

一种方向插值预测变长编码的帧存有损压缩算法

罗瑜^{*①} 张珍珍^②

^①(陕西中医药大学基础医学院 西安 712046)

^②(新加坡国立大学科学信息系统学院 新加坡 119077)

摘要: 为了提高帧存储的压缩性能, 该文提出一种基于方向插值预测变长编码(DIPVLC)的帧存有损压缩算法。首先根据自适应纹理方向插值获取参考像素, 从而得到预测残差, 然后优化率失真模型对预测残差进行量化, 最后通过游程哥伦布算法对量化残差进行变长编码。实验结果显示, 与内容感知自适应量化(CAAQ)的帧存压缩算法相比, 该文算法不但PSNR下降更少, 而且压缩率提高了10.05%, 同时编码时间减少了10.62%。

关键词: 有损压缩; 压缩率; 帧存储; 编码时间

中图分类号: TN919.81

文献标识码: A

文章编号: 1009-5896(2019)10-2495-06

DOI: 10.11999/JEIT181195

A Lossy Frame Memory Compression Algorithm Using Directional Interpolation Prediction Variable Length Coding

LUO Yu^① ZHANG Zhenzhen^②

^①(Department of Basic Medicine, Shaanxi University of Chinese Medicine, Xi'an 712046, China)

^②(Institute of Systems, National University of Singapore, 119077, Singapore)

Abstract: A lossy frame memory compression algorithm using Direction Interpolation Prediction Variable Length Coding (DIPVLC) is proposed to improve frame memory compression performance. Firstly, the prediction residual is obtained by adaptive texture directional interpolation. Then, a new rate-distortion is optimized to quantize prediction residual. Finally, the run length Golomb method is used to entropy coding for quantized residual. Simulation results show that compared with parallel Content Aware Adaptive Quantization (CAAQ) oriented lossy frame memory recompression for HEVC, the proposed algorithm improves the compression rate by 10.05% and reduces the encoding time by 10.62% with less PSNR reduction.

Key words: Lossy compression; Compression ratio; Frame memory; Encoding time

1 引言

21世纪以来, 视频相关领域迅速发展, 4k和8k的应用越来越多。视频领域的发展要求视频编码标准有更高的编码效率, 因此HEVC, VP9, AVS2和AV1等视频编码标准^[1]被相继推出。这些先进的视频编码标准不但有极高的计算复杂度, 同时需要大量的存储空间用于存储参考帧信息。在真实场景的应用中, 一方面, 过多的帧存占用了大量的存储空间; 另一方面, 参考帧的读写也导致系统带宽急剧增加, 实现成本过高。为了减少参考帧的存储空间, 并降低系统带宽, 帧存压缩技术^[2-5]被提出, 并得到广泛关注。

帧存压缩分为无损压缩^[6-9]和有损压缩^[10-12]两种。相对于无损压缩, 有损压缩在进一步提高压缩率的同时, 由于加入了量化的步骤, 导致重建图像的质量变差, 以至视频编码的PSNR下降。因此在有损帧存压缩技术中, 如何以较少的视频编码PSNR下降, 换取更高的压缩率, 就成为研究的关键。2018年, Lian等人^[13]提出了内容感知自适应量化(Content Aware Adaptive Quantization, CAAQ)的帧存压缩算法, 在目前大量的有损帧存压缩算法中, 该算法的压缩性能最为优秀。CAAQ首先在7个纹理方向中, 为每个像素自适应地选择预测方向以得到预测残差(Prediction Residual, PR); 然后采用率失真模型得到的量化参数(Quantized Parameters, QP)对PR进行量化, 最后利用一元哥伦布编码对量化残差(Quantized Residual, QR)进行熵编码。

但是CAAQ存在以下几个缺点:

(1)预测方向过少且预测精度不足; (2)率失真模型中功率谱密度的拟合和量化步长(Quantized

收稿日期: 2019-01-03; 改回日期: 2019-05-20; 网络出版: 2019-05-29

*通信作者: 罗瑜 luoyu2010@163.com

基金项目: 国家高技术研究发展计划(863计划)(2015M16903)

Foundation Item: The National High Technology Research and Development Program of China (2015M16903)

Step, QS)的计算均有欠缺;(3)一元哥伦布编码适用性不够全面,最终使压缩率不能进一步提高。为了克服以上问题,本文提出一种基于方向插值预测变长编码(Direction Interpolation Prediction Variable Length Coding, DIPVLC)的帧存有损压缩算法。仿真实验结果显示,本文算法较CAAQ,平均PSNR和压缩率均有进一步的提高,同时编码时间也有较大幅度的减少。

2 CAAQ算法分析

CAAQ采用 16×16 大小的块为编码单元,首先根据自适应预测方法得到PR,然后利用率失真模型计算QP,最后通过一元哥伦布对QR进行熵编码。

2.1 自适应预测

如图1所示,对于当前预测像素 P_{ij} ,利用其周围的重建像素,根据以下步骤计算PR:

- (1) 根据式(1),在2个区域中分别计算水平梯度值 D_{H1} 和 D_{H2} ,与垂直梯度值 D_{V1} 和 D_{V2} ;
- (2) 将式(1)的结果代入式(2),得到预测方向 θ ;
- (3) 若预测方向为 $45^\circ, 90^\circ, 135^\circ, 180^\circ$ 这4个主方向中的1个,则选取对应方向的临近像素作为参考像素;否则选取该方向临近像素值的平均值作为参考像素。

$$\begin{bmatrix} D_{H2} \\ D_{H1} \\ D_{V2} \\ D_{V1} \end{bmatrix} = \begin{bmatrix} P_{i-2,j} & P_{i-1,j} & P_{i-2,j-1} & P_{i-1,j-1} \\ P_{i-1,j-1} & P_{i,j-1} & P_{i-1,j-2} & P_{i,j-2} \\ P_{i-1,j-1} & P_{i-1,j} & P_{i-2,j-1} & P_{i-2,j} \\ P_{i,j-2} & P_{i,j-1} & P_{i-1,j-2} & P_{i-1,j-1} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \quad (1)$$

$$\left. \begin{aligned} D &= \text{MAX}(|D_{H1}| + |D_{V1}|, |D_{H2}| + |D_{V2}|) \\ D_V &= \begin{cases} D_{V1}, & (D == (|D_{H1}| + |D_{V1}|)) \\ D_{V2}, & (D == (|D_{H2}| + |D_{V2}|)) \end{cases} \\ D_H &= \begin{cases} D_{H1}, & (D == (|D_{H1}| + |D_{V1}|)) \\ D_{H2}, & (D == (|D_{H2}| + |D_{V2}|)) \end{cases} \\ \theta &= \begin{cases} 45, & 0.5 < \eta(D_V/D_H) \leq 2 \\ 67.5, & 2 < \eta(D_V/D_H) \leq 4 \\ 90, & |\eta(D_V/D_H)| > 4 \\ 112.5, & -4 < \eta(D_V/D_H) \leq -2 \\ 135, & -2 < \eta(D_V/D_H) \leq -1 \\ 157.5, & -1 < \eta(D_V/D_H) \leq -0.25 \\ 180, & -0.25 < \eta(D_V/D_H) \leq 0.5 \end{cases} \end{aligned} \right\} \cdot (2)$$

2.2 率失真模型

在视频编码中,QP与重建图像质量成反比,

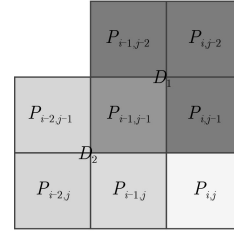


图1 自适应预测坐标示意图

与压缩率成正比,即QP越大,重建图像质量就越差,压缩率就越高。为了平衡图像质量和压缩率的折中,离散平稳高斯率失真函数^[14,15]被利用以计算QP,如式(3)所示, Ψ 和 Δ 分别为视频编码离散变换块和压缩量化噪声的功率谱密度。

$$\left. \begin{aligned} D &= \text{MIN}(QS^2/12, \psi + \Delta) \\ R &= \text{MAX}(0, 0.5 \times \log_2(\psi + \Delta \times 12/QS^2)) \end{aligned} \right\} \quad (3)$$

Ψ 和 Δ 根据式(4)计算得到,式中 Mvx 和 Mvy 分别为 2×2 子单元的水平 and 垂直的运行矢量。最终得到QP的计算公式如式(5)所示,并限制QP最大为3。在得到QP后,将PR右移QP个bit以得到当前像素的QR。

$$\left. \begin{aligned} \Delta &= 2^{2QP} / \alpha \\ \psi &= \frac{\sum_{i=0}^7 \sum_{j=0}^7 (dx_{i,j}^2 MMvx_{i,j}^2 \theta_x + dy_{i,j}^2 MMvy_{i,j}^2 \theta_y)}{64} \\ \theta_x &= |Mvx_{i,j}| > 8 ? 2 : 1 \\ \theta_y &= |Mvy_{i,j}| > 8 ? 2 : 1 \\ dx_{i,j} &= P_{2i,2j+1} + P_{2i+1,2j+1} - P_{2i,2j} + P_{2i+1,2j} \\ dy_{i,j} &= P_{2i+1,2j} + P_{2i+1,2j+1} - P_{2i,2j} + P_{2i,2j+1} \\ MMvx_{i,j} &= \text{mod}(Mvx_{i,j}, 4) / 4 \\ MMvy_{i,j} &= \text{mod}(Mvy_{i,j}, 4) / 4 \end{aligned} \right\} \quad (4)$$

$$QP = \begin{cases} 0, & \psi < (QS^2/12) \\ 0.5 \times \log_2 \left(\frac{2 \times \psi \times \log_2(\frac{\psi}{QS^2/12})}{10000} \right), & \psi \geq (QS^2/12) \end{cases} \quad (5)$$

2.3 一元哥伦布编码

哥伦布编码将PR通过 k 值拆分为商和余数2个分量,商通过查表编码,余数则直接传输。由于传统哥伦布商的码表不适用于较大的QR,因此CAAQ在传统哥伦布商的码表的基础上,加入一元编码的概念。

考虑到QR的纹理相关性,当前像素采用的 k 值与最优方向的临近像素所采用的 k 值保持一致,且

限幅为3。考虑到 k 值的变化会影响商的分布，因此针对不同的 k 值分别建立码表，如表1所示。

表1 哥伦布商码表

QR	$k=0$	$k=1$	$k=2$	$k=3$
0	0	00	000	0000
± 1	10	01	001	0001
± 2	110	100	010	0010
± 3	1110	101	011	0100
± 4	111100	1100	1000	0101
± 5	111101	1101	1001	0110
\vdots	\vdots	\vdots	\vdots	\vdots

2.4 算法分析

根据上述的算法原理介绍，可以发现CAAQ在每个算法步骤均有缺陷，具体如下：

(1) 对于自适应预测，首先没有充分利用周围像素的相关性(如右上相邻像素)，可能会导致预测方向不准；其次仅有7个预测方向，导致PR不能进一步变小；

(2) 对于率失真模型，功率谱密度的拟合没有考虑纹理方向，且QS没有考虑传递效应，导致QP计算不能达到最优；

(3) 对于一元哥伦布码表，一方面商没有限幅，可能导致商和余数的码长会远超过原始数据的比特深度，造成过膨胀现象。另一方面对于较小的QR，压缩率不能进一步提高。

3 DIPVLC算法

针对以上算法的缺点，为了进一步提高压缩性能，本文提出了DIPVLC算法，其在压缩单元大小与CAAQ保持一致的基础上，首先对当前像素进行方向插值预测得到PR，然后对率失真公式进行优化得到QP，最后对QR进行游程哥伦布编码。

3.1 方向插值预测

如图2所示，为了获得更小的PR，利用当前像素 P_{ij} 周围的10个重建像素计算纹理梯度值，并用紧相邻的4个像素计算最终参考像素，具体步骤为：

(1) 采用当前像素周围的10个重建像素，通过式(6)获取4个纹理梯度值；

(2) 选取其中梯度最小值对应的方向为纹理主方向；且在与主方向临近的2个方向中，选取最小梯度值对应的方向为纹理辅方向；举例：若纹理主方向为 0° ，则临近方向为 135° 和 45° ，选取 D_{45} 和 D_{135} 中最小值对应的方向为最终的纹理辅方向；

(3) 根据式(7)确定参考像素值 R ，其中 P_{mian} 和 P_{sub} 为与当前紧相邻的4个像素中，纹理主方向和

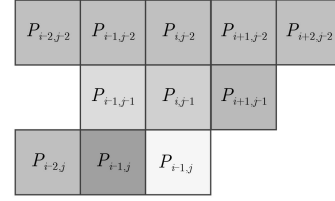


图2 方向插值预测图

辅方向对应的像素值， D_{min} 和 D_{sub} 为主方向和辅方向对应的梯度值。

$$\left. \begin{aligned} D_{90} &= |P_{i+1,j-2} - P_{i+1,j-1}| + |P_{i,j-2} - P_{i,j-1}| \\ &\quad + |P_{i-1,j} - P_{i-1,j-1}| \\ D_{00} &= |P_{i-1,j-1} - P_{i,j-1}| + |P_{i,j-1} - P_{i+1,j-1}| \\ &\quad + |P_{i-2,j} - P_{i-1,j}| \\ D_{45} &= |P_{i-1,j} - P_{i,j-1}| + |P_{i-1,j-1} - P_{i,j-2}| \\ &\quad + |P_{i+1,j-1} - P_{i+2,j-2}| \\ D_{135} &= |P_{i-1,j} - P_{i-2,j-1}| + |P_{i-1,j-1} - P_{i-2,j-2}| \\ &\quad + |P_{i-1,j-2} - P_{i,j-1}| \end{aligned} \right\} (6)$$

$$R = (P_{\text{main}} \times D_{\text{sub}} + P_{\text{sub}} \times D_{\text{min}}) / (D_{\text{sub}} + D_{\text{min}}) \quad (7)$$

3.2 率失真模型优化

本文算法从功率谱密度和QS入手，对率失真模型进行优化。

首先，从式(4)可以看出，梯度值和运动矢量也仅存在水平和垂直方向的拟合，不能满足自然图像的纹理特征，因此修改 Ψ 的计算公式如式(8)所示

$$\left. \begin{aligned} \Psi &= \frac{\sum_{i=0}^7 \sum_{j=0}^7 (d_{i,j}^2 (\mathbf{Mvy}_{i,j}^2 + \mathbf{Mvx}_{i,j}^2) \theta)}{64} \\ \theta &= \left(\left| \sqrt{\mathbf{Mvy}_{i,j}^2 + \mathbf{Mvx}_{i,j}^2} \right| > 8 \right) ? 2 : 1 \\ d_{00_{i,j}} &= P_{2i,2j+1} + P_{2i+1,2j+1} - P_{2i,2j} + P_{2i+1,2j} \\ d_{90_{i,j}} &= P_{2i+1,2j} + P_{2i+1,2j+1} - P_{2i,2j} + P_{2i,2j+1} \\ d_{45_{i,j}} &= P_{2i+1,2j} - P_{2i,2j+1} \\ d_{135_{i,j}} &= P_{2i,2j} - P_{2i+1,2j+1} \\ d_{i,j} &= \text{MIN}(d_{00_{i,j}}, d_{90_{i,j}}, d_{45_{i,j}}, d_{135_{i,j}}) \end{aligned} \right\} (8)$$

其次，由于视频编码具有传递效应，即当前压缩单元不仅决定当前帧的重建图像质量，而且作为参考帧，还会影响后续多帧的编码码率，因此率失真模型中的QS应考虑后续多帧参考该压缩单元的QS。

因为后续QS尚未得到，考虑到运动方向的延续性和帧间同位置QS的相关性，QS根据式(9)进行拟合，其中 CU_i 为当前位置编码单元按mv方向运动 i 次指向的编码单元， N 为参考帧数量。

$$\text{QS} = \frac{1}{N} \sum_{i=0}^N \text{QS}(\text{CU}_i) \quad (9)$$

最终将更新后的 ψ 和 QS 带入式(5), 计算得到当前压缩单元的QP。

3.3 游程哥伦布编码

一元哥伦布编码, 对于纹理复杂和中等区域, 由于 k 值相关性较差, 因此按照纹理方向选取 k 值, 且采用表1进行编码, 会导致因 k 值选取错误而造成码长过长的现象; 甚至由于没有限制最大码长, 会导致过度膨胀现象。对于纹理简单区域或QP较大时, 大部分QR为0, 此时一元哥伦布编码不能进一步减少编码比特数。为了克服以上问题, 游程哥伦布编码被提出, 流程图如图3所示。

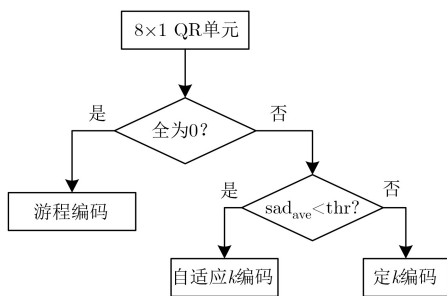


图3 游程哥伦布编码图

对于 16×16 编码单元中每个 8×1 大小的子单元, 步骤为:

(1) 若当前编码单元全为0, 则进行游程模式, 否则进入哥伦布模式, 同时将该模式信号写入码流, 码长为1 bit;

(2) 根据式(10)分析其编码单元 sad_{ave} , 若 sad_{ave} 小于门限 thr , 说明此时QR差异度较小, 采用自适应 k 进行编码; 否则采用定值 k 进行编码, 同时将该 k 模式信号写入码流, 码长为2 bit。

$$\left. \begin{aligned} k &= \text{MAX} \left(1, \text{MIN} \left(3, \log_2 \left(\sum_{i=0}^7 \text{QR}_i / 8 \right) \right) \right) \\ sad_{ave} &= \sum_{i=0}^6 |\text{QR}_i - \text{QR}_{i+1}| / 7 \end{aligned} \right\} (10)$$

为了限制过度膨胀, 优化熵码表如表2所示, 当商对照 k 值码表到达逃逸商值时, 余数码长从 k 个bit变为图像位宽个bit。表2中*表示当前 k 定义的逃逸商。

3.4 分析对比

在压缩率方面, 本文算法在预测步骤虽然只计算了4个预测方向, 但是通过权重插值公式, 拟合出可认为是任意方向的预测角度, 并且周围参考像素更多, 最终提高了预测精度; 量化步骤优化了率失真模型, 得到了更优的QP; 编码步骤区分不同情况采用不同的编码模式, 同时限制了哥伦布的过

表2 哥伦布商码表

QR	$k=0$	$k=1$	$k=2$	$k=3$
0	0	00	000	0000
± 1	10	01	001	0001
± 2	110	100	010	0010
± 3	1110	101	011	0011
± 4	1111*	1100	1000	0100
± 5		1101*	1001	0110
± 6			1010	
± 7			1011*	:
:				:
± 15				1111*

度膨胀情况。通过上述3个步骤的优化, 本算法较CAAQ算法, 最终提高了压缩率。

在运算量方面, 在预测步骤中, CAAQ有23次的加减比较运算和2次乘除法运算, 本文算法有22次的加减法运算和3次乘除法运算; 在量化步骤中, 2个算法基本持平; 在编码步骤中, 本文算法虽然在哥伦布编码中加入了模式判别, 但是游程模式不用编码, 逃逸商限制了查表次数, 且定值 k 忽略了查 k 操作。综上比较, 最终本算法复杂度有所下降。

4 实验结果与分析

为了比较本文算法与CAAQ的算法性能, 将算法嵌入HEVC后, 本实验分别从压缩率(Compression Ratio, CR)、峰值信噪比(Peak Signal to Noise Ratio, PSNR)和相对编码时间(Relative Encoding Time, RET)3个维度进行测试。测试平台为HM16.8^[16], 运行环境为Inter I5-6400 CPU@2.7 GHz-8 G内存-Win7 64位系统。测试序列分为纹理复杂度简单、中等和复杂3种共9个序列。CR和RET计算公式如式(11)

$$\left. \begin{aligned} \text{CR} &= (1 - \text{Size}_{bs} / \text{Size}_{ori}) \times 100\% \\ \text{RET} &= (\text{Time}_1 / \text{Time}_2) \times 100\% \end{aligned} \right\} (11)$$

其中, Size_{ori} 和 Size_{bs} 分别为压缩前后数据量大小; Time_1 和 Time_2 分别为两种算法的编码时间。CR越大, 说明压缩率越好; RET越小, 说明算法1的编码时间相对越少。

首先, 为了确定本文算法最优的阈值 thr , 本实验选取3个不同纹理复杂度的典型序列, 进行压缩率的比较, 实验结果如图4所示, 可以看出平均 thr 为4时, 压缩率最高。因此在后续实验中, 本文设定 thr 等于4。

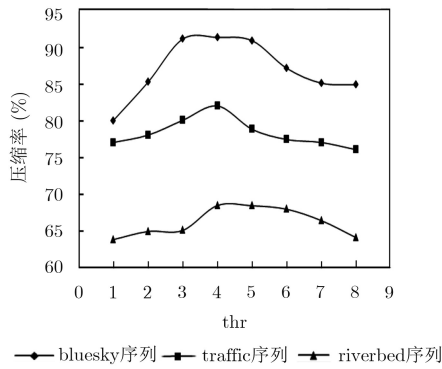


图 4 thr测试对比

其次，进行本文算法各优化模块的性能分析。在CAAQ的基础上，本文算法分别对方向插值预测模块、率失真模型和编码模块进行了优化，其它模块和CAAQ保持一致。选取3个不同纹理复杂度的典型序列，进行CR，PSNR和RET 3个维度的比较，每个优化点带来的性能提升如表3所示。

以CR项为例，表3中CR项分为CAAQ、预测、率失真和编码4个子项，其中CAAQ子项表示针对不同序列CAAQ的CR值；预测子项表示其它模块保持和CAAQ相同，仅优化预测模块后本文算法的CR值；率失真子项表示其它模块保持和CAAQ相同，仅优化率失真模型后本文算法的CR值；编码子项表示其它模块保持和CAAQ相同，仅优化编

码方式后本文算法的CR值。PSNR和RET项类似，可以看出实验结果与3.4节的理论分析一致，本文算法的方向插值预测和率失真优化模型，计算量与CAAQ的对应模块基本相同，但是压缩率得到提升；由于率失真优化模型的运动方向的延续性，与图像纹理相关度不大，所以平均压缩率的提升较大；本文算法的游程哥伦布编码，不但对压缩率的提升较大，而且其中的游程模式和逃逸商是引起计算量减少的关键。

最后，进行本文算法与CAAQ算法最终的压缩性能对比，实验结果如表4所示。在CR和PSNR方面，本算法与CAAQ相比，首先由于采用更加精确的插值预测，所以得到的PR更小；其次由于优化了率失真模式，从而得到了更准确的QP；最后针对QR大多为0区域和其它区域，分别采用了游程编码和不同l模式且支持逃逸商的哥伦布编码，使压缩率得到进一步提升。从表4可看出，本文算法对于简单纹理序列Tennis，压缩率提高最大，为15.02%。从表4还可以发现随着纹理复杂度的逐渐增加，游程编码出现概率逐渐变小，导致压缩率的提升也逐渐降低。最终本文算法较CAAQ，不但平均压缩率提高10.05%，而且PSNR还提高0.04 dB。

在算法复杂度方面，对于预测和量化步骤，本算法与CAAQ的计算复杂度基本一致；对于编码步

表 3 本文算法模块性能提升对比

序列	CR(%)				ΔPSNR(dB)				RET 模块/CAAQ(%)		
	CAAQ	预测	率失真	编码	CAAQ	预测	率失真	编码	预测	率失真	编码
bluesky	80.26	83.69	85.64	86.25	-0.05	-0.05	-0.04	-0.04	99.56	98.15	85.34
traffic	70.95	73.11	77.64	76.61	-0.07	-0.06	-0.03	-0.05	99.12	100.54	90.12
riverbed	60.21	61.21	67.21	65.21	-0.09	-0.09	-0.04	-0.05	98.89	101.51	95.14
平均	70.47	72.67	76.83	76.02	-0.07	-0.07	-0.04	-0.05	99.19	100.07	90.20

表 4 本文算法与CAAQ算法压缩的性能对比

序列	CR(%)		ΔPSNR(dB)		RET
	CAAQ	本文算法	CAAQ	本文算法	本文/CAAQ(%)
Tennis	78.21	93.23	-0.02	-70.01	86.00
bluesky	80.26	91.45	-0.05	-0.03	85.21
Johnny	81.39	93.56	-0.05	-0.02	84.25
crowdrun	71.21	79.56	-0.06	-0.01	89.15
traffic	70.95	82.10	-0.07	-0.02	89.25
stockholm	70.12	79.12	-0.08	-0.03	88.56
racehorses	64.36	73.14	-0.06	-0.01	92.31
riverbed	60.21	68.52	-0.09	-0.02	96.14
mobcal	59.76	67.14	-0.08	-0.03	93.54
平均	70.72	80.87	-0.06	-0.02	89.38

骤, 虽然本算法在分析模块增加了额外的计算量, 但是游程编码和4种不同 l 模式的哥伦布编码节约了大量的编码时间; 最终本算法较CAAQ, 减少了编码时间。从表4可以看出, 平均RET为89.38%, 说明本算法较CAAQ的编码时间减少了10.62%。从表4还可以发现, 图像越简单, 则游程编码的概率越大, RET就越小。

5 结论

为了进一步提高帧存有损压缩的压缩性能, 针对CAAQ算法的缺点, 本文提出了DIPVLC算法。DIPVLC首先充分利用周围像素的纹理方向性和权重插值, 为每个像素自适应的计算最优的参考像素, 并计算获得PR; 然后从功率谱密度和QS维度优化了率失真模型, 得到QP; 最后针对不同类型的QR分布, 采用游程或不同 l 模式的哥伦布进行编码。最终本算法不但提高压缩率和PSNR, 而且还降低了算法复杂度。实现结果显示, 与CAAQ算法相比, 本算法CR平均提升了10.05%, PSNR平均提升了0.04 dB, RET仅为CAAQ的89.38%。

参考文献

- [1] ITU-T Study Group 16. Recommendation ITU-T h.265 High efficiency video coding[S]. Geneva: ITU, 2014.
- [2] FAN Yibo, SHANG Qing, and ZENG Xiaoyang. In-block prediction-based mixed lossy and lossless reference frame recompression for next-generation video encoding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2015, 25(1): 112–124. doi: [10.1109/TCSVT.2014.2329353](https://doi.org/10.1109/TCSVT.2014.2329353).
- [3] LI Weigang. Optimize genomics data compression with hardware accelerator[C]. 2017 Data Compression Conference (DCC), Snowbird, USA, 2017: 446.
- [4] GUPTE A D, AMRUTUR B, MEHENDEALE M M, *et al.* Memory bandwidth and power reduction using lossy reference frame compression in video encoding[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2011, 21(2): 225–230. doi: [10.1109/TCSVT.2011.2105599](https://doi.org/10.1109/TCSVT.2011.2105599).
- [5] QIAN Dong and LI Bing. A lossless compression method for RTK in hardware compressors[C]. 2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC), Taiwan, China, 2017: 1–2.
- [6] MA Yanzhuo and KANG Lijuan. Adaptive granularity selection in reference picture memory compression[C]. The 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering, Kuala Lumpur, Malaysia, 2015: 1158–1161.
- [7] LEE Y, RHEE C E, LEE H J. A new frame recompression algorithm integrated with h.264 video compression[C]. 2007 IEEE International Symposium on Circuits and Systems, New Orleans, USA, 2007: 1621–1624.
- [8] SAMPALIO F, ZATT B, SHAFIQUE M, *et al.* Content-adaptive reference frame compression based on intra-frame prediction for multiview video coding[C]. 2013 IEEE International Conference on Image Processing, Melbourne, Australia, 2013: 1831–1835.
- [9] BAGA Y, GHAFARI F, DECLERCQ D, *et al.* Reduction of frames storage size in AFDX reception end-system using a lossless compression algorithm[C]. The 36th IEEE/AIAA Digital Avionics Systems Conference (DASC), Saint Petersburg, USA, 2017: 1–8.
- [10] WILLÈME A, MACQ B, DESCAMPE A, *et al.* JPEG XS-based frame buffer compression inside HEVC for power-aware video compression[C]. 2018 IEEE International Conference on Image Processing (ICIP), Athens, Greece, 2018: 3598–3602.
- [11] ZHOU Xin, LIAN Xiacong, ZHOU Wei, *et al.* A low power lossy frame memory recompression algorithm[C]. 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Jeju, South Korea, 2016: 1–4.
- [12] CHEN Qiubo, SUN Hongbin, and ZHENG Nanning. Worst case driven display frame compression for energy-efficient ultra-HD display processing[J]. *IEEE Transactions on Multimedia*, 2018, 20(5): 1113–1125. doi: [10.1109/tmm.2017.2762004](https://doi.org/10.1109/tmm.2017.2762004).
- [13] LIAN Xiacong, LIU Zhenyu, ZHOU Wei, *et al.* Parallel content-aware adaptive quantization-oriented lossy frame memory recompression for HEVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 28(4): 958–971. doi: [10.1109/TCSVT.2016.2638857](https://doi.org/10.1109/TCSVT.2016.2638857).
- [14] WIEGAND T and GIROD B. Lagrange multiplier selection in hybrid video coder control[C]. 2001 International Conference on Image Processing, Thessaloniki, Greece, 2001: 542–545.
- [15] ZHANG Fan and BULL D R. Rate-distortion optimization using adaptive lagrange multipliers[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 20(3): 150–153. doi: [10.1109/TCSVT.2018.2873837](https://doi.org/10.1109/TCSVT.2018.2873837).
- [16] International Telecommunication Union. HM16.8[EB/OL]. https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware.2018.12.1.

罗 瑜: 女, 1984年生, 副教授, 研究方向为图形图像处理。

张珍珍: 女, 1984年生, 博士生, 研究方向图形图像处理。