

文章编号 1004-924X(2019)04-0990-10

## 联合多特征的未来视频快速编码

崔鑫, 彭宗举\*, 陈芬

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

**摘要:** 与前一代高效视频编码标准相比, 未来视频编码标准增加了新的二叉树加四叉树编码结构。针对该结构虽然显著提升了超高清视频编码效率, 但是大幅增加了编码复杂度的问题, 提出了联合多特征快速编码算法。该算法联合多个编码特征的后验概率信息, 估计当前编码单元的划分方式, 通过提前终止若干编码单元的划分来节省编码时间。此外, 针对单个特征贝叶斯分类不准确的问题, 算法在简化联合概率模型的同时提升了分类算法的准确度。实验结果表明: 在随机访问配置、低延迟 P 配置和低延迟 B 配置下, 算法可以平均减少 35.7%, 25.6% 和 26.7% 的编码复杂度, 而 BDBR 则分别增长了 4.3%, 3.1% 和 2.89%。算法在保证视频主观质量的前提下, 节省了编码时间。

**关键词:** 未来视频编码; 联合多特征快速编码算法; 二叉树加四叉树结构; 快速编码

**中图分类号:** TP394.1; TH691.9 **文献标识码:** A **doi:** 10.3788/OPE.20192704.0990

## Joint multi-feature fast coding for future video coding

CUI Xin, PENG Zong-ju\*, CHEN Fen

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

\* Corresponding author, E-mail: pengzongju@nbu.edu.cn

**Abstract:** Future video coding considerably improves compression efficiency over that of the previous high-efficiency video coding standard. The quadtree plus binary tree structure of future video coding is more consistent with the texture of high-definition video but at the cost of tremendous computational complexity. In this study, a joint multi-feature classification algorithm was proposed to reduce the complexity of the quadtree plus binary tree block structure in future video coding. The partition mode of the current coding unit was estimated using the proposed algorithm, and the partition of the coding unit was then terminated. Consequently, coding time was conserved by the proposed algorithm. In general, classification with a single feature was inaccurate. Thus, a joint multi-feature classification model was used in the proposed algorithm. The proposed algorithm not only improved the accuracy of the classification, but also simplified the joint probability model. Experimental results show that our algorithm can reduce the coding complexity on average by 35.7%, 25.6%, and 26.7% with only a 4.3%, 3.1%, and 2.89% Bjøntegaard delta bit rate increment with the random access main, low delay P, and low delay B configurations, respectively. The encoding time is conserved by the proposed algorithm based on subjective video quality.

**收稿日期:** 2018-07-03; **修订日期:** 2018-09-05.

**基金项目:** 国家自然科学基金资助项目 (No. 61771269, No. 61620106012, No. 61671258); 浙江省自然科学基金资助项目 (No. LY17F010005)

**Key words:** future video coding; joint multi-feature classification algorithm; quadtree plus binary tree block structure; fast coding technique

## 1 引言

随着人们对视频图像质量要求的不断提高,为了更好地适用于不同的现实场景,相应的编码技术和标准也与时俱进。为此国际电信联盟电信标准化部门(ITU-T)联合国际标准化组织(ISO)/国际电工委员会(IEC)的动态图像专家组在高效视频编码标准(High Effective Coding, HEVC)的基础上,开发了未来视频编码技术 H.266/FVC(Future Video Coding, FVC)。该标准与 HEVC 最大的不同是编码结构由四叉树结构(Quadtree Tree, QT)变为二叉树加四叉树结构(Quadtree Plus Binary Tree, QTBT)。QTBT 结构能更好地适应 4K 甚至 8K 超高清视频的纹理结构,显著地提高了编码效率。然而,QTBT 结构也使得编码复杂度大幅增加<sup>[1]</sup>。因此,采用行之有效的快速编码算法,可以在保证相同编码效率的同时,减少编码时间。

针对 QTBT 复杂度的研究虽然尚处于起步阶段,目前也取得一些进展。根据实现方式主要分为三类:第一类是基于编码特征信息来限制 QTBT 划分冗余,如 Wang 等人<sup>[2]</sup>根据局部编码特征信息,提出了快速编码算法,提前终止不必要的编码单元(Coding Unit, CU)划分,但是该算法只针对 QTBT 中二叉树的划分并未考虑四叉树划分,而且该算法需要同时作用于编码和解码阶段,因此增加了额外的编码时间;Yamamoto 等人<sup>[3]</sup>根据编码时间层重要性的不同来限制 CU 最大划分深度,进而降低编码复杂度,但是该方法没有建立系统的算法模型,因此算法的准确性和实用性不足;Lin 等人<sup>[4]</sup>通过设计特定阈值来跳过二叉树中不必要的划分;Park 等人<sup>[5]</sup>提出了基于参考帧的快速算法,算法跳过若干参考帧搜索过程中的运动估计,并减少当前 CU 块的参考帧搜索范围,进而节省编码时间。Lin 和 Park 等人的算法只针对 QTBT 中的二叉树结构,并未考虑四叉树结构。第二类是基于统计分类模型快速算法,如 Wang 等人<sup>[6]</sup>提出了联合分类决策树结构算法(Joint-Classifer Decision Tree Structure,

JCDT),该算法联合编码特征信息,并利用信息增益属性决策算法,提前判断当前 CU 划分状态而终止不必要的划分,进而节省编码时间,但是该算法只考虑了 HEVC 中加入 QTBT 结构后的情况,而并未考虑 FVC 中加入该结构的情况。第三类是基于编码率失真代价函数的快速算法,如 Wang 等人<sup>[7]</sup>针对 FVC 提出了基于置信区间分类的 QTBT 快速算法,该算法根据编码特征的统计信息重新建模了不同划分模式下的率失真代价函数,因此在 QTBT 划分前预测出各种划分模式的概率,然后基于置信区间进行划分模式分类选择,并取得较好的结果,但是该算法需要离线进行参数统计来建立率失真代价函数,因此制约了模型的实际应用。

贝叶斯分类模型被广泛地应用于 HEVC 中的快速 QT 结构划分<sup>[8-12]</sup>。然而,它们都是基于单个特征的贝叶斯分类模型,对于 QTBT 结构的预测准确度不高,并不能直接用来加速 FVC 编码。因此,本文利用多个编码特征信息,提出了联合多特征快速编码算法(Joint multi-feature fast coding, JMFC)。该算法能准确预测 QTBT 编码结构,降低编码复杂度。

## 2 FVC 编码树结构分析

QTBT 编码结构支持更灵活的 CU 块划分类型,其划分形状可以是正方形也可以是矩形。在编码过程中,根据叶节点和根节点的率失真代价值进行“由上向下划分”和“由下向上裁剪”<sup>[17]</sup>。与 HEVC 中的 QT 编码结构相似,QTBT 也存在着划分冗余。

如图 1 所示,当量化参数(Quantized Parameter, QP)为 32 时,统计的视频序列 BasketballPass、BQMall、RaceHorses 和 Johnny 中四叉树、水平二叉树和垂直二叉树的不划分概率,可以得出以下结论:

QTBT 结构中可能存在一定编码冗余,如 Johnny 序列中水平二叉树和垂直二叉树不划分概率分别为 0.90 和 0.86,因此可以通过提前终止 CU 划分来减少编码迭代次数,进而节省编码时间。

QTBT 中四叉树与二叉树划分概率不同。如图 1 中所示,序列的四叉树不划分的概率要小于二叉树不划分概率,因此本文分类算法分情况考虑二叉树结构和四叉树结构。

不同序列的划分概率不同。划分概率与视频本身的运动特性和纹理结构密切相关,如 Johnny 序列的不划分概率要大于 BasketballPass 序列,是因为 Johnny 序列运动相对静止,且纹理平坦区域较多。

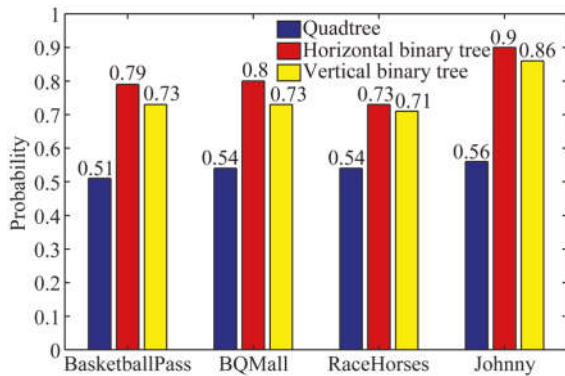


图 1 QTBT 不划分概率

Fig. 1 Non-Split CU probability in QTBT

通过以上分析可以看出,在编码过程中 QTBT 存在大量的编码冗余,因此可以结合视频图像的特征信息,去除编码冗余以达到节省编码时间的目的。

### 3 联合多特征 FVC 快速编码算法

#### 3.1 算法整体流程

图 2 为本文提出的基于在线学习的 FVC 快速编码流程,编码过程主要分为:训练阶段和测试阶段。在训练阶段,进行正常地 CU 编码,并学习和更新参数。在测试阶段,首先利用训练阶段得到的参数建立联合多特征分类模型,再利用分类模型预测 QTBT 划分结构,提前终止 CU 划分。

QTBT 结构中的二叉树划分会影响四叉树划分,进而造成分类错误累加。JMFC 算法在分类过程中首先判断二叉树划分情况,如果判断当前水平和垂直二叉树有且只有一个划分,则为水平或者垂直二叉树划分,四叉树为不划分。若判断水平和垂直二叉树都为不划分时,再判断四叉树是否划分。

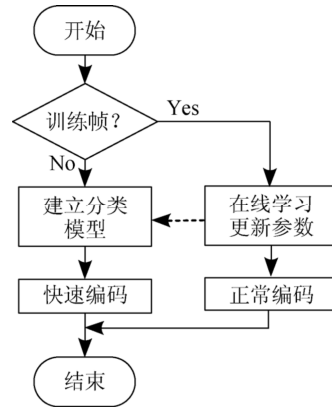


图 2 联合多特征快速编码算法整体流程图

Fig. 2 Overall flowchart of JMFC algorithm

#### 3.2 特征提取分析

本文 JMFC 算法,选择了 4 个编码特征,分别是  $B^{\text{Inter}}$ ,  $D^{\text{Inter}}$ ,  $J^{\text{Merge}}$  和  $J^{\text{Non-split}}$ 。其中  $D^{\text{Inter}}$  和  $B^{\text{Inter}}$  是帧间模式的失真和码率,二者与编码图像质量相关。 $J^{\text{Merge}}$  是 Merge 模式的率失真代价,反映了当前 CU 块的运动信息。 $J^{\text{Non-split}}$  是 CU 块在进行 QTBT 之前的不划分率失真代价值,反映了当前 CU 的划分或不划分趋势。

如图 3(a)~图 3(l)所示为序列 BQMall 编码过程中,一帧视频图像中 4 个特征的 CU 划分/不划分概率分布曲线,其中图 3(a)~图 3(d)是四叉树划分/不划分概率分布,图 3(e)~图 3(h)是水平二叉树划分/不划分概率分布,图 3(i)~图 3(l)是垂直二叉树划分/不划分概率分布,可以看出它们近似符合高斯分布,且不划分概率分布曲线更“凸出”,而划分概率分布曲线更“平缓”。还可以看出,划分与不划分概率分布曲线存在“交叠区域”,该区域会对分类模型造成错误判断,例如图 3(b)、图 3(i)、图 3(j)、图 3(k)和图 3(l)中存在比较明显的“交叠区域”,原因是该视频图像中的 CU 垂直二叉树划分与不划分概率接近。

通过以上分析,特征变量的划分和不划分概率分布曲线,能够体现当前 CU 划分/不划分的趋势,但是对于单特征分类模型,当划分和不划分的概率分布曲线“交叠区域”较大时,会导致分类准确性降低,因此针对以上问题,本文提出了联合多特征分类模型。

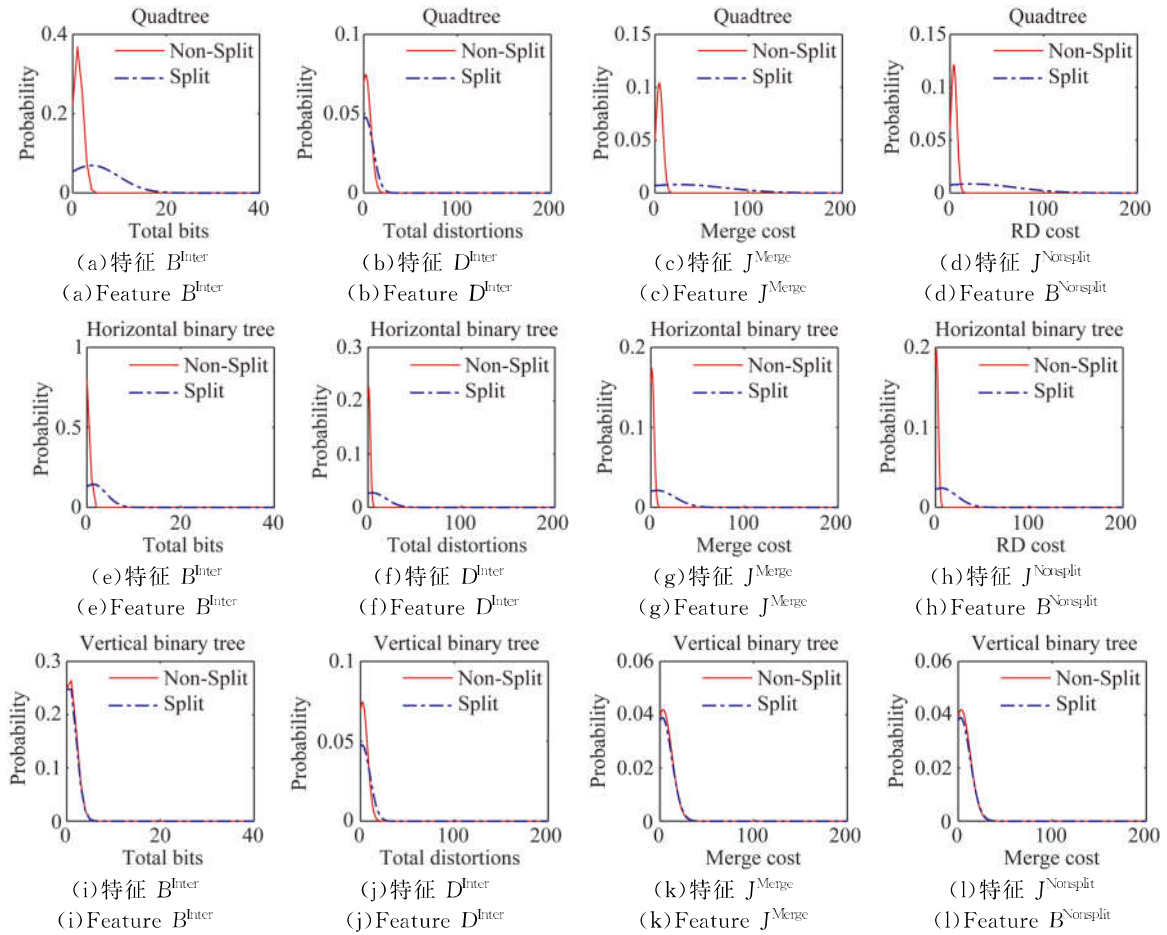


图 3 序列 BQMall 的 QTBT 划分/不划分概率分布 (QP=32)

Fig. 3 Non-Split/Split probability distribution of QTBT in BQMall sequence (QP=32)

### 3.3 联合多特征分类模型

联合多特征分类模型利用了贝叶斯和因子图“和-积”算法的基本准则, 以及最大似然置信消息传递算法的基本思想<sup>[13-14]</sup>。如图 4 所示, 模型主要包括三部分: 贝叶斯模块、特征节点和判决节点, 其主要流程可以概述为:

(1) 贝叶斯预估计。根据贝叶斯准则, 获得当前 CU 在编码特征  $\mathbf{J}$  条件下的不划分概率  $P(W_{NS} | \mathbf{J})$ , 即:

$$P(W_{NS} | \mathbf{J}) = \frac{P(\mathbf{J} | W_{NS}) \cdot P(W_{NS})}{P(\mathbf{J} | W_{NS}) \cdot P(W_{NS}) + P(\mathbf{J} | W_S) \cdot P(W_S)}, \quad (1)$$

其中:  $\mathbf{J} = (B^{Inter}, D^{Inter}, J^{Merge}, J^{Nonsplit})$  为特征矩阵,  $P(W_{NS})$  表示一帧图像中 CU 不划分的概率,  $P(\mathbf{J} | W_{NS})$  表示在不划分条件下的特征先验概率,  $P(W_S)$  表示 CU 划分的概率,  $P(\mathbf{J} | W_S)$  表示在划分条件下的特征先验概率。因此根据训练帧

得到的特征先验概率信息并结合贝叶斯准则, 得出后验概率  $P(W_{NS} | \mathbf{J})$ , 同理按照式 (1) 可得  $P(W_S | \mathbf{J})$ , 二者作为当前 CU 不划分和划分的初始估计。

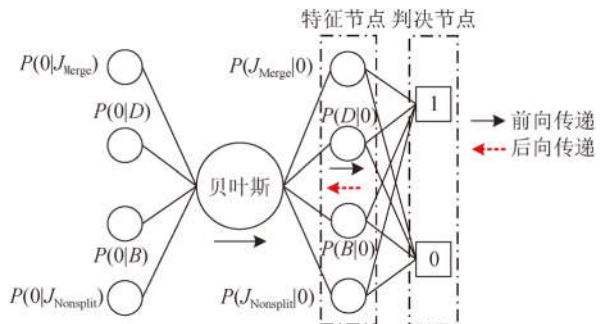


图 4 联合多特征分类模型

Fig. 4 Joint multi-feature classification model

(2) 初始化特征节点消息。利用式 (1) 中得到的特征后验概率  $P(0 | \mathbf{J})$  和  $P(1 | \mathbf{J})$  初始化接收矩

阵  $\mathbf{R}$  的消息, 可得:

$$\mathbf{R}_{i,j} = \begin{cases} P(W_{NS} | \mathbf{J}) = P(0 | \mathbf{J}), j=0 \\ P(W_S | \mathbf{J}) = P(1 | \mathbf{J}), j=1 \end{cases}, \quad (2)$$

其中:  $P(0 | \mathbf{J})$  和  $P(1 | \mathbf{J})$  分别表示在特征矩阵  $\mathbf{J}$  条件下“0”和“1”的概率, “0”表示 CU 块不划分, “1”表示 CU 块继续划分。然后利用接收矩阵初始特征节点处的消息, 即:

$$\mathbf{M}_{i,j} = \mathbf{R}_{i,j}, \quad (3)$$

其中:  $\mathbf{M}$  为  $j \times i$  的特征矩阵,  $\mathbf{R}$  为  $j \times i$  的接收矩阵。

(3) 更新判决节点。利用式(3)得到的特征矩阵  $\mathbf{M}$  前向更新判决节点  $\mathbf{E}$  的消息, 即:

$$\mathbf{E}_{i,j} = \log \frac{1 + \prod_{i' \in \mathbf{I}_j, i' \neq i} \tanh \left( \frac{M_{i',j}}{2} \right)}{1 - \prod_{i' \in \mathbf{I}_j, i' \neq i} \tanh \left( \frac{M_{i',j}}{2} \right)}, \quad (4)$$

其中:  $\mathbf{E}$  为  $j \times i$  矩阵,  $\mathbf{I}_j$  表示第  $j$  行所有变量的集合,  $i' \in \mathbf{I}_j, i' \neq i$  表示第  $j$  行中, 除第  $i$  个变量以外其他所有变量。

(4) 更新特征节点。利用式(4)中得到的判决矩阵  $\mathbf{E}$  反向更新特征节点  $\mathbf{M}$  的消息, 即:

$$\mathbf{M}_{i,j} = \sum_{j' \in \mathbf{C}_i, j' \neq j} \mathbf{E}_{i,j'} + \mathbf{R}_i, \quad (5)$$

其中:  $\mathbf{C}_i$  表示第  $i$  列所有变量的集合,  $j' \in \mathbf{C}_i, j' \neq j$  表示第  $i$  列中除第  $j$  个变量以外其他所有变量。

(5) 计算判决矩阵  $\mathbf{L}$ 。利用若干次前向和后向消息迭代后, 得到的判决节点  $\mathbf{E}$  的消息和接收矩阵  $\mathbf{R}$  的消息计算判决矩阵, 可得:

$$\mathbf{L}_i = \sum_{j \in \mathbf{C}_i} \mathbf{E}_{i,j} + \mathbf{R}_i. \quad (6)$$

(6) 贝叶斯最小风险函数。通过加入最小风险函数来提高算法的分类精度, 如式(7)和式(8)所示:

$$Q_{NN}^{\text{Merge}} = \frac{|J_{\text{Split}}^{\text{Merge}} - J_{\text{Nonsplit}}^{\text{Merge}}|}{J_{\text{Nonsplit}}^{\text{Merge}}}, \quad (7)$$

$$Q_{SS}^{\text{Merge}} = \frac{|J_{\text{Nonsplit}}^{\text{Merge}} - J_{\text{Split}}^{\text{Merge}}|}{J_{\text{Split}}^{\text{Merge}}}, \quad (8)$$

分别表示基于 Merge 模式的率失真代价值计算的贝叶斯最小风险函数, 其中  $Q_{NN}^{\text{Merge}}$  表示当前 CU 不划分的权值,  $Q_{SS}^{\text{Merge}}$  表示当前 CU 划分的权值。

(7) 分类判断。通过设定判决阈值, 来控制模型迭代次数, 即:

$$K_n = \prod_{i \in \mathbf{I}_j} L_i, n \in [0, 1], \quad (9)$$

其中: 当  $Q_{NN}^{\text{Merge}} K_0 \geq Q_{SS}^{\text{Merge}} \cdot K_1 \cap K_0 \geq TH$ , 则判定为“0”, 即当前 CU 不划分; 否则判定为“1”, 即当前 CU 划分, 其中  $TH$  为判决阈值, 阈值的选择在后面章节介绍。

## 4 实验结果分析

### 4.1 算法判决阈值选择分析

本文 JMFC 算法阈值的设置主要考虑了编码节省时间和 BDBR (Bjontegaard Delta Bit Rate) 之间的平衡, 其中 BDBR 衡量了算法的整体性能。编码节省时间  $\Delta T$  衡量了编码的复杂度, 表示为:

$$\Delta T = \frac{T_{\text{proposed}} - T_{\text{reference}}}{T_{\text{reference}}} \times 100\%, \quad (10)$$

其中:  $T_{\text{proposed}}$  表示利用 JMFC 算法编码花费的时间,  $T_{\text{reference}}$  表示利用原始 JEM-7.0 平台编码花费的时间。

如表 1 所示, 为 JMFC 算法阈值  $TH$  设定范围, 该范围是根据特征的先验概率计算得出。表中列出了特征先验概率从 91%~95% 之间对应的阈值  $TH$ 。

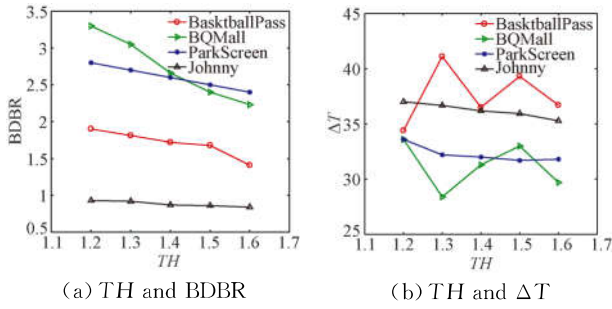
表 1 阈值范围设定

Tab. 1 Setting of threshold range

阈值 $TH$	1.2	1.3	1.4	1.5	1.6
$P(0   X) \geq 91\%$	$\geq 91\%$	$\geq 92\%$	$\geq 93\%$	$\geq 94\%$	$\geq 95\%$

如图 5(a) 所示, 分别为视频序列 BasketballPass、BQMall、ParkScreen 和 Jonny 的 BDBR 在不同阈值  $TH$  下的分布情况。从图 5(a) 可以看出随着阈值  $TH$  的增加, 4 个视频序列的 BDBR 逐渐减小。说明随着阈值  $TH$  的增加, 利用 JMFC 算法编码后的视频质量越高。通过表 1 还可以看出, 初始的先验概率越准确, 根据后验概率计算得到的阈值也越大。

如图 5(b) 所示, 为 JMFC 算法编码节省时间与阈值  $TH$  之间的关系, 从 5(b) 可以看出, 编码节省时间在不同阈值  $TH$  下, 分布相对均匀, 因此 JMFC 算法在权衡了 BDBR 和编码节省时间, 选择 1.6 作为算法的阈值。

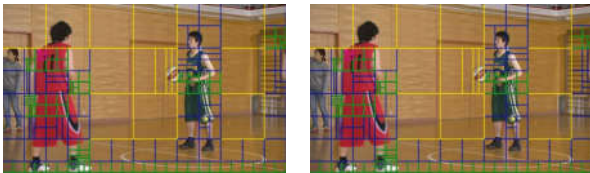


(a) TH and BDBR (b) TH and  $\Delta T$

图 5 BDBR 和  $\Delta T$  与阈值 TH 的关系

Fig. 5 Relationship between TH and BDBR,  $\Delta T$

如图 6(a)和图 6(b)分别表示采用原始 JEM-7.0 平台编码后和 JMFC 算法编码后的 CU 块划分图。由图 6(b)中可以看出使用 JMFC 算法编码后,与图 6(a)相比,编码过程中采用较大尺寸 CU,尤其是平坦区域,说明 JMFC 算法抑制了若干大尺寸 CU 继续划分为小尺寸 CU,而且并未造成视频主观质量下降。因此 JMFC 算法在保证一定分类准确性的前提下,减少了划分冗余。



(a)原始图像 (a)Original image (b)JMFC 算法 (b)JMFC algorithm

图 6 BasketballPass 第 3 帧 CU 块划分图

Fig. 6 Third frame image of BasketballPass sequence

#### 4.2 训练帧选择规则

训练帧的选择影响分类模型的准确性,因此本文分别根据随机访问(Random Access, RA)配置和低延迟(Low Delay, LD)配置的不同特点进行选择。训练帧的选择主要考虑了 QP 的影响,对于不同时间层的帧,QP 的补偿值不同,尤其是小 QP 情况下,对于视频质量影响较大。因此根据 RA 配置和 LD 配置下的不同编码顺序,在权衡了编码时间和 BDBR 之后,得出了如表 2 所示的训练帧选择规则,在 RA 配置下,非 I 帧每隔 8 帧训练一次,在 LD 配置下,前 4 帧全部作为训练帧,其余非 I 帧每隔 4 帧训练一次。

表 2 RA 与 LD 配置下训练帧及 QP 选择规则

Tab. 2 Selection rules of training frames and QP under RA and LD configuration

基准 QP	训练帧选择规则	
	RA	LD
22	QP ≤ 26	QP = 27
27	QP ≤ 32	QP = 34
32	QP ≤ 38	QP = 40
37	QP ≤ 44	QP = 45

#### 4.3 算法性能分析

实验测试了 QP 为 22、27、32 和 37 条件下, JMFC 算法在随机访问 RA、低延迟 LDP 和 LDB 配置下的性能。并对比了 Yamamoto 等人<sup>[3]</sup>、Lin 等人<sup>[4]</sup>和 Park 等人<sup>[5]</sup>的算法以及文献[6]提出的 JCDT 算法,由于 JCDT 算法是基于 HM-13.0-QTBT 实现的,因此本文将 JCDT 算法移植到 JEM-7.0 平台上进行测试,而其他对比算法都采用了 JEM 编码平台进行测试,各版本之间并无很大差异,都是基于 H. 266 编码标准的平台,采用了相同的 QTBT 编码结构,因此保证了对比的一致性。如表 3 所示是本文 JMFC 算法在 RA 配置下的算法性能比较,从表 3 可以看出,与原始编码平台相比, JMFC 算法的 BDBR 大约增加了 4.3%,编码时间大约节省 35.7%。与 JCDT 算法相比,本文提出的 JMFC 算法的 BDBR 仅增加了大约 1.9%,但是却节省了大约 17.3% 编码时间。与 Yamamoto 等人和 Lin 等人的算法相比, BDBR 分别增加了 3.8% 和 4.1%,但是编码时间分别减少了 17.1% 和 27.2%,可以看出本文 JMFC 算法在大部分序列上,编码复杂度都优于对比算法。

如表 4 所示为本文 JMFC 算法在低延迟 LDP 和 LDB 配置下的算法性能,与原始编码平台相比, BDBR 分别增加了 3.1% 和 2.89%,编码时间分别减少了 25.6% 和 26.7%。如表 5 所示是本文 JMFC 算法在低延迟 LDB 配置下与其他算法的性能对比,与 Park 等人和 Lin 等人算法相比, BDBR 分别增加了 2.3% 和 2.4%,编码时间分别减少了 14.9% 和 21.7%。因此在低延迟

LDB 下, JMFC 算法在大部分序列上, 编码复杂度都优于对比算法。由表 3、表 4 和表 5 可以看出, 序列 BasketballDrill, BasketballDrive, Kimono 和 Drums100 的 BDBR 值较大, 原因是以上序列场景

运动比较剧烈或者景物切换频繁, 造成算法估计不准, 例如 Kimono 序列中的背景和物体同时移动。如 E 类视频序列由于场景中背景和物体相对静止, 因此算法估计比较准确。

表 3 RA 配置下 JMFC 算法与其他算法性能比较

Tab. 3 Performance comparison of JMFC with other algorithms under RA configuration (%)

测试序列	Yamamoto[3] (JEM-3.0)		Lin[4](JEM-5.0)		JCDT [6](JEM-7.0)		JMFC(JEM-7.0)	
	BDBR	$\Delta T$	BDBR	$\Delta T$	BDBR	$\Delta T$	BDBR	$\Delta T$
A1 Drums100	+0.9	-20.0	+0.2	-10.0	+1.8	-14.0	+9.5	-36.8
A2 TrafficFlow	+0.3	-20.0	+0.1	-9.0	+0.8	-10.8	+3.7	-34.3
B Kimono	+0.5	-16.8	+0.2	-10.6	+0.4	-12.8	+8.2	-41.2
ParkScene	+0.4	-17.8	+0.1	-8.5	+2.9	-18.8	+2.9	-32.3
Cactus	+0.4	-18.4	+0.2	-10.0	+1.7	-14.8	+4.7	-34.0
BasketballDrive	+0.7	-17.5	+0.2	-9.5	+1.0	-15.0	+9.0	-40.2
BQTerrace	+0.4	-17.6	+0.2	-8.5	+2.6	-16.1	+2.4	-32.3
C BasketballDrill	+0.5	-22.2	+0.2	-9.0	+1.2	-17.9	+6.1	-41.0
BQMall	+0.5	-19.4	+0.3	-7.9	+3.8	-20.9	+2.5	-36.0
PartyScene	+0.4	-18.6	+0.1	-7.7	+2.9	-17.7	+2.2	-32.5
RaceHorsesC	+0.7	-21.5	+0.2	-7.7	+2.7	-17.6	+4.1	-37.8
D BasketballPass	+0.5	-17.7	+0.1	-7.2	+3.1	-26.3	+3.3	-34.6
BQSquare	+0.2	-15.4	+0.1	-7.4	+3.9	-25.5	+1.1	-30.1
BlowingBubbles	+0.3	-17.5	+0.1	-7.1	+3.4	-23.8	+2.7	-35.7
RaceHorses	+0.6	-18.2	+0.1	-7.3	+4.3	-24.1	+2.8	-36.5
平均值	+0.5	-18.6	+0.2	-8.5	+2.4	-18.4	+4.3	-35.7

表 4 JMFC 算法在 LD 配置下的算法性能

Tab. 4 Performance of JMFC with LD configuration (%)

测试序列	LDP		LDB	
	BDBR	$\Delta T$	BDBR	$\Delta T$
A1 Drums100	+6.1	-25.3	+6.2	-26.2
A2 TrafficFlow	+3.1	-31.4	+2.8	-23.1
B Kimono	+5.8	-37.3	+5.7	-41.6
ParkScene	+2.6	-21.9	+2.4	-22.7
Cactus	+3.8	-25.7	+3.5	-27.9
BasketballDrive	+6.5	-28.4	+6.2	-29.0
BQTerrace	+2.3	-22.3	+2.0	-23.1
C BasketballDrill	+4.9	-26.4	+4.6	-27.5
BQMall	+1.8	-24.5	+1.6	-25.3
PartyScene	+1.6	-36.0	+1.4	-36.5
RaceHorsesC	+3.0	-27.2	+2.9	-28.5
D BasketballPass	+2.2	-23.8	+2.0	-26.3
BQSquare	+0.7	-18.6	+1.0	-20.3
BlowingBubbles	+2.0	-23.7	+2.2	-24.8
RaceHorses	+2.1	-25.1	+2.1	-25.6
E FourPeople	+3.2	-21.1	+2.7	-23.0
Johnny	+2.0	-21.9	+1.5	-24.3
KristenAndSara	+1.2	-21.0	+1.2	-24.1
平均值	+3.1	-25.6	+2.89	-26.7

表 5 LDB 配置下 JMFC 算法与其他算法性能比较

Tab. 5 Performance comparison of JMFC algorithm with other algorithms under LDB configuration (%)

测试序列	Park[5] (JEM-3.1)		Lin[4] (JEM-5.0)		JMFC(JEM-7.0)	
	BDBR	$\Delta T$	BDBR	$\Delta T$	BDBR	$\Delta T$
B Kimono	+0.2	-13.0	+0.2	-10	+5.7	-41.6
	+0.3	-11.0	+0.2	-5.0	+2.4	-22.7
	+0.3	-13.0	+0.2	-7.3	+3.5	-27.9
C BasketballDrill	+0.2	-12.0	+0.2	-6.5	+4.6	-27.5
	+0.2	-13.0	+0.2	-5.6	+1.6	-25.3
	+0.5	-9.0	+0.1	-5.3	+1.4	-36.5
D BasketballPass	+0.2	-12.0	+0.2	-7.5	+2.0	-26.3
	+0.4	-10.0	+0.2	-7.2	+2.2	-24.8
E FourPeople	+0.2	-16.0	+0.0	-6.0	+2.7	-23.0
	+0.4	-16.0	+0.0	-4.1	+1.5	-24.3
	+0.2	-16.0	+0.2	-5.4	+1.2	-24.1
	+0.3	-12.8	+0.2	-6.0	+2.6	-27.7

如图 7(a) 和 7(b) 所示, 分别为视频序列 Drums100 和 Kimono 采用原始 JEM-7.0 平台编码后和本文 JMFC 算法编码后的率失真(Rate Distortion, RD)曲线, 从图中可以看出, 二者与原始率失真曲线基本重合, 由于 Y-PSNR 表征了视频图像质量, 因此视频质量没有太大损失。从图 7 中还可以看出, 本文提出的算法采用小 QP 编码时要优于大 QP 编码。

后图像没有明显差异, 进一步说明该算法在节省编码时间的同时并未造成视频主观质量的下降。

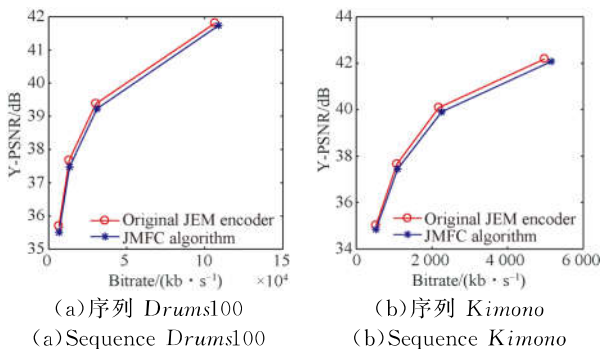


图 7 RA 配置下视频序列利用原始 JEM 平台编码与 JMFC 算法编码后的率失真性能曲线比较

Fig. 7 RD curves of two video sequences encoded by original JEM encoder and JMFC algorithm

如图 8(a) 为序列 Drums100 原始视频图像和其局部图像, 图 8(b) 为利用 JMFC 编码后的图像和其局部图像。在主观质量上, 原始图像和编码



(a) 原始图像和其局部图像  
(a) Original image and local enlargement



(b) JMFC 算法编码图像和其局部图像  
(b) Coded image and local enlargement by JMFC algorithm

图 8 视频序列 Drums100 的第 2 帧图像  
Fig. 8 Second frame image of Drums100 sequence

## 5 结 论

本文针对 FVC 编码复杂度升高的问题, 提出了 JMFC 算法。本文算法主要贡献有: 联合 FVC 中多个编码特征的划分概率信息, 在 QTBT 划分前, 预测当前 CU 的划分方式, 提前终止 CU 的划分, 降低编码复杂度。联合多个编码特征的先验

概率信息,提出一种解决单个弱分类模型不准确的强化提升方法。实验结果表明,在 RA 配置、LDP 配置和 LDB 配置下,本文提出的算法的编码复杂度,可以平均减少 35.7%,25.6% 和 26.7% 而 BDBR 只分别增加了 4.3%,3.1% 和

2.89%。在未来的工作,主要针对编码复杂度,设计出更加有效的快速编码算法,使编码性能更加优化。本文针对 H.266/FVC 编码标准,提出了有效的视频快速编码技术,在保证视频图像主观质量的前提下,节省了编码时间。

#### 参考文献:

- [1] CHEN J, ALSHINA E, SULLIVAN J G, *et al.*. Algorithm description of joint exploration test model 1 [C]. *1st Meeting Joint Video Exploration Team (JVET)*, Document JVET-A1001, Geneva, 2015: 1-25.
- [2] WANG Z, WANG S, ZHANG J, *et al.*. Local-constrained quadtree plus binary tree block partition structure for enhanced video coding [J]. *2016 Visual Communications and Image Processing (VCIP)*, 2016:1-4.
- [3] YAMAMOTO Y, IKAI T. AHG5: Fast QTBT encoding configuration[C]. *4th Meeting Joint Video Exploration Team (JVET)*, Document JVET-D0095, Chengdu, 2016:1-6.
- [4] LIN P H, LIN C L, CHANG Y Z, *et al.*. AHG5: Enhanced fast algorithm of JVET-E0078 [C]. *6th Meeting Joint Video Exploration Team (JVET)*, Document JVET-F0063, Hobart, 2017:1-2.
- [5] PARK S H, DONG T, JANG E S. Low complexity reference frame selection in QTBT structure for JVET future video coding [J]. *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018:1-4.
- [6] WANG Z, WANG S, ZHANG J, *et al.*. Effective quadtree plus binary tree block partition decision for future video coding [J]. *2017 Data Compression Conference (DCC)*, 2017:23-32.
- [7] WANG Z, WANG S, ZHANG J, *et al.*. Probabilistic decision based block partitioning for future video coding [J]. *IEEE Transactions on Image Processing*, 2018, 27(3): 1475-1486.
- [8] HU Q, SHI Z, ZHANG X, *et al.*. Early SKIP mode decision based on Bayesian model for HEVC [J]. *2015 Visual Communications and Image Processing (VCIP)*, 2015:1-4.
- [9] CHEN J, YU L. Effective HEVC intra coding unit size decision based on online progressive Bayesian classification [J]. *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016:1-6.
- [10] KIM H S, PARK R H. Fast CU partitioning algorithm for HEVC using an online-learning-based bayesian decision rule [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016, 26(1): 130-138.
- [11] LI Y, LIU X, YU T, *et al.*. Fast online-learning parameters decision algorithm based on bayesian decision rule for 3D-HEVC [J]. *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2016:894-898.
- [12] ZHANG J, KWONG S, WANG X. Two-stage fast inter CU decision for HEVC based on bayesian method and conditional random fields [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 28(11): 3223-3235.
- [13] AL-BASHABSHEH A, MAO Y, VONTOBEL P O. Normal factor graphs: A diagrammatic approach to linear algebra [J]. *2011 IEEE International Symposium on Information Theory Proceedings*, 2011:2178-2182.
- [14] LOELIGER H A, VONTOBEL P O. A factor-graph representation of probabilities in quantum mechanics [J]. *2012 IEEE International Symposium on Information Theory Proceedings*, 2012: 656-660.
- [15] TSAI T H, SU S S, LEE T Y. Fast mode decision method based on edge feature for HEVC inter-prediction [J]. *IET Image Processing*, 2018, 12(5):644-651.
- [16] 李大维,孙海江,刘伟宁,等.基于统计信息的改进滑动平均目标检测算法 [J]. *液晶与显示*, 2018, 33(6):497-503.
- [17] LI D W, SUN H J, LIU W N, *et al.*. Improved sliding average target detection algorithm based on statistical information [J]. *Chinese Journal of Liquid Crystals and Displays*, 2018, 33(6): 497-503. (in Chinese)
- [17] 陆牧,朱明,高扬,等.基于元胞自动机的动态背景运动目标检测 [J]. *光学 精密工程*, 2017, 25(7):

- 1934-1940.  
LU M, ZHU M, GAO Y, *et al.*. Moving target detection based on dynamic background of cellular automaton [J]. *Opt. Precision Eng.*, 2017, 25(7): 1934-1940. (in Chinese)
- [18] 周同雪,朱明. 视频图像中的运动目标检测 [J]. 液晶与显示, 2017, 32(1): 40-47  
ZHOU T X, ZHU M. Moving target detection of the video images [J]. *Chinese Journal of Liquid Crystals and Displays*, 2017, 32(1): 40-47. (in Chinese)
- [19] 张强, 邵喜斌, 杨虹. 大尺寸 8K 显示行业关键技术分析 [J]. 液晶与显示, 2018, 33(5): 389-396.  
ZHANG Q, SHAO X B, YANG H. Key technology analysis of large size 8K display [J]. *Chinese Journal of Liquid Crystals and Displays*, 2018, 33(5): 389-396. (in Chinese)
- [20] 蒋刚毅, 杨小祥, 彭宗举, 等. 高效视频编码的快速编码单元深度遍历选择和早期编码单元裁剪 [J]. 光学精密工程, 2014, 22(5): 1322-1330.  
JIANG G Y, YANG X X, PENG Z J, *et al.*. Fast CU depth range selection and early CU pruning for HEVC [J]. *Opt. Precision Eng.*, 2014, 22(5): 1322-1330. (in Chinese)

#### 作者简介:



崔鑫(1986—),男,山东济南人,博士研究生,2013年于山东科技大学获得硕士学位,主要从事新一代高效视频编码技术(HEVC)、多媒体信息处理等领域研究。E-mail: cuixin2461@163.com



彭宗举(1973—),男,四川南部人,教授,博士生导师,1995年于四川师范学院获得学士学位,1998年于四川大学获得硕士学位,2010年于中国科学院计算技术研究所获得博士学位,主要从事三维视频信号处理、传输与编码等方面的研究。E-mail: pengzongju@nbn.edu.cn

(本栏目编辑:秦思)