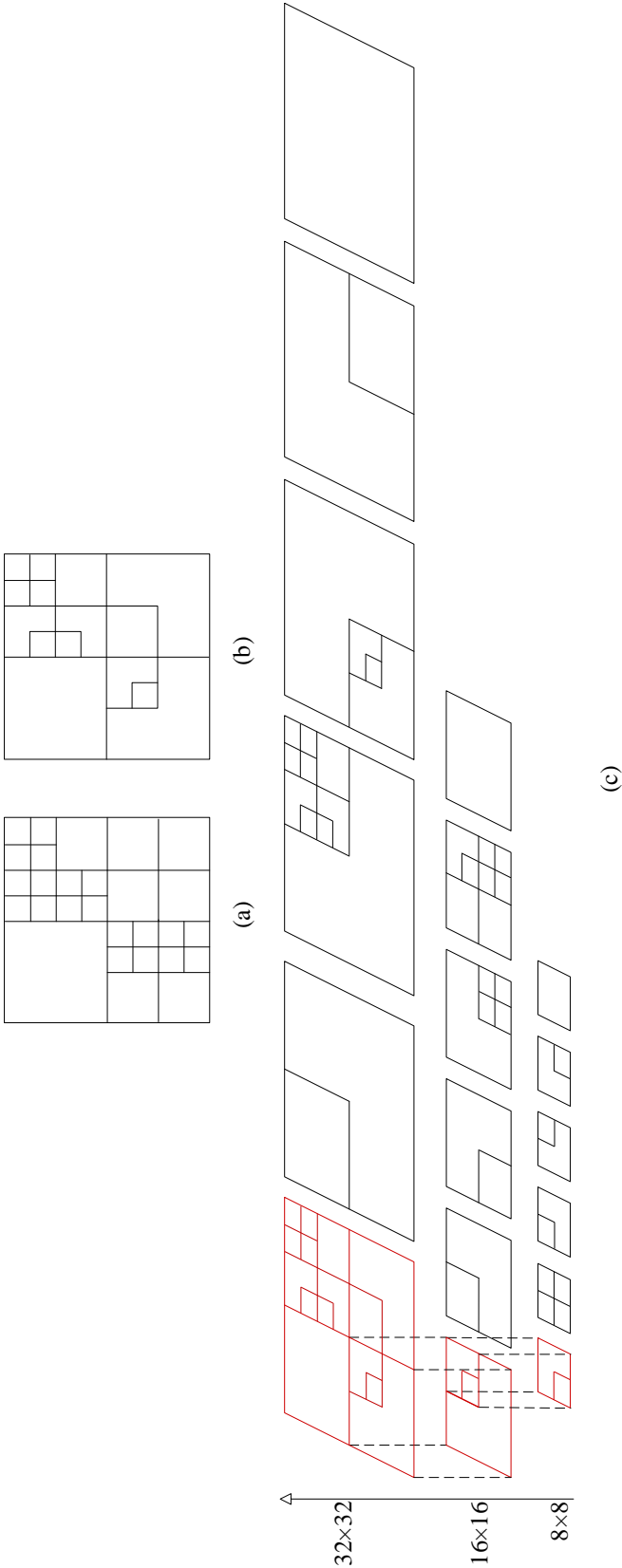


图 3.7 应用 L-BP 算法后的分块决策流程

表 3.4 L-BP 算法性能测试

序列类别	序列	分辨率	码率	编码时间
ClassA	PeopleOnStreet	2560×1600	-1.78‰	100.98‰
	Traffic	2560×1600	-1.89‰	100.03‰
	BasketballDrive	1920×1080	-2.28‰	101.06‰
	BQTerrace	1920×1080	-1.80‰	101.31‰
ClassB	Cactus	1920×1080	-1.15‰	101.28‰
	Kimono	1920×1080	-1.57‰	101.27‰
	ParkScene	832×480	-1.48‰	101.29‰
	BasketballDrill	832×480	-1.96‰	100.82‰
ClassC	BQMall	832×480	-1.46‰	100.82‰
	PartyScene	832×480	-1.42‰	100.49‰
	RaceHorsesC	832×480	-1.46‰	100.50‰
	BasketballPass	416×240	-2.62‰	100.00‰
ClassD	BlowingBubbles	416×240	-1.05‰	100.00‰
	BQSquare	416×240	-1.54‰	100.63‰
	RaceHorses	416×240	-1.17‰	101.32‰
	FourPeople	1280×720	-3.52‰	100.73‰
ClassE	Johnny	1280×720	-4.11‰	100.95‰
	KristenAndSara	1280×720	-4.40‰	100.80‰
	BasketballDrillText	832×480	-1.96‰	101.00‰
ClassF	ChinaSpeed	1024×768	-4.75‰	100.45‰
	SlideEditing	1280×720	-2.64‰	100.66‰
	SlideShow	1280×720	-5.43‰	102.68‰
ClassA~E 均值		-	-2.04‰	100.79‰
ClassF 均值		-	-3.70‰	101.20‰
ClassA~F 均值		-	-2.34‰	100.87‰
解码时间			100.71‰	

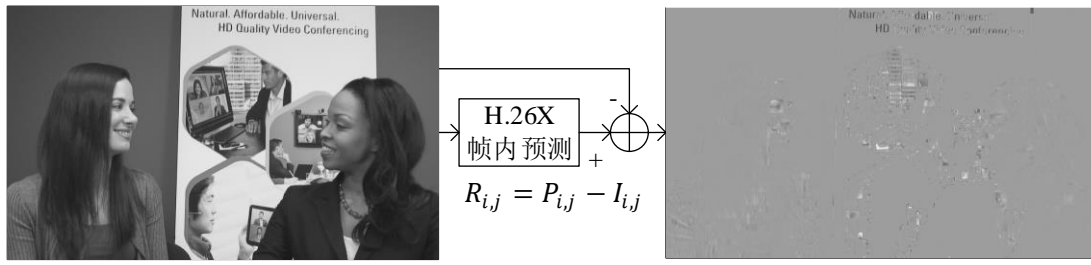


图 3.8 KristenAndSara 第一帧及其帧内预测残差（亮度）

特征。首先，由于 PU 最小尺寸的限制，角度预测在图像中纹理丰富的区域始终无法得到良好的预测结果；其次，当被预测块靠近参考像素的边沿存在不连续性时，角度预测很可能会插入原始块中本不存在的方向性条纹。

帧内预测的目的是消除图像的空间相关性，显然，通过帧内预测得到的残差仍具有较强的特殊的空间相关性。与自然图像不同，帧内预测残差的空间相关性体现为含有丰富的边缘特征。在充分理解 H.26X 帧内预测残差存在的特征后，下一节将对本课题的改进算法进行说明。

3.4.2 残差中值边缘检测算法

为了利用帧内预测残差特殊的空间相关性，进一步提高视频帧内编码的效率，本课题提出了一种在 H.264、H.265、H.266 中通用的基于残差中值边缘检测的待编码系数再处理算法 R-MED。R-MED 利用了帧内预测残差具有丰富的边缘特征这一统计结果，对预测残差进行二次预测。首先将当前编码块的残差首行、首列作为参考点；然后利用中值边缘检测对剩余点逐点进行预测并求得新的残差；最后统计处理前后的残差能量，选择能量较小的一组进行熵编码。如图 3.9(c) 所示。

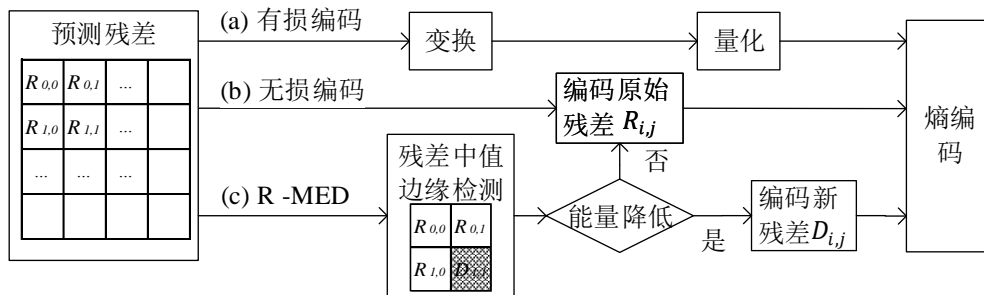


图 3.9 H.26X 无损编码方案及 R-MED 应用方案

中值边缘检测 MED 已被应用在图像的低复杂度无损压缩 (Low Complexity Lossless Compression for Images, LOCO-I)^[51] 算法中。但与 LOCO-I 使用 MED 对原始像素进行预测不同, 考虑到 MED 具有很强的边缘检测能力, 恰好残差图像中包含丰富的边缘特征, 因此将其应用到残差图像的二次预测上。R-MED 的新预测值 $P'_{i,j}$ 由下式给出:

$$P'_{i,j} = \begin{cases} \min(R_{i,j-1}, R_{i-1,j}), & R_{i-1,j-1} \geq \max(R_{i,j-1}, R_{i-1,j}) \\ \max(R_{i,j-1}, R_{i-1,j}), & R_{i-1,j-1} \leq \min(R_{i,j-1}, R_{i-1,j}) \\ R_{i,j-1} + R_{i-1,j} - R_{i-1,j-1}, & o.w \end{cases} \quad (3.6)$$

式中 $R_{i,j}$ 表示 H.26X 经过帧内预测后得到的残差。如图 3.10 所示, R-MED 可以精准地得到强边缘处的预测值。当 $R_{i-1,j-1} \geq \max(R_{i,j-1}, R_{i-1,j})$ 时 (见图 3.10(a)), 如果 $R_{i,j-1} > R_{i-1,j}$, 那么在 $P'_{i,j}$ 左侧的垂直方向上很可能存在明显的垂直边缘, 此时最佳预测值为 $R_{i-1,j}$; 反之则很可能在上方存在明显的水平边缘, 此时最佳预测值为 $R_{i,j-1}$ 。类似地, 图 3.10(b) 展示了 $R_{i-1,j-1} \leq \min(R_{i,j-1}, R_{i-1,j})$ 时, $P'_{i,j}$ 的选择策略。

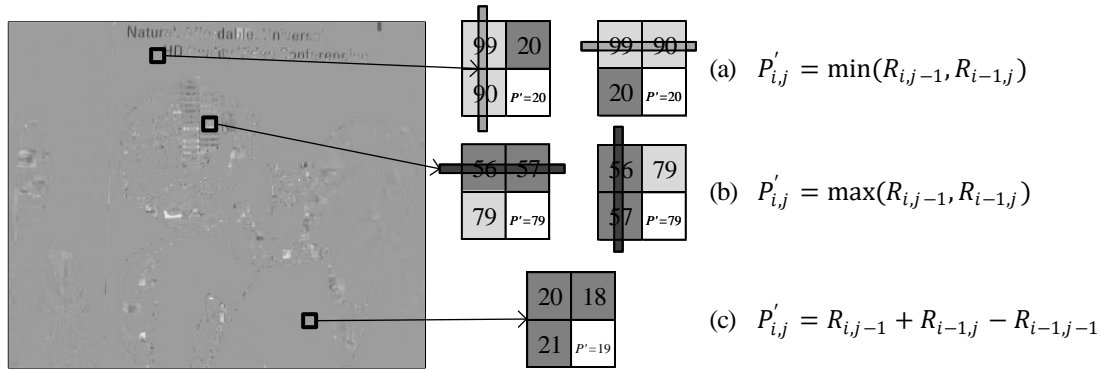


图 3.10 R-MED 的预测策略

此外, 如果 R-MED 没有检测到明显的水平或垂直边缘, 将采取模仿水平/垂直方向相邻点变化趋势的预测策略。如图 3.10(c) 所示, $P'_{i,j}$ 周围的参考点并不满足式 3.6 的前两个条件, 表明该区域变化平坦没有复杂纹理。 $P'_{i,j}$ 上方参考点的数值由 20 降低到 18(-2), 左侧参考点的数值由 20 增加到 21(+1), 因此 $P'_{i,j}$ 模仿上方的变化趋势得到 $P'_{i,j} = R_{i,j-1} - 2 = 21 - 2 = 19$, 或模仿左侧的变化趋势得到 $P'_{i,j} = R_{i-1,j} + 1 = 18 + 1 = 19$ 。

通过上述分析，证明了 R-MED 具有对边缘信息预测准确的特点，因此适合将其应用到 H.26X 残差图像的二次预测中。R-MED 对经过 H.26X 帧内预测后，进行熵编码前的预测残差进行再处理。以 PU 为基本单位，将各 PU 的首行、首列作为参考点，对剩下的数据点逐点应用 R-MED 算法计算新的残差 $D_{i,j}$ ：

$$D_{i,j} = \begin{cases} R_{i,j}, & i = 0 \text{ or } j = 0 \\ P'_{i,j} - R_{i,j}, & o.w \end{cases} \quad (3.7)$$

式中 $R_{i,j}$ 表示 H.26X 经过帧内预测后得到的残差， $P'_{i,j}$ 表示通过 R-MED 算法得到的预测值。然后分别计算处理前后的残差总能量 E_{src} 、 E_{R-MED} ：

$$E_{src} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} R_{i,j}^2 \quad (3.8)$$

$$E_{R-MED} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} D_{i,j}^2 \quad (3.9)$$

最后选择总能量较小的一组进行熵编码。通过残差的总能量判断是否选用经过二次处理的残差是因为，H.26X 在对预测残差进行编码前，需要使用哥伦布-莱斯码或指数哥伦布码对其进行二值化，二值化后码字的长度将与残差的绝对幅值呈指数关系^[42]。图 3.11 是一个 4×4 的预测单元经过 H.265+R-MED 算法处理前后的效果，其残差总能量明显降低。

0	0	-2	-1
-1	-1	-2	-1
0	1	0	0
0	-1	-2	-1

$E_{src} = 19$

0	0	-2	-1
-1	0	0	0
0	-1	0	0
0	2	1	-1

$E_{R-MED} = 13$

图 3.11 一个经过 R-MED 处理的预测单元

3.4.3 算法性能测试与分析

R-MED 算法提出的初衷是充分利用待编码系数的特殊空域相关性，通过再预测降低其整体能量，为了验证降低帧内预测残差能量的效果，统计了在 H.265

表 3.5 应用 H.265+R-MED 算法后残差能量的变化率

类别	序列	分辨率	$\sum_{CU} E_{src}$	$\sum_{CU} E_{R-MED}$	变化率
A	PeopleOnStreet	2560×1600	633451646	94108069	-85.14%
B	Kimono	1920×1080	73752438	11522730	-84.38%
C	BQMall	832×480	59358254	25243957	-57.47%
D	BlowingBubbles	416×240	14714641	8036874	-45.38%
E	KristenAndSara	1280×720	97320940	12361610	-87.30%
F	SlideEditing	1280×720	1133005976	595102758	-47.48%
均值	-	-	-	-	-67.86%

标准下，部分测试序列第一帧经过 R-MED 算法处理前后整体能量的变化率。统计结果如表 3.5 所示，单帧图像的残差能量平均降低了 67.86%，且表现出高分辨率下性能较好的趋势。通过绘制残差图像也能验证 R-MED 降低预测残差空域冗余和整体能量的效果。图 3.12 是测试序列 KristenAndSara 第一帧经过 R-MED 处理前后的残差图像，可直观地观察到其边缘特征极大地减少，证明 R-MED 算法对预测残差图像有优秀的处理能力。

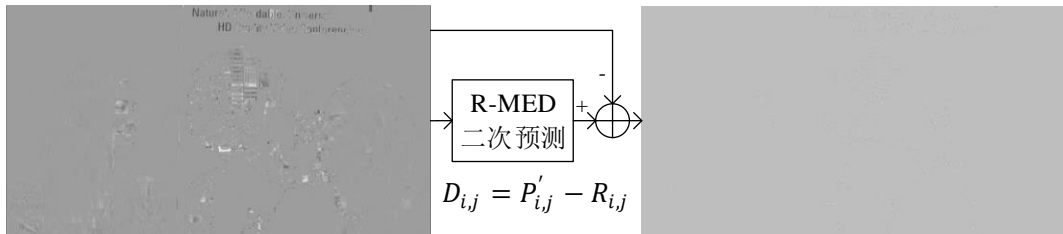


图 3.12 KristenAndSara 第一帧残差经 R-MED 处理后效果（亮度）

最后，应用 R-MED 算法进行实际编码测试。与 3.2.3 节、3.3.3 节的测试不同，由于 R-MED 算法通用性极强，可在 H.26X 系列编码标准中通用，因此 R-MED 分别在 H.265 标准参考软件 HM-16.21 和 H.266 标准参考软件 VTM-12.3 中实现，并严格按照 H.265 通用测试环境、H.266 无损编码通用测试环境^[52]，使用 JCT-VC 推荐的标准测试序列进行测试。针对 H.265 的结果统计如表 3.6 所示，针对 H.266 的结果统计如表 3.7 所示。

与 H.265 参考软件 HM-16.21 相比，经过 R-MED 算法处理的视频序列达到

表 3.6 R-MED 算法应用在 H.265 的性能测试

序列类别	序列	分辨率	H.265+R-MED vs	
			Bypass TransQuant	RDPCM ^[9]
A	PeopleOnStreet	2560×1600	-11.10%	-5.99%
	Traffic	2560×1600	-10.10%	-3.54%
	BasketballDrive	1920×1080	-2.19%	-0.49%
B	BQTerrace	1920×1080	-5.43%	-0.96%
	Cactus	1920×1080	-2.92%	-0.99%
	Kimono	1920×1080	-6.82%	-2.63%
	ParkScene	832×480	-5.85%	-2.26%
C	BasketballDrill	832×480	-2.61%	-1.14%
	BQMall	832×480	-4.16%	-0.66%
	PartyScene	832×480	-3.80%	-0.51%
	RaceHorsesC	832×480	-7.03%	-2.64%
D	BasketballPass	416×240	-9.10%	-1.92%
	BlowingBubbles	416×240	-4.11%	-1.93%
	BQSquare	416×240	-3.10%	-0.48%
	RaceHorses	416×240	-8.29%	-3.53%
E	FourPeople	1280×720	-9.23%	-3.83%
	Johnny	1280×720	-6.99%	-2.58%
	KristenAndSara	1280×720	-7.35%	-2.60%
	Vidyo1	1280×720	-9.93%	-3.30%
F	Vidyo3	1280×720	-9.10%	-3.60%
	Vidyo4	1280×720	-8.92%	-2.65%
	BasketballDrillText	832×480	-3.01%	-1.19%
	ChinaSpeed	1024×768	-9.13%	-1.35%
	SlideEditing	1280×720	-7.33%	-0.51%
	SlideShow	1280×720	-18.45%	-5.16%
平均码率优化			-7.04%	-2.26%
编码时间			104%	107%
解码时间			95%	98%

表 3.7 R-MED 算法应用在 H.266 的性能测试

序列类别	序列	分辨率	H.266+R-MED vs	
			Bypass TransQuant	BDPCM ^[1]
A	PeopleOnStreet	2560×1600	-9.67%	-5.99%
	Traffic	2560×1600	-8.60%	-3.77%
	BasketballDrive	1920×1080	-1.42%	-0.52%
B	BQTerrace	1920×1080	-4.16%	-1.15%
	Cactus	1920×1080	-2.14%	-1.05%
	Kimono	1920×1080	-5.96%	-3.35%
	ParkScene	832×480	-4.87%	-2.58%
C	BasketballDrill	832×480	-1.92%	-1.22%
	BQMall	832×480	-3.10%	-0.86%
	PartyScene	832×480	-2.65%	-0.69%
	RaceHorsesC	832×480	-6.13%	-3.25%
D	BasketballPass	416×240	-7.71%	-2.69%
	BlowingBubbles	416×240	-3.15%	-0.75%
	BQSquare	416×240	-2.24%	-0.56%
	RaceHorses	416×240	-7.23%	-3.58%
E	FourPeople	1280×720	-7.92%	-3.97%
	Johnny	1280×720	-6.13%	-3.04%
	KristenAndSara	1280×720	-6.40%	-2.82%
	Vidyo1	1280×720	-8.56%	-3.65%
F	Vidyo3	1280×720	-7.64%	-3.80%
	Vidyo4	1280×720	-7.36%	-3.04%
	BasketballDrillText	832×480	-2.21%	-1.17%
	ChinaSpeed	1024×768	-7.86%	-1.47%
	SlideEditing	1280×720	-7.20%	-0.80%
	SlideShow	1280×720	-17.36%	-6.73%
平均码率优化			-5.98%	-2.50%
编码时间			117%	111%
解码时间			83%	93%

了最大 18.45%，平均 7.04% 的码率优化，平均编码时间仅增加 4%，同时由于经过处理的待编码残差能量大幅减少，缓解了后续熵编解码器的压力，因此平均解码速度加快 5%，另外还证明比 HEVC-SCC 扩展标准中的 RDPCM 性能更好。与 H.266 参考软件 VTM-12.3 相比，视频码率达到了最大 17.36%，平均 5.98% 的优化，编码时间平均增加 17%，解码速度提升 17%，同时证明比 H.266 标准中的 BDPCM 性能更好。R-MED 在 H.265 和 H.266 中的应用证明了该算法具有很强的通用性，也可与各种帧内预测优化算法结合使用，甚至有可能在编码结构相似的数字音视频编码技术标准 (Audio Video Standard, AVS) 或 AOMedia Video 1(AV1) 标准中应用。

此外，为了证明 R-MED 算法性能良好，在表 3.8 中给出了近年多种针对 H.265 无损帧内编码的优化算法与 R-MED 算法的性能比较。表中数据指对比 H.265 跳过变换量化方案的优化程度。由于大部分文献并未给出完整的测试序列的实验数据，因此仅能对比下表中所示的 5 个测试序列的性能：

表 3.8 R-MED 与其他算法的性能比较（对比 H.265）

类别	序列	SAP-HV/%	SAP /%	SAP1 /%	SAP+SWP2 +DTM/%	RDPCM +pwm/%	R-MED /%
B	Kimono	-2.32	-4.42	-4.47	-5.84	-5.07	-6.82
B	ParkScene	-3.38	-4.95	-4.97	-6.12	-5.31	-5.85
F	BasketballDrillText	-1.38	-5.77	-5.93	-6.19	-3.33	-3.01
F	SlideEditing	-8.03	-9.5	-9.76	-11.15	-8.2	-7.33
F	SlideShow	-9.59	-12.69	-12.74	-14.07	-13.04	-18.45
平均码率优化		-4.94	-7.47	-7.57	-8.67	-6.99	-8.29
编码时间		97	97	97	273	104	104
解码时间		99	99	98	196	103	95

上述算法中，SAP+SWP2+DTM^[13] 方案获得了最佳的码率优化，但编解码时间大幅增加。相比之下 R-MED 以增加少量编码时间的代价获得了可观的码率优化，证明所提算法具有较高的实用价值。

值得一提的是，本课题还尝试将 R-MED 算法应用在有损压缩上，测试结

果显示其在编码高分辨率图像 (ClassA) 和屏幕图像 (ClassF) 的结果中得到了 2%~3% 的码率优化。尽管有损编码时的待编码系数不再具有传统的空间相关性, 但 R-MED 仍有一定的降低数据能量的作用, 这涉及到将来课题的研究内容, 在此不再具体展开。

3.5 联合算法及性能测试

根据 3.2 节与 3.3 节所述, L-IP 算法与 L-BP 算法可结合在一起对帧内无损编码进行优化, 达到 $1 + 1 > 2$ 的效果。因此, 本课题提出了将 L-IP 与 L-BP 有机融合的联合算法: L 形分块迭代预测算法 (L-shape-based Block Partitioning and Iterative Prediction, L-BPIP)。算法在编码过程仍进行与 H.265 类似的 RDO, 但优化过程中的可选项大幅增多。对任意一 CU, RDO 过程将在 11 个可选项中进行, 即在图 3.7 的基础上增加 $2N \times 2N$ 块、L 形区域的 L 形迭代预测方案。其中 RD 代价最小的一个将被确定为当前 CU 最终的预测、分块模式, 如图 3.13 所示。

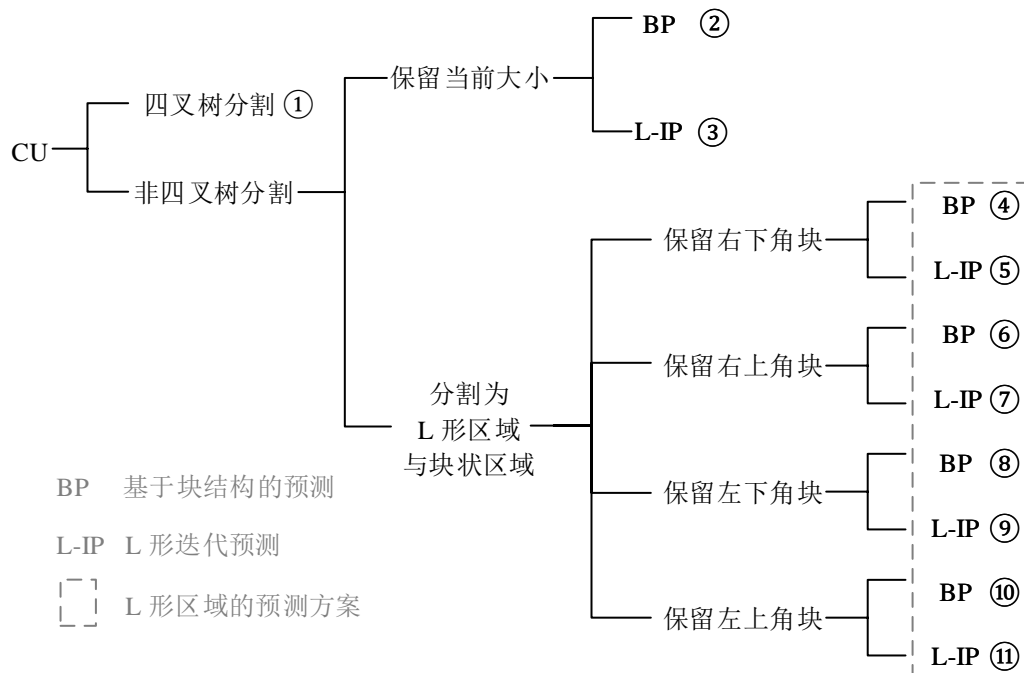


图 3.13 L-BPIP 算法 RDO 过程的比较对象

由于存在 11 种分块可能, 直接编码该信息需要 4 比特数据量。通过测试发现, 四叉树分割 (1)、保留右下角块 BP (3)、保留右下角块 L-IP (4) 三种模式

占比高达 79%，因此决定使用变长编码编码分块模式信息，使平均码字长度最短。联合算法 L-BPIP 需要处理的另一个重要问题是编码单元的处理顺序问题。在 H.26X 框架中，由于预测过程需要使用当前编码单元右上侧和左下侧的重建点作为参考点，编码单元总是按照 Z 形扫描顺序进行处理，以保证尽可能多的参考点是已知的重建点。L-BPIP 算法在预测过程中同时存在 L 形迭代预测和基于块结构的预测，为了保证参考点的可用性需要对编码顺序进行调整。以编码单元保留左上角块和保留右下角块为例进行说明，如图 3.14 所示，对 B2 进行预测

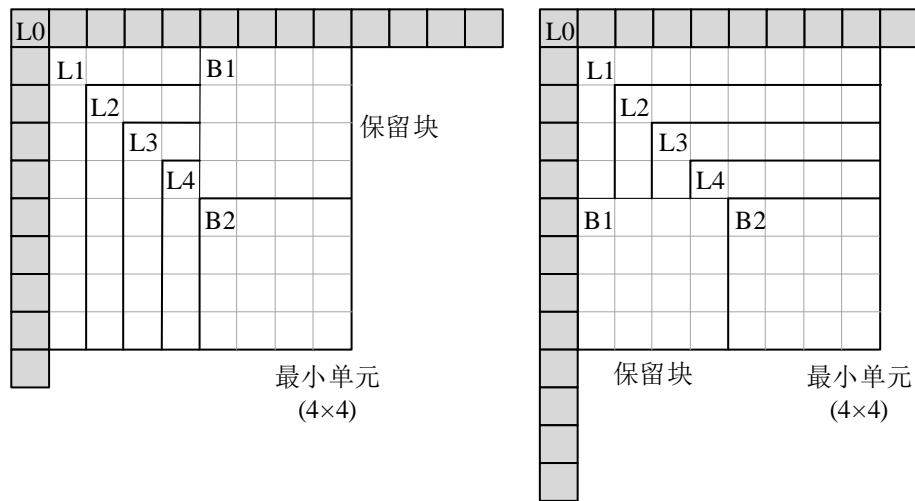


图 3.14 L-BPIP 处理顺序说明

需要 B1 边缘以及 L1~L4 边缘的重建点作为参考点，对 B1 进行预测需要 L1~L4 边缘的重建点作为参考点，对 L2~L4 进行预测需要其前一级 L 形区域的重建点作为参考点。因此该编码单元最终的编码顺序为：L1→L2→L3→L4→B1→B2。在解码端将采取相同的处理顺序，可确保在对任意一个 L 形区域或块状区域进行预测时，其需要的参考点已尽可能多地被重建完成。

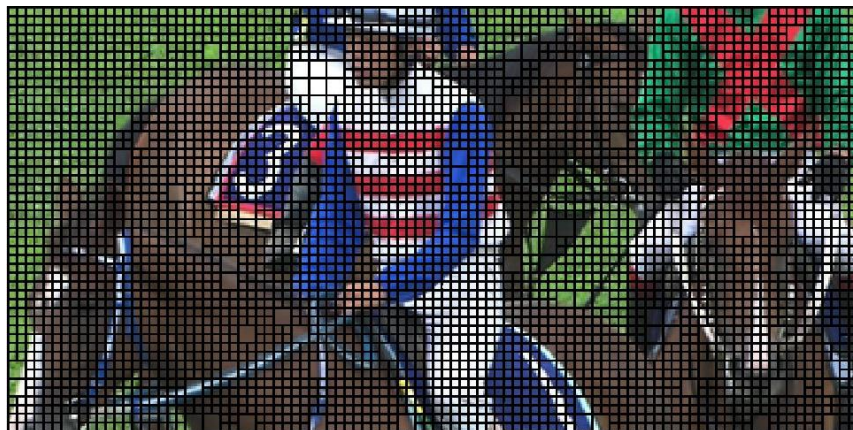
最后，应用 L-BPIP 算法进行实际编码测试，测试环境、测试序列、对比对象均与 3.2.3 节所述一致，统计结果如表 3.9 所示。ClassA~E 中，序列 FourPeople 得到了最大的 14.14% 的编码效率提升，均值为 8.87%；ClassF 中，序列 SlideShow 获得了最大的 26.42% 的编码效率提升，均值为 16.24%。使用编码时间衡量算法复杂度，L-BPIP 使编码时间增加了 29%，略高于 L-IP，几乎没有影响解码时间。统计数据证明 L-BPIP 算法性能良好，具有较高的应用价值。联合算法 L-BPIP 对比单独的 L-IP 提升了 0.89% 的编码效率，对比 L-BP 提升了 7.87%，证明了

表 3.9 L-BPIP 算法性能测试

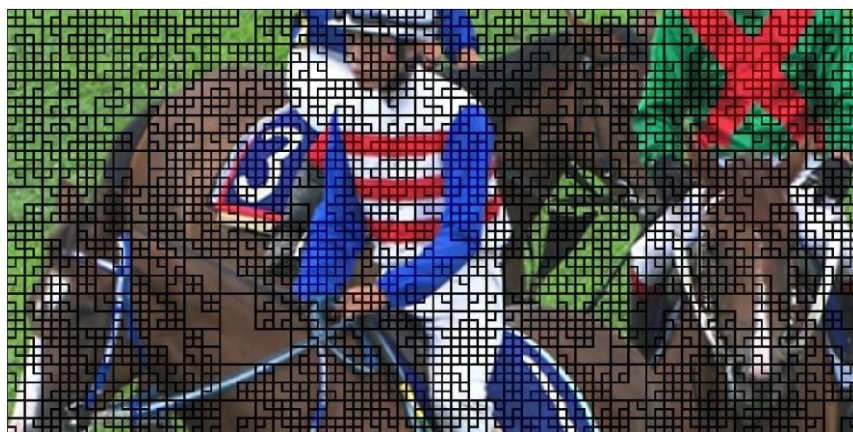
序列类别	序列	通道			
		Y	Cb	Cr	All
A	PeopleOnStreet	-14.03%	-13.86%	-11.43%	-13.42%
	Traffic	-14.30%	-9.60%	-13.98%	-13.26%
	BasketballDrive	-2.74%	-3.72%	-9.90%	-4.30%
	BQTerrace	-9.32%	-7.40%	-7.84%	-9.12%
B	Cactus	-3.76%	-2.84%	-5.34%	-3.92%
	Kimono	-6.74%	-13.72%	-21.14%	-9.07% (-7.57%) ^①
	ParkScene	-7.50%	-9.36%	-10.43%	-7.95% (-8.27%)
	BasketballDrill	-2.81%	-2.06%	-4.63%	-3.76%
C	BQMall	-5.80%	-7.06%	-10.91%	-6.55%
	PartyScene	-5.27%	-6.74%	-8.30%	-5.71%
	RaceHorsesC	-8.14%	-14.89%	-15.70%	-9.21%
	BasketballPass	-13.02%	-16.15%	-17.76%	-13.79%
D	BlowingBubbles	-4.89%	-8.35%	-11.11%	-5.42%
	BQSquare	-4.37%	-6.20%	-8.33%	-5.05%
	RaceHorses	-8.58%	-13.80%	-14.54%	-9.38%
	FourPeople	-11.43%	-23.62%	-22.93%	-14.14%
E	Johnny	-9.15%	-20.92%	-20.63%	-12.29%
	KristenAndSara	-10.44%	-21.16%	-20.91%	-13.24%
	BasketballDrillText	-3.39%	-3.30%	-5.83%	-4.50% (-6.38%)
F	ChinaSpeed	-19.37%	-15.40%	-16.22%	-18.56% (-14.42%)
	SlideEditing	-16.86%	-8.24%	-9.54%	-15.49% (-11.80%)
	SlideShow	-25.25%	-26.96%	-26.22%	-26.42% (-18.92%)
ClassA~E 均值		-7.90%	-11.19%	-13.10%	-8.87%
ClassF 均值		-16.22%	-13.48%	-14.45%	-16.24% (-12.88%)
ClassA~F 均值		-9.42%	-11.61%	-13.35%	-10.21%
编码时间				129%	
解码时间				103%	

① 圆括号中的数值为 SAP-E^[11, 16] 算法性能, 作为对比

将二者融合的有效性和必要性。图 3.15 展示了联合算法对单帧图像编码块划分的显著影响，数据统计显示测试序列中 16×16 与 32×32 的大尺寸编码块增加了



(a) H.265 编码块划分结果



(b) L-BPIP 编码块划分结果

图 3.15 H.265 标准与 L-BPIP 编码块划分结果对比

25.82%。很容易想象，编码块划分结果中的大尺寸块越多，所提算法发挥的性能越高，因为大尺寸块将节省大量预测模式、分块标志信息的开销，同时还能得到相同甚至更高精确度的预测结果。

此外，据我们所知，近年 H.265 无损帧内编码的优化算法中 SAP 系列得到的编码效率优化名列前茅，其中又属 SAP-E^[11, 16] 表现最为突出。因此将本课题所提算法 L-BPIP 与其进行对比，由于文献 [16] 中仅给出了 SAP-E 在测试序列 ParkScene(ClassB), Kimono(ClassB), BasketballDrillText(ClassF), ChinaSpeed(ClassF), SlideEditing(ClassF), SlideShow(ClassF) 的统计结果，故只对比上述 6 个序列，数据列于表 3.9 圆括号中。对比结果显示，在 ClassB 中，所提

L-BPIP 算法比 SAP-E 算法编码效率高 0.59%，在 ClassF 中高 3.36%，证明所提算法性能优秀，处于学术界前沿。

3.6 本章小结

本章从软件算法层面介绍了课题研究内容与成果。对 H.26X 无损编码中帧内预测、编码块划分过程进行分析，研究其存在的缺陷。针对缺陷提出了 L-IP、L-BP、R-MED 以及 L-BPIP 共 4 个优化算法，并分别进行了测试分析，编码效率最大提高超过 10%，证明了算法的有效性。下一章将介绍算法的硬件实现。

第四章 H.265 帧内无损编码及优化算法硬件实现

在视频实时编码的场景中,使用专用集成电路 (Application Specific Integrated Circuit, ASIC) 实现可做到功耗与性能的良好平衡,是业界的主流做法。本章介绍 H.265 帧内无损编码的 ASIC 实现,并给出行为级仿真结果。

4.1 硬件系统框架

如图 4.1 所示, H.265 帧内无损编码器包含的主要模块有: 帧内预测模块 (分为 2 个阶段进行)、熵编码模块、环路后处理模块、变换量化模块 (无损编码时跳过处理) 以及与外部存储器交互的 Fetch 与像素 Buffer 模块,同时存在大量用于待编码系数的中间缓存模块,不一一列出。

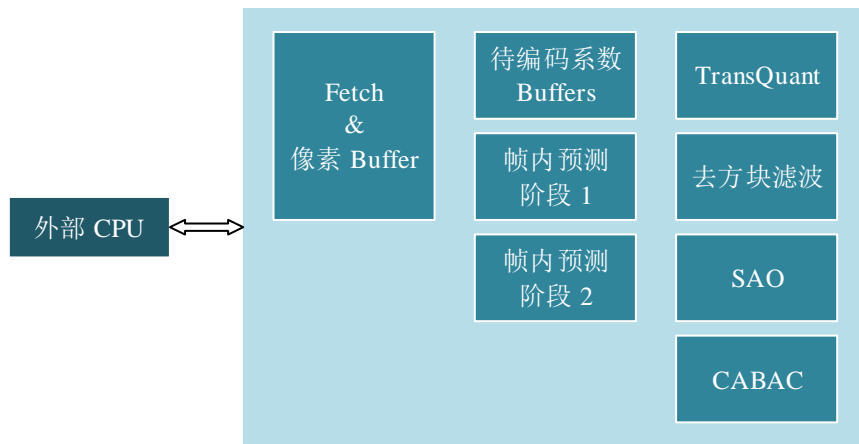


图 4.1 H.265 编码器硬件系统框架

图中的外部 CPU 并非系统框架内的模块,系统框架预留了与外部控制器的交互接口,用于进行诸如感兴趣区域 (Range of Interest, ROI)、码率控制、时延控制等外部控制操作。像素缓存模块用于存储编码过程中的原始图像、重建图像、环路后处理后的图像以及参考帧图像,同时处理核心编码模块的寻址信息。帧内预测分为 2 个阶段完成,目的是提高系统吞吐率减少流水线停滞,具体将在 4.2 节进行说明。去方块滤波、SAO 以及重建模块耦合性较强,放置在一个顶层模块中处理。

H.265 硬件编码器的最顶层模块及接口如图 4.2 所示，包含全局时钟/复位、系统控制接口、码流输出接口、外部存储器访问接口以及图中未列出的 debug 接口。

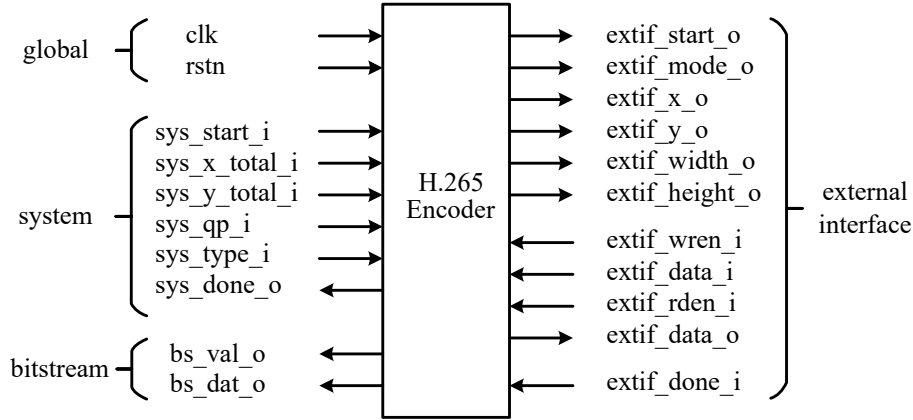


图 4.2 H.265 编码器顶层接口

4.2 关键模块的硬件实现

根据前文所述，H.265 标准下在进行帧内预测时，需要对一个 PU 内的 35 种预测模式进行扫描，从而判断哪种模式是最优的。然而在硬件实现中出现一个问题：在进行预测之前，需要获取其经过重建的参考像素，而参考像素大概率位于前一个 PU 之中，因此，为了等待重建需要让整个预测处理在时序中长时间停留，导致系统吞吐率和硬件的利用率降低^[53]，如图 4.3(a) 所示。这一情况在处理大尺寸预测单元时尤为明显。基于上述分析，使用了如图 4.3(b) 所示的硬件

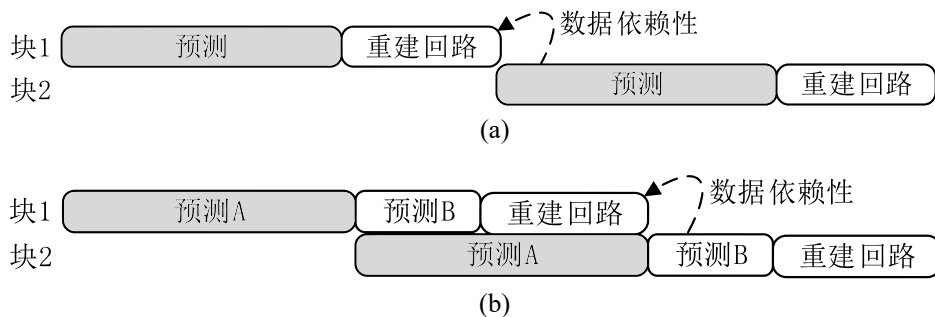


图 4.3 帧内预测时序说明

架构，将预测流程拆分为 2 部分进行：预测 A 使用原始像素而非重建像素作为

参考点，来扫描所有的预测模式找出最优的一种，预测 B 与传统预测一致，使用重建像素作为参考点进行预测，但不进行扫描而是使用预测 A 找到的最佳模式直接计算结果，后续的残差计算、熵编码也是以预测 B 的输出结果为准进行的。尽管预测 A 得到的最佳模式可能与标准结果存在出入，但经过大量测试证明该影响可忽略不计，容忍这极少量的编码效率损失可换来更加流畅的系统流水线和更高的吞吐率，显然是值得的。最终该帧内预测模块的顶层结构如图 4.4 所示。

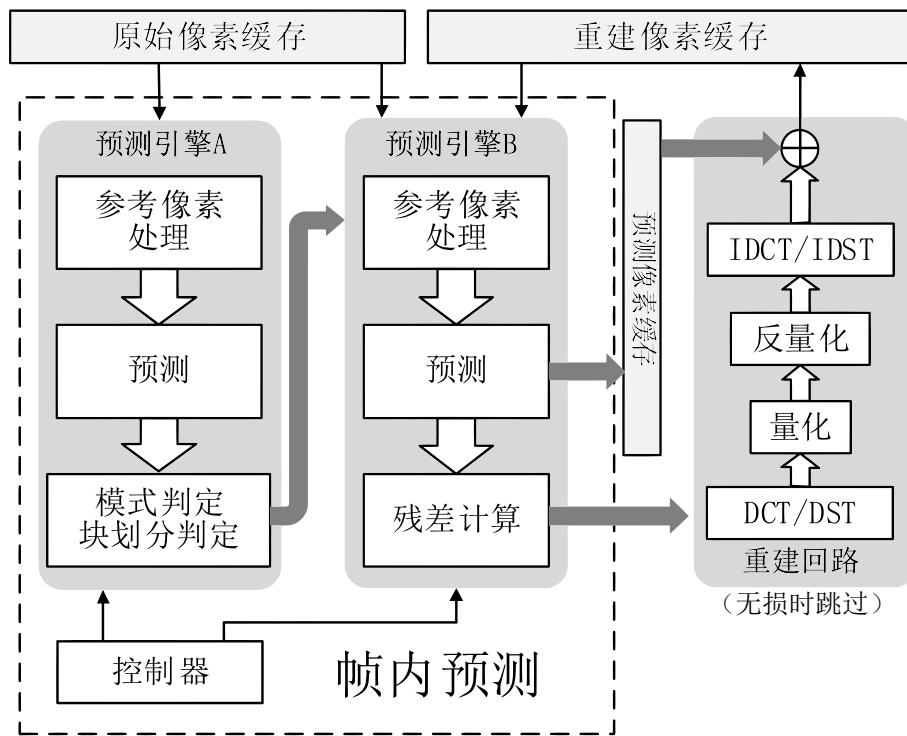


图 4.4 帧内预测模块顶层视图

4.3 行为级仿真与测试

对所设计编码器进行行为级仿真，以验证其正确性。系统验证方案如图 4.5 所示。准备一已降采样为 YUV4:2:0 的测试视频序列，提前使用标准软件编码器进行编码，由于 H.265 标准中还规定了网络适配层码流的语法语义，而硬件编码器仅输出视频编码层码流，因此需要编写脚本抽取出软件结果的视频编码层码流作为对照数据。运行硬件编码器的行为级仿真，将其配置成与软件编码时的参数完全一致的状态，逐 CTU 编码测试序列，同时将编码结果的码流逐字节

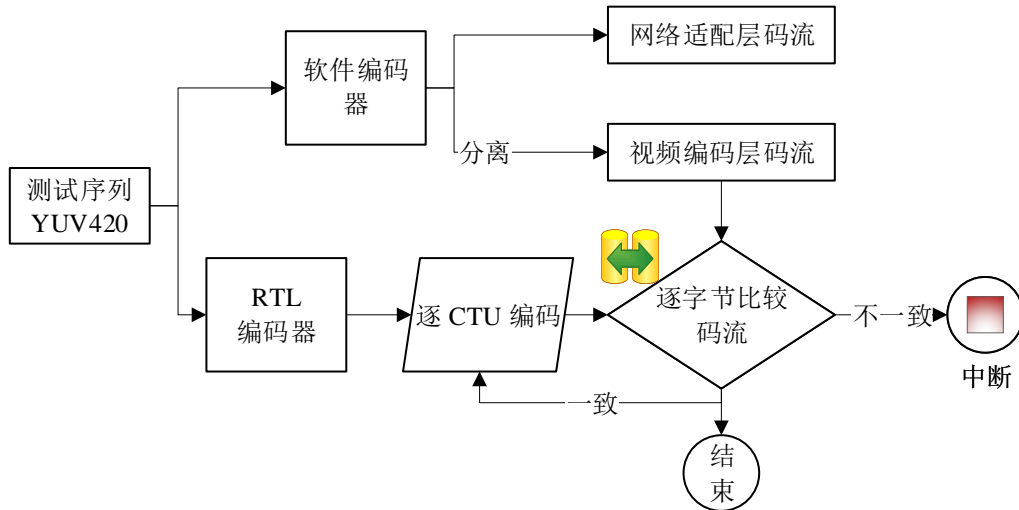


图 4.5 硬件编码器行为级仿真方案

与对照数据进行比较，一旦出现不一致的情况即中断仿真流程。

行为级仿真在 Synopsys VCS 软件平台进行，同时使用 perl、make 等脚本进行辅助验证，实现全自动的行为级仿真验证。仿真结果如图 4.6 所示。

```

*** TEST I FRAMES ! ***

at 10660075, Frame Number = 00, mb_x_first = 00, mb_y_first = 00
at 10778975, Frame Number = 00, mb_x_first = 01, mb_y_first = 00
at 10876295, Frame Number = 00, mb_x_first = 02, mb_y_first = 00
...
...
...
at 40551735, Frame Number = 09, mb_x_first = 06, mb_y_first = 03
at 40638475, Frame Number = 09, mb_x_first = 00, mb_y_first = 00
at 40680215, Frame Number = 10, mb_x_first = 00, mb_y_first = 00

*** CHECK FINISHED ! ***

Simulation complete via $finish(1) at time 40682215 NS + 0
./tb_top.v:578      $finish ;
  
```

图 4.6 硬件编码器行为级仿真屏幕截图

最后统计应用前文所述联合优化算法前后的行为级仿真编码结果，比较其编码所用时钟数及输出码流字节数。由于硬件行为级仿真耗时较长，仅统计 ClassA~ClassF 6 类测试序列中各一个序列的仿真结果，如表 4.1 所示。

统计结果证明硬件实现的优化算法与软件测试结果在编码效率上相当，在编码时间上由于硬件架构的并行性，增加的时钟数百分比少于软件测试结果。

最后，值得一提的是本课题尝试着将所设计编码器进行初步的硬件综合（未进行布局布线等后续处理），在 TSMC 65nm 的工艺下以 400MHz 的时钟进行综

表 4.1 H.265 硬件编码器优化算法行为级仿真性能统计

序列类别	序列	码率	编码时间 (时钟数)
ClassA	PeopleOnStreet	-11.86%	118.84%
ClassB	Kimono	-8.14%	120.73%
ClassC	BQMall	-6.53%	120.59%
ClassD	BlowingBubbles	-4.94%	117.88%
ClassE	KristenAndSara	-13.04%	117.53%
ClassF	ChinaSpeed	-17.15%	114.50%
ClassA~E 均值		-8.90%	119.11%
ClassA~F 均值		-10.28%	118.35%

合，综合结果显示编码器共使用 1399k 个等效逻辑门（未包含外部交互用的存储器）。

4.4 本章小结

本章描述 H.265 帧内无损编码器的 ASIC 实现。介绍了编码器硬件系统架构及其中关键模块的硬件设计，最后给出的行为级仿真验证了编码器的正确性，资源利用报告亦显示该编码器可 ASIC 化。下一章将介绍课题中设计、使用的 FPGA 原型验证平台。

第五章 H.265 帧内无损编码 FPGA 原型验证平台

当电路系统复杂度高于一定程度时，传统的行为级仿真已经难以满足设计的验证需求，使用 FPGA 进行原型验证灵活度高，验证结果可靠，且有利于软硬件协同设计、加快项目开发进度。因此 FPGA 原型验证成为了复杂系统 ASIC 化之前不可或缺的一个步骤。本课题亦完成了一 FPGA 原型验证平台，对硬件编码系统进行验证。

5.1 FPGA 原型验证方案

为了对设计的硬件编码器进行验证，本课题设计、搭建了一个完整的 FPGA 原型验证平台。该验证平台由视频源、预处理模块、通信模块、硬件编码器和上位机解码模块组成。利用该平台可实时编码由视频源提供的图像信息，经映射入 FPGA 的硬件编码器编码后将码流传回上位机，在上位机对码流进行网络适配层码流的拼接后，由上位机软解码，实时查看软解码结果正确与否即可对硬件编码器进行验证。完整的验证平台如图 5.1 所示。

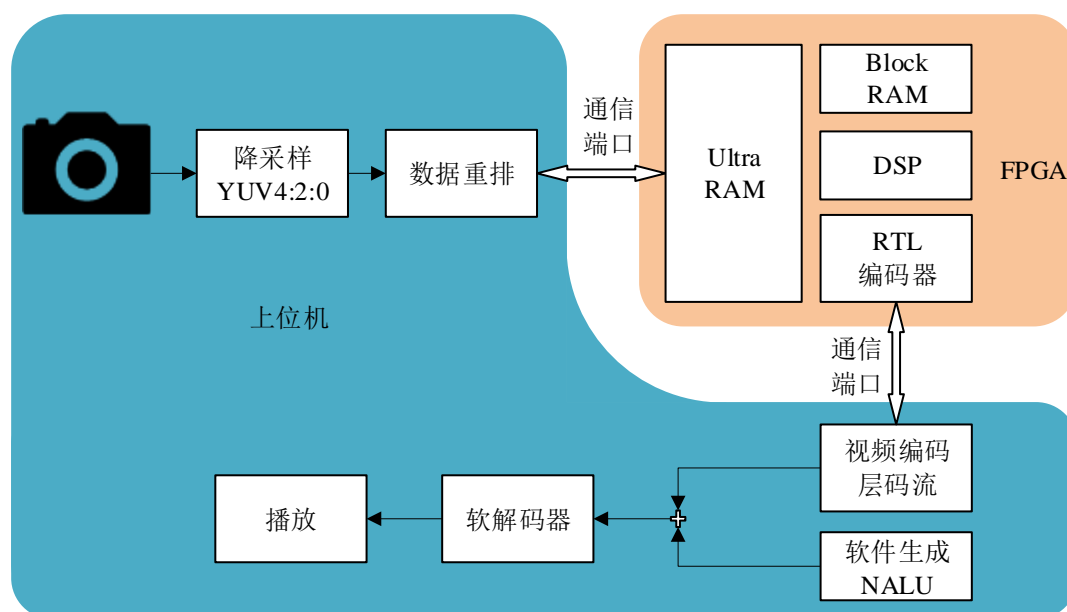


图 5.1 FPGA 原型验证平台

5.2 FPGA 验证平台关键模块

验证平台由上位机与 FPGA 共同组成，上位机进行视频源的获取与预处理，同时负责对 FPGA 输出的码流进行加工、软解码，最终实时播放；FPGA 负责映射硬件编码器，其中使用大量的专用存储器资源实现数据缓存和交换，以保证留出足够的查找表实现编码逻辑的映射。通信端口由上位机和 FPGA 共同处理，根据通信速率的需求可选择串口、以太网口甚至 PCI-E 接口进行通信。

5.2.1 FPGA 器件

FPGA 原型验证平台所使用的器件是基于 XILINX Zynq UltraScale+ MPSoCs 开发平台的开发板 AXU15EG。主芯片为 XCZU15EG-2FFVB1156I，外挂 2GB DDR4。底板扩展了外围接口，本课题所设计的部分包括 USB3.0 接口、千兆以太网接口、UART 接口、FMC 接口、RS485 接口以及 MIPI 接口。所用器件的完整结构如图 5.2 所示。

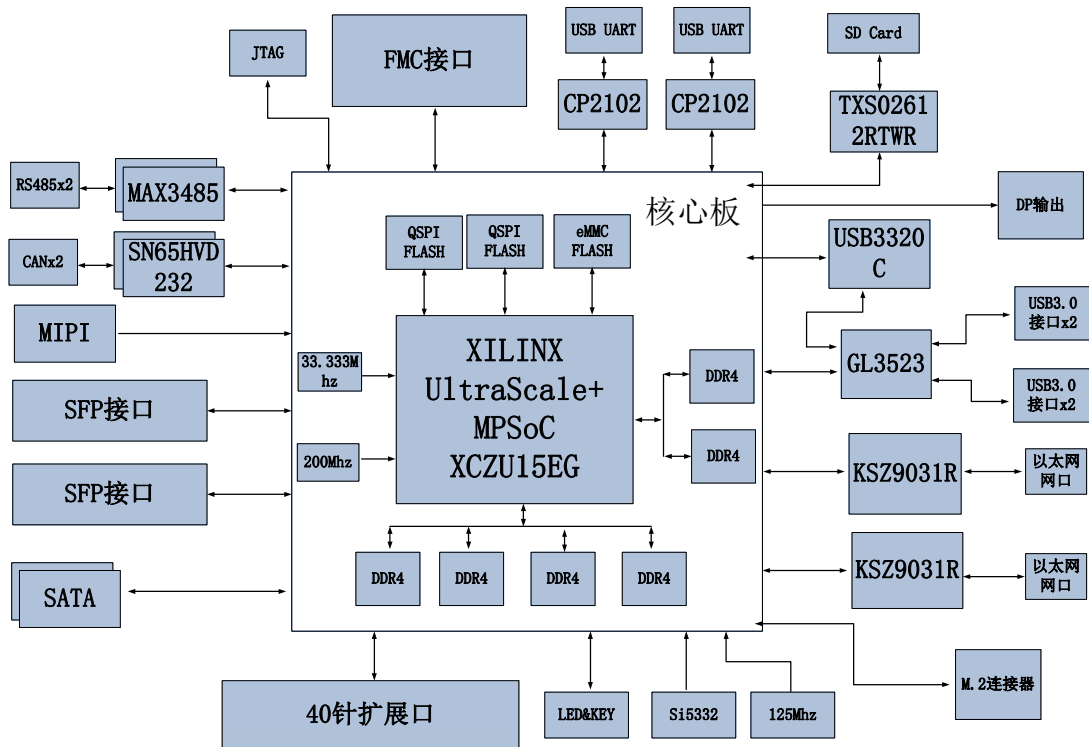


图 5.2 FPGA 器件完整结构

5.2.2 视频源及预处理

原型验证平台的待编码视频源可有多种形式，灵活适用于不同场景。

1. 视频文件

使用上位机中存储的视频文件作为视频源。其方便提供、数据内容固定的特性使该方式多用于严谨的编码器性能统计、分析。另外，由于不存在与其余硬件的交互，可将开发过程中的不确定性控制在最小，因此也多用于 FPGA 的仿真验证阶段。

2. HDMI 输入

使用显卡或工业相机的 HDMI 输出流作为视频源。FPGA 可与 FMC 视频扩展子板组合使用，直接将 HDMI 差分信号转换为 RGB 信号。但由于 HDMI 信号几乎无法在上位机进行预处理，需要在硬件部分进行降采样、重排序等操作，对前期平台搭建十分不利，无法定位问题出现的模块。因此该方案一般用于验证平台成熟后的扩展。

3. USB 相机输入

使用 USB 相机的输出流作为视频源。USB 相机可通过直接连接在上位机的形式提供视频源，并且直接在上位机进行各类预处理，以保证 FPGA 器件上的工作只有编码和通信，最大程度上地减少了硬件验证的不确定性。此外，由于视频源率先存储在上位机，故可将待编码帧暂存，待收到 FPGA 器件的输出码流后进行软解码并与视频源做对比，统计出 PSNR 等各类信息。综上所述，FPGA 验证平台选用 USB 相机作为输入。

预处理是上位机需要进行的操作之一。预处理包括色彩格式转换、降采样和数据重排。

1. 色彩格式转换

大部分视频输出设备的输出色彩格式是 RGB，因为 RGB 通道格式最适合各类显示器显示。但在视频编解码领域，最常使用的是 YUV 色彩通道，即亮度-色差通道。因为 YUV 格式天然地将人眼不同敏感程度的色彩分量区分了出来，因此可在编码过程中人为的将色差通道进行高失真高编码效率的处理而不产生明显的视觉损失。色彩格式转换在 FPGA 平台上实现需要

消耗大量的中间缓存存储器，为了给复杂的 H.265 编码器留出资源，将色彩格式转换转移到上位机中进行。

2. 降采样

通常来说，为了最大程度地提高编码效率，视频编码器直接处理的数据会经过降采样，即常见的 Y:U:V=4:2:0 格式，将色差通道的数据在水平和垂直方向上各降采样 1/2，经过降采样后待编码的数据量直接减少了 50% 而不会产生明显的视觉损失。这一操作并不复杂，仅涉及到部分数据的舍弃，但与色彩格式转换类似需要消耗一定的存储器资源。且受到通信接口速率的限制，在 FPGA 平台上进行降采样直接影响了验证平台的吞吐率，因此将降采样转移到上位机中进行。

3. 数据重排

通常来说，测试视频序列与视频源输出存储的文件中，像素值的排列顺序属于平面模式，即在内存空间中按照 YYYYYYYY....UUUU..VVVV.. 的顺序，依次存放 3 个色彩空间的数据。然而在图像编解码中，由于其以编码块为基本单位的架构，编码器在处理数据时往往是按照块状空间进行的，且 YUV 3 个色彩空间的数据要求交替出现。因此硬件视频编码器中总是需要一块极大的存储器缓存视频源提供的数据。同样出于减轻 FPGA 不必要的资源消耗的目的，本原型验证平台将数据重排的任务交给上位机进行，数据通过通信端口进入 FPGA 后几乎只需要按照一般队列的模式顺序读取即可，极大地减小了 FPGA 资源消耗与 debug 难度。

5.2.3 与上位机通信接口

通信接口是上位机与 FPGA 开发板通信的唯一渠道。根据使用场景的需求，可选择使用串口（USB2UART）或以太网口。

1. 串口（USB2UART）

UART 是极为经典的一种简单通信接口，以串行方式进行数据通信，以 1 周期的低电平表示数据开始发送。目前传统意义上的串口（RS232）因波特率过低，在学术界已很少使用，原型验证平台中 FPGA 的串口是通过 CP2102 转换芯片由 USB2.0 转换而来，因此极限波特率能达到 1.5MBps，

已经能够满足基本的图像传输。因其控制逻辑简单，debug 流程简洁，在非超高清场景下进行原型验证时可以选择串口通信的方式。

2. 以太网口

以太网口通信是目前 FPGA 与上位机进行高速通信的主流做法之一。通信速率由 10/100/1000 Mbps 3 种模式，主流做法是选择 100 Mbps 进行与上位机的通信。原型验证平台中使用基于 UDP 协议的方案建立通信，相比 TCP 要简单不少。此外网络通信产生的数据很容易在上位机通过网络抓包程序、脚本进行处理，非常适合在学术研究中应用。因此，在进行超高分辨率场景下的原型验证时选择以太网口进行通信。

5.2.4 编码模块

编码模块与第 4 章相比，在数字逻辑上并未产生区别。但考虑到视频编码器将消耗大量的数字逻辑资源，为了让 FPGA 有限的查找表资源被高效地利用，对编码器中大部分的存储器做强制手动替换处理，即显式地用 Xilinx RAM IP 替换能够用存储器逻辑实现的模块。根据存储空间的大小，使用 3 类不同的 RAM IP 进行替换，对于极小存储器使用分布式 RAM 实现，以不至于过多浪费专用 RAM；对于超过 4 kbits 的存储器使用 Block RAM 实现，Block RAM 是 FPGA 核心中的专用存储器资源，最小的例化单位是 18 kbits；对于极大存储器使用 Ultra RAM 实现，与 Block RAM 类似，但单块 URAM 即为 128 kbits。经过上述处理后编码模块以使用大量专用存储单元的代价，换来减少 20% 以上查找表资源消耗的效果。

5.2.5 上位机解码模块

FPGA 原型验证通过的标志之一是在上位机成功解码出收到的硬件编码码流。这里需要引出在前文从未讨论过的内容：网络适配层 (Network Abstract Layer, NAL) 码流。H.26X 中不仅规定了视频编码的方法、语法规义等，还规定了其在通信过程中的数据帧格式，即在熵编码码流之前仍有一部分头信息，即为 NAL 码流。NAL 中承载了视频参数数据、序列参数集、图像参数集以及 Slice 头信息，如图 5.3 所示。

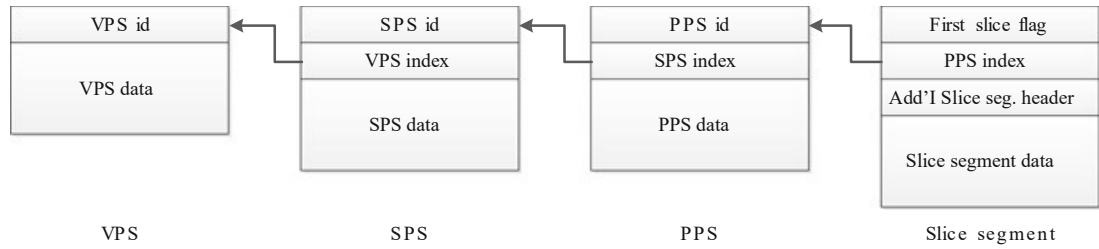


图 5.3 H.265 网络适配层承载信息

为了保证硬件编码器的简洁，所实现的编码器仅输出视频编码层码流，将 NAL 单元的处理工作交给外部 CPU 进行处理，目前暂未实现。因此上位机接收到的码流无法直接进行解码，故解码模块有另一辅助程序参与，即 NAL 码流生成、提取模块。对相同配置参数软编码出的码流进行分析，提取出其中 NAL 码流将其拼接在 FPGA 输出码流的相应位置，得到的拼接文件即可在上位机正常解码，如图 5.1 上位机解码部分所示。

5.3 FPGA 原型验证结果

FPGA 原型验证平台在 XCZU15EG-2FFVB1156I 上映射占用的资源如表 5.1 所示。

表 5.1 编码器 FPGA 映射资源使用情况

资源	数量	使用率
LUT	250660	73.45%
FF	9200	1.35%
BRAMs	99	13.27%
DSP	463	13.11%
LUTRAM	11680	3.42%

FPGA 原型验证平台实物如图 5.4 所示，图中包含摄像机、FPGA 开发板和主机显示器，其中摄像头与 FPGA 开发板上的 USB2UART 均通过 USB2.0 连接到显示器所对应的主机上。显示器中显示实时编、解码后的摄像头采集画面。

与算法软件验证、行为级仿真验证不同，FPGA 原型验证采用编码实时摄像机获取的图像的方式，通过对比硬件编码输出码流的字节数进行验证。验证中

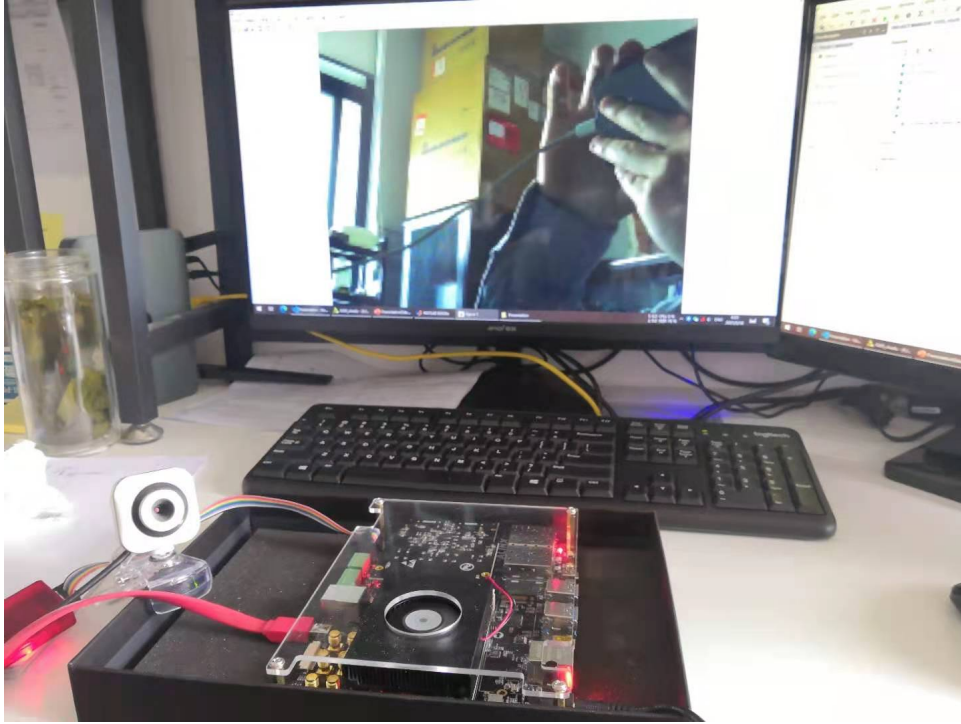


图 5.4 FPGA 原型验证平台实物

选取了室内、建筑物内走廊、空旷室外和人像 4 个典型场景，验证结果如表 5.2 所示。

表 5.2 FPGA 原型验证平台实际环境测试结果

场景	码率	编码时间
室内	-12.80%	118%
建筑物内走廊	-9.11%	122%
空旷室外	-12.91%	117%
人像	-11.02%	118%
平均值	-11.46%	119%

验证结果普遍比测试序列优化程度更高，是因为测试序列大多属于较难编码的界面，编码块划分中很少出现大尺寸单元，不利于 L-BPIP 算法的发挥。

5.4 本章小结

本章介绍了课题研究过程中搭建的 FPGA 原型验证平台。使用 FPGA 进行原型验证灵活度高，验证结果可靠，且有利于软硬件协同设计、加快项目开发进度。该验证平台由视频源、预处理模块、通信模块、硬件编码器和上位机解码模块组成。利用该平台可实时编码由视频源提供的图像信息，经映射入 FPGA 的硬件编码器编码后将码流传回上位机软解码，实时查看软解码结果正确与否即可对硬件编码器进行验证。

第六章 总结与展望

6.1 课题总结

本课题的目的是提高 H.265 无损帧内编码效率。为此,本课题对 H.265 的帧内预测过程、编码块划分过程进行了详尽的分析,提出 4 个优化算法并进行了软、硬件实现。测试验证结果显示达到了预期的性能指标。本课题的主要工作内容总结如下:

1. 分析帧内预测过程及原理,针对其部分待预测像素距离参考点距离过远的缺陷,提出了一种 L 形迭代预测算法,将待预测像素与参考点的距离缩短到 1,极大地提高了帧内预测的准确性,从而提高编码效率;
2. 分析编码块划分过程及原理,提出了一种不同于 H.265 基于四叉树结构的分块方法,即 L 形分块算法。同时提出了将其与 L 形迭代预测算法融合的联合算法,进一步提高编码效率;
3. 分析系数的编码过程,利用待编码系数中存在的特殊的空间结构性,提出了一种残差中值边缘检测算法,通过对预测残差进行二次处理降低待编码系数的整体能量,进一步提高编码效率;
4. 在 H.265 硬件编码器的基础上,实现上述改进算法,同时利用 FPGA 搭建硬件编码系统原型验证平台,使用该平台可对提出的改进算法进行测试验证。

6.2 工作展望

针对 H.265 帧内编码效率的优化,本课题还有多项想要尝试的实验与优化方案,记录如下。期望本课题在同窗的后续耕耘下能够百尺竿头,更进一步。

1. 碍于迭代预测时对重建点的需求, L-IP 算法目前仅能应用在无损编码中。将其应用在有损编码的一个方案是寻找一种合适的一维变换,使能够以 L 形的结构完成变换-量化-去量化-逆变换回路。如何找到一种去相关性能良

好的一维变换是亟待解决的问题；

2. R-MED 算法经过简单测试发现其在大分辨率图像或屏幕图像的有损编码中能得到 2%~3% 的编码效率优化。有损编码中待编码系数已不具有常规的空间相关性，此时 R-MED 仍能改进编码效率，其中机理需要进一步分析梳理，经过后续改进后有望将其应用到有损编码中。类似地，在帧间编码中运动补偿残差又有其独特的数据相关性，能否在其上应用 R-MED 值得做实验尝试；
3. 本课题提出的算法理论上均可应用在最新的 H.266 标准中。H.266 在帧内预测和编码块划分中加入了大量新机制，如何将所提算法与新机制有机结合，使其在新标准中得到更高的编码效率优化是一个值得研究的内容。

插图索引

图 2.1	H.26X 标准视频编码框架	7
图 2.2	H.26X 预测编码的基本流程	7
图 2.3	选择不同的帧内预测模式	8
图 2.4	H.26X 帧间预测	9
图 2.5	H.26X 中变换量化过程的拆分与合并	12
图 2.6	去方块滤波应用场景及效果	13
图 2.7	量化失真引起的振铃效应	14
图 2.8	SAO 边界补偿分类	15
图 2.9	H.265 待编码系数扫描顺序	18
图 3.1	H.265 帧内预测角度模式	25
图 3.2	H.265 角度预测 1/32 精度投影	25
图 3.3	H.265 帧内预测结果示例	26
图 3.4	H.265 帧内预测所有模式残差分布可视图	27
图 3.5	L-IP 算法	29
图 3.6	基于四叉树的分块算法与 L-BP 算法	34
图 3.7	应用 L-BP 算法后的分块决策流程	36
图 3.8	KristenAndSara 第一帧及其帧内预测残差（亮度）	38
图 3.9	H.26X 无损编码方案及 R-MED 应用方案	38
图 3.10	R-MED 的预测策略	39
图 3.11	一个经过 R-MED 处理的预测单元	40

图 3.12	KristenAndSara 第一帧残差经 R-MED 处理后效果（亮度）	41
图 3.13	L-BPIP 算法 RDO 过程的比较对象	45
图 3.14	L-BPIP 处理顺序说明	46
图 3.15	H.265 标准与 L-BPIP 编码块划分结果对比	48
图 4.1	H.265 编码器硬件系统框架	50
图 4.2	H.265 编码器顶层接口	51
图 4.3	帧内预测时序说明	51
图 4.4	帧内预测模块顶层视图	52
图 4.5	硬件编码器行为级仿真方案	53
图 4.6	硬件编码器行为级仿真屏幕截图	53
图 5.1	FPGA 原型验证平台	55
图 5.2	FPGA 器件完整结构	56
图 5.3	H.265 网络适配层承载信息	60
图 5.4	FPGA 原型验证平台实物	61

表格索引

表 2.1	k 阶指数哥伦布码	16
表 3.1	H.265 与 L-IP 各尺寸 PU 平均残差幅值统计	30
表 3.2	H.265 与 L-IP 各尺寸 PU 占比统计	31
表 3.3	L-IP 算法性能测试	32
表 3.4	L-BP 算法性能测试	37
表 3.5	应用 H.265+R-MED 算法后残差能量的变化率	41
表 3.6	R-MED 算法应用在 H.265 的性能测试	42
表 3.7	R-MED 算法应用在 H.266 的性能测试	43
表 3.8	R-MED 与其他算法的性能比较（对比 H.265）	44
表 3.9	L-BPIP 算法性能测试	47
表 4.1	H.265 硬件编码器优化算法行为级仿真性能统计	54
表 5.1	编码器 FPGA 映射资源使用情况	60
表 5.2	FPGA 原型验证平台实际环境测试结果	61

参考文献

- [1] Bross B, Chen J, Ohm J-R, et al. Developments in International Video Coding Standardization After AVC, With an Overview of Versatile Video Coding (VVC) [J]. *Proceedings of the IEEE*, 2021: 1–31. doi: 10.1109/JPROC.2020.3043399.
- [2] Sullivan G J, Ohm J-R, Han W-J, et al. Overview of the High Efficiency Video Coding (HEVC) Standard [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22 (12): 1649–1668. doi: 10.1109/TCSVT.2012.2221191.
- [3] Marpe D, Wiegand T, Sullivan G J. The H.264/MPEG4 Advanced Video Coding Standard and Its Applications [J]. *IEEE Communications Magazine*, 2006, 44 (8): 134–143. doi: 10.1109/MCOM.2006.1678121.
- [4] De-Luxán-Hernández S, Venugopal G, George V, et al. A Fast Lossless Implementation of the Intra Subpartition Mode for VVC [C]. In *2020 IEEE International Conference on Image Processing (ICIP)*, October 2020: 1118–1122. doi: 10.1109/ICIP40778.2020.9191103.
- [5] Lee S-W, Kim H-Y. An Energy-Efficient Low-Memory Image Compression System for Multimedia IoT Products [J]. *EURASIP Journal on Image and Video Processing*, 2018, 2018 (1): 87. doi: 10.1186/s13640-018-0333-3.
- [6] Mrak M, Xu J. Improving Screen Content Coding in HEVC by Transform Skipping [C]. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, August 2012: 1209–1213.
- [7] Kamisli F. Lossless Image and Intra-Frame Compression with Integer-to-Integer DST [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019, 29 (2): 502–516. doi: 10.1109/TCSVT.2017.2787638.
- [8] Zhang K, Chen J, Zhang L, et al. Enhanced Cross-Component Linear Model for Chroma Intra-Prediction in Video Coding [J]. *IEEE Transactions on Image Processing*, 2018, 27 (8): 3983–3997. doi: 10.1109/TIP.2018.2830640.
- [9] Xu J, Joshi R, Cohen R A. Overview of the Emerging HEVC Screen Content Coding Extension [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016, 26 (1): 50–62. doi: 10.1109/TCSVT.2015.2478706.
- [10] Chen C-C, Peng W-H. Intra Line Copy for HEVC Screen Content Intra-Picture Prediction [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, 27 (7): 1568–1579. doi: 10.1109/TCSVT.2016.2543098.

- [11] Sanchez V, Aulí-Llinàs F, Bartrina-Rapesta J, et al. HEVC-Based Lossless Compression of Whole Slide Pathology Images [C]. In 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), December 2014: 297–301. doi: 10.1109/GlobalSIP.2014.7032126.
- [12] Sanchez V, Llinàs F A, Rapesta J B, et al. Improvements to HEVC Intra Coding for Lossless Medical Image Compression [C]. In 2014 Data Compression Conference, March 2014: 423–423. doi: 10.1109/DCC.2014.76.
- [13] Wige E, Yammine G, Amon P, et al. Sample-Based Weighted Prediction with Directional Template Matching for HEVC Lossless Coding [C]. In 2013 Picture Coding Symposium (PCS), December 2013: 305–308. doi: 10.1109/PCS.2013.6737744.
- [14] Zhou M, Budagavi M. RCE2: Experimental Results on Test 3 and Test 4, Joint Collaborative Team on Video Coding - M0056 version 1 [R]. 2013.
- [15] Zhou M, Gao W, Jiang M, et al. HEVC Lossless Coding and Improvements [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22 (12): 1839–1843. doi: 10.1109/TCSVT.2012.2221524.
- [16] Sanchez V, Aulí-Llinàs F, Serra-Sagristà J. Piecewise Mapping in HEVC Lossless Intra-Prediction Coding [J]. IEEE Transactions on Image Processing, 2016, 25 (9): 4004–4017. doi: 10.1109/TIP.2016.2571065.
- [17] 元辉, 常义林, 卢朝阳, 等. 一种降低预测模式开销的帧内预测方法 [J]. 西安电子科技大学学报, 2010, 37 (6): 981. doi: 10.3969/j.issn.1001-2400.2010.06.001.
- [18] Li J, Li B, Xu J, et al. Efficient Multiple-Line-Based Intra Prediction for HEVC [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2018, 28 (4): 947–957. doi: 10.1109/TCSVT.2016.2633377.
- [19] Chen Z, Shi J, Li W. Learned Fast HEVC Intra Coding [J]. IEEE Transactions on Image Processing, 2020, 29: 5431–5446. doi: 10.1109/TIP.2020.2982832.
- [20] Li J, Li B, Xu J, et al. Fully Connected Network-Based Intra Prediction for Image Coding [J]. IEEE Transactions on Image Processing, 2018, 27 (7): 3236–3247. doi: 10.1109/TIP.2018.2817044.
- [21] Omari M, Yaichi S. Image Compression Based on Genetic Algorithm Optimization [C]. In 2015 2nd World Symposium on Web Applications and Networking (WSWAN), March 2015: 1–5. doi: 10.1109/WSWAN.2015.7210304.
- [22] Pakdaman F, Adelimanesh M A, Gabbouj M, et al. Complexity Analysis Of Next-Generation VVC Encoding And Decoding [C]. In 2020 IEEE International Conference on Image Processing (ICIP), October 2020: 3134–3138. doi: 10.1109/ICIP40778.2020.9190983.

- [23] Bossen F, Sharman K, Suehring K, et al. H.264/AVC JM Reference Software [CP/OL]. [2021-04-23]. <https://vcgit.hhi.fraunhofer.de/jvet/JM>.
- [24] Bossen F, Suehring K, Iwamura S, et al. H.265/HEVC HM Reference Software [CP/OL]. [2021-03-18]. <https://vcgit.hhi.fraunhofer.de/jvet/HM>.
- [25] Bossen F, Gallasso M P, Wiecekowski A, et al. H.266/VVC VTM Reference Software [CP/OL]. [2021-02-01]. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM.
- [26] Joy H K, Kounte M R. A Comprehensive Review of Traditional Video Processing [J]. Advances in Science, Technology and Engineering Systems Journal, 2020, 5 (6): 274–279. doi: 10.25046/aj050633.
- [27] Sze V, Budagavi M, Sullivan G J. High Efficiency Video Coding (HEVC): Algorithms and Architectures [M]. Cham: Springer International Publishing, 2014. doi: 10.1007/978-3-319-06895-4.
- [28] 万帅, 杨付正. 新一代高效视频编码 H.265/HEVC: 原理、标准与实现 [M]. 第 1 版. 北京: 电子工业出版社, 2017.
- [29] Lainema J, Bossen F, Han W-J, et al. Intra Coding of the HEVC Standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2012, 22 (12): 1792–1801. doi: 10.1109/TCSVT.2012.2221525.
- [30] Siqueira I, Correa G, Grellert M. Rate-Distortion and Complexity Comparison of HEVC and VVC Video Encoders [C]. In 2020 IEEE 11th Latin American Symposium on Circuits & Systems (LASCAS), San Jose, Costa Rica, February 2020: 1–4. doi: 10.1109/LASCAS45839.2020.9069036.
- [31] Purnachand N, Alves L N, Navarro A. Improvements to TZ Search Motion Estimation Algorithm for Multiview Video Coding [C]. In 2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP), April 2012: 388–391.
- [32] Rao K R, Kim D N, Hwang J J. Fast Fourier Transform - Algorithms and Applications [M]. Springer Netherlands, 2010. doi: 10.1007/978-1-4020-6629-0.
- [33] Lam E Y, Goodman J W. A Mathematical Analysis of the DCT Coefficient Distributions for Images [J]. IEEE Transactions on Image Processing, 2000, 9 (10): 1661–1666. doi: 10.1109/83.869177.
- [34] Zeng Y, Cheng L, Bi G, et al. Integer DCTs and Fast Algorithms [J]. IEEE Transactions on Signal Processing, 2001, 49 (11): 2774–2782. doi: 10.1109/78.960425.
- [35] Saxena A, Fernandes F C. DCT/DST-Based Transform Coding for Intra Prediction in Image/Video Coding [J]. IEEE Transactions on Image Processing, 2013, 22 (10): 3974–3981. doi: 10.1109/TIP.2013.2265882.

- [36] Stockman A, Sharpe L T. The Spectral Sensitivities of the Middle- and Long-Wavelength-Sensitive Cones Derived from Measurements in Observers of Known Genotype [J]. *Vision Research*, 2000, 40 (13): 1711–1737. doi: 10.1016/S0042-6989(00)00021-3.
- [37] Horé A, Ziou D. Image Quality Metrics: PSNR vs. SSIM [C]. In 2010 20th International Conference on Pattern Recognition, August 2010: 2366–2369. doi: 10.1109/ICPR.2010.579.
- [38] Norkin A, Bjontegaard G, Fuldseth A, et al. HEVC Deblocking Filter [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22 (12): 1746–1754. doi: 10.1109/TCSVT.2012.2223053.
- [39] Fu C-M, Alshina E, Alshin A, et al. Sample Adaptive Offset in the HEVC Standard [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22 (12): 1755–1764. doi: 10.1109/TCSVT.2012.2221529.
- [40] Xue Y, Wang Y. A Novel Video Coding Framework Using a Self-Adaptive Dictionary [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 28 (12): 3478–3491. doi: 10.1109/TCSVT.2017.2760143.
- [41] Chen Y, Wan G C, Xia Z W, et al. A Hardware Design Method for Canonical Huffman Code [C]. In 2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL), November 2017: 2212–2215. doi: 10.1109/PIERS-FALL.2017.8293507.
- [42] Sole J, Joshi R, Nguyen N, et al. Transform Coefficient Coding in HEVC [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2012, 22 (12): 1765–1777. doi: 10.1109/TCSVT.2012.2223055.
- [43] Sullivan G, Wiegand T. Rate-Distortion Optimization for Video Compression [J]. *IEEE Signal Processing Magazine*, 1998, 15 (6): 74–90. doi: 10.1109/79.733497.
- [44] Wieckowski A, Brandenburg J, Schmidt L, et al. Fraunhofer Versatile Video Encoder (VVenC) [CP/OL]. [2021-04-28]. <https://github.com/fraunhoferhhi/vvenc>.
- [45] Liu X, Liu Y, Wang P, et al. An Adaptive Mode Decision Algorithm Based on Video Texture Characteristics for HEVC Intra Prediction [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017, 27 (8): 1737–1748. doi: 10.1109/TCSVT.2016.2556278.
- [46] Fan K, Wang R, Li G, et al. Efficient Prediction Methods with Enhanced Spatial-Temporal Correlation for HEVC [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019, 29 (12): 3716–3728. doi: 10.1109/TCSVT.2018.2885002.
- [47] Bossen F. Common HM Test Conditions and Software Reference Configurations, Joint Collaborative Team on Video Coding - L1100 version 1 [R]. 2013.
- [48] Choi B-D, Ko S-J. Split-and-Merge-Based Block Partitioning for High Efficiency Image Coding [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 28 (2): 540–549. doi: 10.1109/TCSVT.2016.2615498.

- [49] Kamisli F, Lim J S. 1-d Transforms for the Motion Compensation Residual [J]. IEEE Transactions on Image Processing, 2011, 20 (4): 1036–1046. doi: 10.1109/TIP.2010.2083675.
- [50] Seung-Hwan Kim, Je-Won Kang, Kuo C-C J. Improved H.264/AVC Lossless Intra Coding with Two-Layered Residual Coding (TRC) [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2011, 21 (7): 1005–1010. doi: 10.1109/TCSVT.2011.2133170.
- [51] Weinberger M J, Seroussi G, Sapiro G. The LOCO-i Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS [J]. IEEE Transactions on Image Processing, 2000, 9 (8): 1309–1324. doi: 10.1109/83.855427.
- [52] Ma T-C, Nalci A, Nguyen T. JVET Common Test Conditions and Software Reference Configurations for Lossless, near Lossless, and Mixed Lossy/Lossless Coding, Joint Video Experts Team - Q2014 version 5 [R]. 2020.
- [53] Cong Liu, Weiwei Shen, Tianlong Ma, et al. A Highly Pipelined VLSI Architecture for All Modes and Block Sizes Intra Prediction in HEVC Encoder [C]. In 2013 IEEE 10th International Conference on ASIC, Shenzhen, China, October 2013: 1–4. doi: 10.1109/ASICON.2013.6811849.

作者在攻读硕士学位期间发表的论文与研究成果

发表的学术论文

1. 林敏, 林庆毫, 翁晓雨, 陈国捷, 应用于 H.26X 的通用无损帧内编码优化算法. 哈尔滨工业大学学报, 2021, 53(8). (EI 中文期刊, 已录用)
2. Wenwen Liu, Qinghao Lin, Jingwei Shi, Gang Wang, Haibo Liu, Min Lin. An RS (255,239) Decoder with e-PIBM Algorithm, 25.6Gb/s Throughput and 623 TSNT FoM Value Using Fractional Folding Technique in 40nm Technology. 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401210. (EI 会议, 已发表)

发明专利

1. 林敏, 林庆毫, 翁晓雨, 陈国捷. 一种基于 HEVC 的视频编码方法. CN202110151483.5, 2021-05-14.

参与研究项目

1. 国家重点研发计划 “超低功耗体域网智能节点芯片”

致 谢

衷心感谢恩师林敏教授对本人的精心指导。恩师阅历之深，涉猎之广，经验之丰富，思考问题之周全无不让我钦佩。仍记得初入研究室时，恩师便教导我们做科研不要浮躁，不要只追求一些“快知识”，要冷静下来钻到经典的、高水平的书籍和文章中去，要多了解身边同窗的研究内容、行业的发展状况。恩师的教导让我从只会看博客、查百科转变成了会看文献、会思考问题的合格的研究生。若无恩师引导鞭策，三载光阴必将落得碌碌无为。

衷心祝愿研究室同窗百尺竿头更进一步。作为研究室首届成员之一，求学之路难免有筚路蓝缕之艰辛，正是有你们的陪伴和帮助，才让我有了直面困难的勇气，打消了偷懒、放弃的念头。还要特别感谢图像组的两位同窗，正是有你们的协助我才能顺利完成此文，相信你们能带领图像组创造佳绩。

对导师、同窗致谢的同时还应对自己进行反思。求学又三载，这三年对自我缺陷的认识更多了。想学的太多，精通的太少，这对研究和将来的工作是十分不利的；自我约束能力差，疫情期间工作效率低下，浪费的时间太多；缺乏大局观，做不好长时间规划；安于现状，害怕周遭的变化……步入社会后一定要逐步改掉这些坏毛病。

囿于篇幅，不再一一言谢，惟愿大家科研顺利、工作顺利、身体安康。