

## 计算资源受限的视频编码多模式决策\*

陆寄远<sup>1,2</sup>, 侯 昉<sup>1</sup>, 黄承慧<sup>1</sup>, 刘宇熹<sup>1</sup>, 朝红阳<sup>2</sup>

<sup>1</sup>(广东金融学院 计算机科学与技术系, 广东 广州 510521)

<sup>2</sup>(中山大学 软件学院, 广东 广州 510006)

通讯作者: 陆寄远, E-mail: dtc005001@163.com

**摘要:** 由于视频编码技术趋向于采用越来越复杂的分块模式,多模式决策技术也随之成为一种非常重要的编码技术.多模式决策的优劣不仅会大幅度地影响视频编码的计算消耗,而且对编码性能的高低起到关键的作用.为使多模式决策在计算能力相差悬殊的平台上都能获得优化的率失真性能,给出一种计算复杂度自适应的优化多模式决策算法.首先,利用视频序列中不同宏块模式间的时空相关性,预测这些宏块多模式决策后的拉格朗日代价和计算复杂度的斜率(Lagrangian cost and complexity slope,简称 J-C slope).J-C slope 越大,说明在该宏块上的模式决策消耗每单位的计算资源可以获取的率失真收益越多.在计算资源有限的情况下,多模式决策应该按照 J-C slope 的大小顺序执行,也就是性价比优先的顺序,以便保证计算资源优先分配给率失真收益大的宏块.另外,还通过建立 J-C slope 阈值与实际计算复杂度的关系模型,设计了一种根据给定计算约束自适应调整计算复杂度的算法.根据实验结果,该方法不仅可以准确地控制多模式决策的计算复杂度,而且还能在不同的计算约束下获得优化的率失真性能.

**关键词:** 多模式决策;计算复杂度自适应;率失真优化;视频编码

**中图法分类号:** TP391

中文引用格式: 陆寄远,侯昉,黄承慧,刘宇熹,朝红阳.计算资源受限的视频编码多模式决策.软件学报,2014,25(11):2690-2701.  
<http://www.jos.org.cn/1000-9825/4695.htm>

英文引用格式: Lu JY, Hou F, Huang CH, Liu YX, Chao HY. Multi-Mode decision under computational resource constraints for video coding. Ruan Jian Xue Bao/Journal of Software, 2014,25(11):2690-2701 (in Chinese). <http://www.jos.org.cn/1000-9825/4695.htm>

### Multi-Mode Decision Under Computational Resource Constraints for Video Coding

LU Ji-Yuan<sup>1,2</sup>, HOU Fang<sup>1</sup>, HUANG Cheng-Hui<sup>1</sup>, LIU Yu-Xi<sup>1</sup>, CHAO Hong-Yang<sup>2</sup>

<sup>1</sup>(Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou 510521, China)

<sup>2</sup>(School of Software, Sun Yat-Sen University, Guangzhou 510006, China)

Corresponding author: LU Ji-Yuan, E-mail: dtc005001@163.com

**Abstract:** As more and more flexible block modes are introduced into video coding, mode decision technology becomes an important coding tool. The performance of mode decision has great impact on both coding performance and computational complexity. This article proposes a complexity controllable multi-mode decision algorithm to attain optimized coding performance under different computational complexity constraints herein. Instead of speeding up multi-mode decision merely, the algorithm predicts the Lagrangian cost and complexity slope (J-C slope) of MD for each macroblock (MB) by exploiting their temporal and spatial correlations. The larger J-C slope is, the higher coding gain over each unit of computational cost will be. In the environment of limited computational resources, multi-mode decision should be performed in a cost effective order, i.e. the order of their J-C slopes, to achieve optimized coding performance. In addition, an adaptive method is proposed to adjust the computational complexity of the algorithm dynamically by discovering the relationship of J-C slope thresholds and their corresponding complexity. Experiments demonstrate the proposed algorithm can both precisely adjust the computational complexity and optimally perform mode decision under different computational constraints.

\* 基金项目: 国家自然科学基金(61173081); 广东省自然科学基金(S2011020001215, S2012040007847)

收稿时间: 2013-07-18; 定稿时间: 2014-07-09

**Key words:** multi-mode decision; adaptive computational complexity; rate distortion optimization; video coding

硬件与网络技术的不断发展,引发了视频编码技术的广泛应用.新的视频编码技术虽然拥有很好的率失真性能,但同时也消耗了更多的计算资源.随着多媒体视频技术越来越多地出现在各种计算能力相差悬殊的平台上,计算消耗不可控的编码算法成为这些应用最大的阻碍.虽然编码算法可以通过多媒体指令进行优化,但是在当前对高清视频需求日益增长的环境下,单纯地快速优化,其效果是极其有限的<sup>[1]</sup>.

视频编码标准如 H.26x,AVS,VC-1 等为了提高整体的压缩性能,采用了各种新的编码技术,如不同大小分块的多模式运动补偿、多参考帧和高精度分数运动估计、更为复杂的环路滤波器和上下文自适应算术编码等.其中,多模式决策就是为采用多模式运动补偿技术的宏块选择一种拉格朗日代价函数值最小的模式.早期的视频编码标准只使用 16×16 宏块作为唯一的运动补偿基本单元,不存在多模式决策问题.现存的几大视频编码标准,如 H.26x/MPEG-x,AVS 和 VC-1,都把运动补偿分块从单一宏块(16×16)扩展到了 4×4.如图 1 所示,以 H.264 为例,每一宏块的模式可以是 16×16,16×8,8×16 或 8×8.其中,如果宏块的模式是 8×8,那么还需要为每一个 8×8 的分块指定一个子模式,该子模式可以是 8×8,8×4,4×8 或 4×4.每个分块都拥有各自的运动向量,以适应不同的运动情况.在细节繁琐、运动剧烈的部分,应该采用比较小的分块;反之,在图像比较平坦、变化比较单一的部分,则倾向于采用较大的分块.多模式技术虽然提高了运动补偿的性能,但也增加了模式决策的计算复杂度.以支持 4×4 分块的视频编码器为例,使用全模式搜索所需的计算量是单一模式搜索的 7 倍.

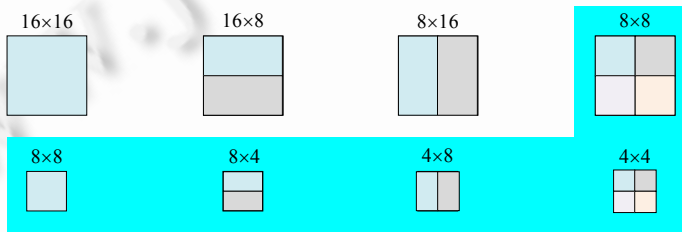


Fig.1 Seven modes of H.264/AVC  
图 1 H.264/AVC 中的 7 种模式

此外,可伸缩视频编码(scalable video coding,简称 SVC)在上述 H.264 的基础上引入了更多的编码模式,从而进一步增加了模式决策的计算消耗.SVC 是视频编码标准 H.264 继面向高清视频的修订版(H.264 high profile)后的又一修订,解决了一次编码满足各种不同再现能力终端需求问题<sup>[2]</sup>.SVC 码流的可伸缩性主要体现在时域、空域和质量这 3 方面.图 2 是 SVC 应用的一个范例.

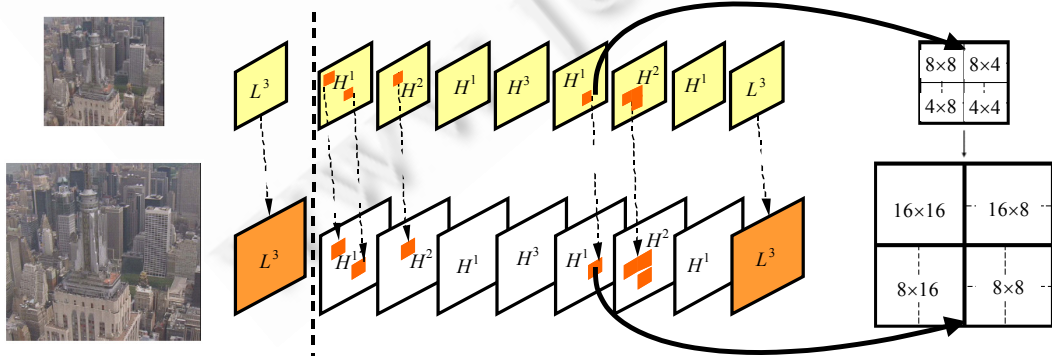


Fig.2 Different modes of H.264 SVC  
图 2 H.264 SVC 中不同的模式

SVC引入了层(layer)这个概念以区分不同分辨率和不同质量的视频信号,每层的图像都被分割为一个个宏块后才进行编码.因为层与层之间的图像仅在分辨率或质量上有不同,所以这些图像在内容上有着很强的相关性.SVC与常用的单层视频编码标准(如H.264)的区别在于额外提供了不少提取层间相关性的编码工具.SVC虽然可以很方便地适应不同带宽容量和计算能力的解码终端,但却在编码端消耗了很多的计算资源.尤其在同时提供上述3个方面的可伸缩性时,每个宏块可供选择的编码模式比以往的标准H.264或H.264 High Profile都要多得多,从而导致了模式决策计算变得复杂.使用穷举遍历的方法虽然可以找到最优的模式,但所耗费的计算量是非常庞大的.

除了以上标准以外,新一代视频编码标准HEVC不仅采用了更为复杂的CU/PU/TU编码结构,而且还扩大了每个宏块的尺寸<sup>[3]</sup>.图3是在这种编码结构下可以具有的不同分块模式.这种新的分块模式导致模式决策算法必须具备在不同平台上的自动适应能力,否则,HEVC将难以在实际应用中快速推广.

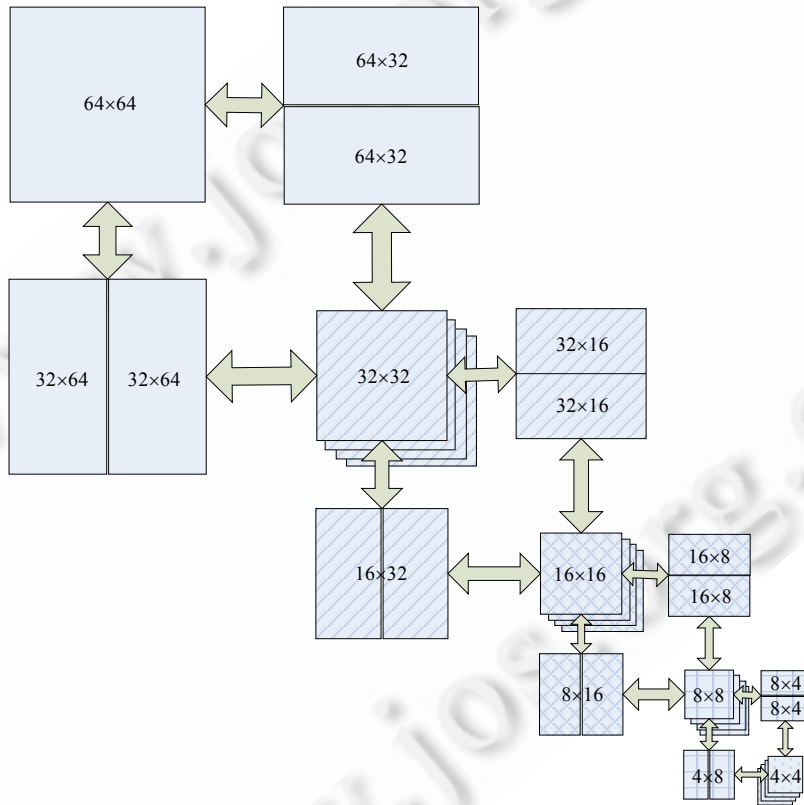


Fig.3 Relationship of different modes of HEVC

图3 HEVC中不同模式之间的关系

为了在不同的硬件平台上准确地进行多模式决策,业界提出了不少针对多模式决策的优化算法.这些优化算法大致划分为两种思路.一类是传统的快速算法.Kim等人<sup>[4]</sup>提出了一种根据视频时空相关性和运动特性的快速帧内/帧间选择算法.Ri等人<sup>[5]</sup>采用相邻宏块的模式信息和相应的拉格朗日代价进行快速模式决策.Vatis等人<sup>[6]</sup>利用一系列从运动补偿中得到的信息训练一组学习机制,以确定最优的模式.Zeng等人<sup>[7]</sup>通过评估时域和空域相邻宏块的运动情况获得最优的模式.Lee等人<sup>[8]</sup>提出了一系列的技巧进行快速模式决策,以大幅度地减少模式决策的庞大计算消耗,这些技巧包括提前SKIP模式确认(early SKIP mode decision)、宏块一致性确认(homogeneity determination)和候选模式选择(candidate mode selection).Shen等人<sup>[9]</sup>利用4x4模式下空域相邻的运动向量信息,提出了一种快速的多模式决策算法,虽然这种利用运动向量场预测帧间模式的方法是十分有效

的,但其不足之处在于:4×4 模式下的所有运动向量必须先被得出,而这将导致额外的计算消耗。所有上述的这些快速算法都只着眼于尽可能多地提高多模式决策速度,而对于根据不同的计算资源提供可伸缩的率失真性能则欠缺考虑。这种可伸缩的特性恰恰是在不同硬件平台上使用多模式决策编码技术的关键<sup>[2]</sup>。另一类算法是可以提供计算和编码效率可伸缩性的复杂度可调整多模式决策算法(complexity adjustable mode decision,简称 CAMD)。Su 等人<sup>[10]</sup>通过选择不同的候选帧间模式集(candidate multi-mode set)调整多模式决策的计算复杂度。Akyol 等人<sup>[11]</sup>提出通过调整模式搜索阈值(mode search gradient threshold)控制计算复杂度。因为这两种方法采用固定的策略调整计算复杂度,所以在某些情况下难以自动适应不同的视频内容,算法的性能仍有进一步优化的空间。为了提高编码算法对视频内容的适应性,Luo 等人<sup>[12]</sup>提出一种视频内容自适应的快速运动估计方法。每个宏块根据各自不同的特性采用不同的运动估计策略,并根据率失真收益和计算复杂度斜率(R-D coding gain and complexity slope)分配不同宏块的计算复杂度。但这种方法只适用于视频编码中的运动估计。要把这种方法扩展到多模式决策中,不仅需要综合地利用两者的优点,还需根据多模式决策的特征做重新的设计。

上述这些 CAMD 算法只关注了如何调整计算复杂度的问题。对于如何准确控制计算复杂度的问题,Akyol 等人<sup>[11]</sup>使用了机器训练的方法控制视频编码的相关参数。这种方法采用离线的方式训练参数,其准确率极其依赖于训练所使用的样本,难以动态适应不同视频的情况。

总括上面的所有情况,在不同的硬件平台上进行优化的多模式决策需要解决两个核心问题:一是如何设计复杂度可调的算法,使其在不同计算约束下都能提供最优化的决策性能;二是如何自适应地调整算法参数,以使算法的复杂度精准地满足实际约束。

为了解决这两个核心问题,本文给出一种计算复杂度自适应的多模式决策算法。该算法按照拉格朗日代价和计算复杂度斜率(Lagrangian cost and complexity slope,简称 J-C slope)的大小顺序调整模式决策的计算复杂度,在计算资源受限的情况下,只对每帧中的部分宏块实施多模式决策。为了确定要对哪一部分的宏块实施多模式决策才能得到优化的率失真性能,必须首先确定每个宏块的优先级,较高的优先级表示在相同计算消耗的情况下能够获得更好的率失真性能。为此,本文用 J-C slope 表示不同宏块的优先级。由于各个宏块的 J-C slope 只能在模式决策完成后才可得出,所以不能被直接使用。为了解决这个问题,本文不仅利用 J-C slope 在时间轴上的连续性,而且通过空域相邻的 J-C slope 插值进行预测。这些预测的 J-C slope 被作为宏块多模式决策的优先顺序,用以决定当前宏块是否进行多模式决策,或者只执行 16×16 的运动估计。

此外,对于如何根据给定的计算约束准确地调整算法参数的问题,本文自适应地为每帧计算一个 J-C slope 的阈值,以决定到底有多少个宏块将会采用多模式决策。由于采用多模式宏块的数目与多模式决策的计算复杂度相关,而该数目的多少则是由 J-C slope 阈值所控制,本文给出一个根据给定的计算约束自适应地调整该阈值的方法。由于本文的算法只针对某个宏块是否实施多模式决策进行控制,并不涉及某个宏块上具体的决策算法,所以该算法还可以与任意的快速多模式决策算法配合使用,以达到更好的效果。

## 1 时空域的 J-C slope 预测方法

当前,视频编码领域中一个非常热门的课题是研究在各种情况下如何获得最佳的率失真性能<sup>[13,14]</sup>,其中的一个核心问题就是如何提高各种编码方法对视频内容的自适应性。根据文献<sup>[15]</sup>的研究可知:要获得优化的编码性能,需要对视频序列中运动平缓和内容相近的区域采用大的分块模式编码;而对于运动剧烈和细节复杂的区域,则采用小的分块模式。由于物体的轮廓和运动轨迹都有一定程度的连续性,所以无论时域相邻宏块还是空域相邻宏块,其分块模式都具有较强的相关性。也就是说,如果一个占据几个相邻宏块的物体具有剧烈的运动或者复杂的细节,这些宏块在模式的选择上也都会倾向于较小的分块模式;而一个运动平缓或细节简单的物体,则趋向于选择比较大的分块模式。同样的结果也体现在时域相邻帧上:如果当前帧和前一帧的内容没有太大的变化,那么当前的宏块和前一帧相应位置的宏块很可能表示同一个物体,它们在模式的选择上也会倾向于采用相同的模式。

图 4 是利用 H.264 兼容编码器对 Paris 序列编码后不同宏块模式的标识,从图中可看出,大多数的宏块使用

了 16×16 模式.这些宏块的内容均是平坦和静止的,小的分块模式不会出现在这些平坦的内容中.此外,图中背景(如书架和书本)的纹理虽然十分复杂,但这些背景所在的宏块也都采用了 16×16 模式.这是因为这些宏块的内容是静止的,在运动补偿时仍适合采用 16×16 的大分块模式.另一方面,图中人物的嘴唇和手掌所涉及的宏块不仅带有复杂的纹理,而且有着剧烈的运动,所以这些宏块都使用较小的分块模式.而这些小的分块模式主要出现在运动物体的边缘,如人物轮廓、手掌和面部等.

为了方便研究不同模式的相关性,本文把宏块间的相邻关系分成两类(如图 5 所示,其中,第  $n$  帧中的当前块表示当前要进行模式决策的宏块):第 1 类的相邻关系是时域相邻关系,即,在相邻两帧中相同位置的宏块;第 2 类的相邻关系是空间相邻关系,因为视频编码按照光栅顺序(即从左到右,从上到下)编码,所以空间相邻块通常指左上、上、右上和左 4 个方向的宏块.这两类相邻宏块的模式之间具有很高的相关性,有效地利用该相关性预测当前块的模式,可以大幅度减少模式决策的运算.



Fig.4 An example of multi-modes for Paris  
图 4 Paris 视频序列中多模式的例子

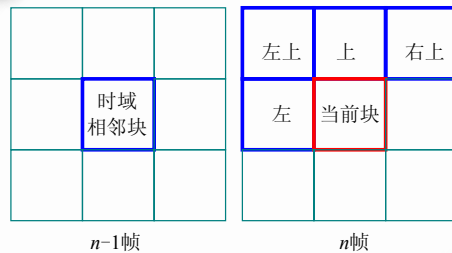


Fig.5 Different neighboring blocks  
图 5 不同的相邻块

在展开具体论述之前,下面首先介绍一些基础知识.拉格朗日代价(Lagrangian cost)是度量多模式预测方法失真度和码率综合性能的指标.在视频编码中,多模式决策的率失真性能可以用拉格朗日代价来表示.拉格朗日代价越小,率失真性能就越好;反之,拉格朗日代价越大,则率失真性能越差.J-C slope 表示拉格朗日代价在单位计算复杂度上的变化速率.要在同等计算资源的条件下获得最好的率失真性能,就要优先在 J-C slope 大的宏块上采用复杂的编码技术.比如,Luo 等人<sup>[12]</sup>提出了一种基于内容自适应的运动估计算法.该算法根据每个宏块各自特性的不同采用相应的运动估计策略.与之不同的是,本文结合了多模式决策的特点,按照不同宏块的 J-C slope 特性决定是否采用多模式决策.

首先,采用 J-C slope 作为对多模式决策率失真性能和计算代价之间关系的度量指标,其通用计算公式为

$$s = \frac{\Delta J}{\Delta C} \tag{1}$$

其中, $\Delta J$  表示某一编码操作可以减少的拉格朗日代价值, $\Delta C$  表示某一编码操作所带来的计算增量.某一编码操作的 J-C slope 较大,表示该操作耗费单位计算时间所得到的率失真收益越大,该操作应该获得的计算资源就越多;反之亦然.针对宏块 $(i,j)$ 多模式决策的 J-C slope 计算公式如下:

$$s_{(i,j)}^n = \frac{\Delta J_{(i,j)}^n}{\Delta C_{(i,j)}^n} = \frac{J_{16 \times 16(i,j)}^n - J_{multi-mode(i,j)}^n}{C_{multi-mode(i,j)}^n} \tag{2}$$

其中, $n$  表示当前帧的帧号, $(i,j)$ 表示宏块在当前帧中的位置. $J_{16 \times 16(i,j)}$ 和  $J_{multi-mode(i,j)}$ 分别表示仅使用 16×16 模式和使用多模式决策后的拉格朗日代价. $C_{multi-mode(i,j)}$ 表示使用多模式决策后额外增加的计算量.公式(2)的物理意义是各个宏块采用多模式决策的有效性,J-C slope 越大的宏块,采用多模式决策后对编码率失真性能贡献就越大.计算复杂度的分配应该按照 J-C slope 的大小分配.

由于真正的 J-C slope 需要宏块在完成多模式决策后才可以准确地得出,这与使用 J-C slope 来决定多模式决策的计算量恰恰是相互矛盾的.为了解决这个矛盾,本文给出了一种时空域的 J-C slope 动态预测补偿方法.

首先,使用前一帧对应宏块的 J-C slope 预测当前宏块的 J-C slope.因为视频序列中的对象具有很强的时域相关性,所以除了在场景大幅度变换的情况下以外,相同位置宏块的 J-C slope 会非常相近.此外,J-C slope 值较大的宏块都有复杂的纹理或剧烈的运动,这些宏块的预测残差也会相应较大.因此,16×16 模式下的拉格朗日代价也对 J-C slope 的计算起着重要的作用.综合考虑上述两个因素,本文使用下面的公式对 J-C slope 进行近似计算:

$$\hat{s}_{(i,j)}^n = \frac{\Delta J_{(i,j)}^n}{\Delta C_{(i,j)}^n} = s_{(i,j)}^{n-1} \times \frac{J_{16 \times 16(i,j)}^n}{J_{16 \times 16(i,j)}^{n-1}} \quad (3)$$

其中,  $\hat{s}_{(i,j)}^n$  是当前宏块 J-C slope 的预测值;  $s_{(i,j)}^{n-1}$ ,  $J_{16 \times 16(i,j)}^{n-1}$  和  $J_{16 \times 16(i,j)}^n$  分别是前一帧相应宏块的实际 J-C slope、前一帧 16×16 模式下的拉格朗日代价和当前帧 16×16 模式下的拉格朗日代价.这三者都可以在当前宏块进行模式决策前得到.

然而,时域方法的不足之处在于必须准确地知道前一帧对应宏块的 J-C slope.也就是说,前一帧的所有宏块都要已经执行了多模式决策.在计算资源受限的情况下,多模式决策仅会作用在 J-C slope 大的那部分宏块上,所以前一帧对应宏块的 J-C slope 并不保证都可以得到.为了解决这一问题,空间域的信息可被作为对 J-C slope 计算的补偿和校正.我们通过计算空间相邻运动向量的不连续性(motion vector discontinuity,简称 MVD),间接得出 J-C slope.MVD 是指当前宏块 16×16 模式的运动向量与空间相邻运动向量的相异性,其计算公式如下:

$$MVD = \frac{1}{n} \sum_{i=1}^n \frac{|MV_i - MV_{16 \times 16}|}{|MV_{16 \times 16}|} \quad (4)$$

其中,  $MV_{16 \times 16}$  表示当前宏块运动向量,  $MV_i$  表示空域相邻宏块运动向量.图 6 是这些相邻运动向量的示例.浅色的相邻运动向量就是  $MV_i$ ,当前宏块在进行模式决策前必然先完成 16×16 下的运动估计,从而得出如图所示的深色运动向量  $MV_{16 \times 16}$ .这些运动向量在多模式决策前都是已知的.如果相邻宏块包含不同的运动物体,该宏块的 MVD 则较大,对该宏块多模式决策所获得的率失真收益就会较高;相反,相邻宏块如果属于同一个物体,则 MVD 较小,那么多模式决策在这些宏块上对率失真性能的影响则相对较低.

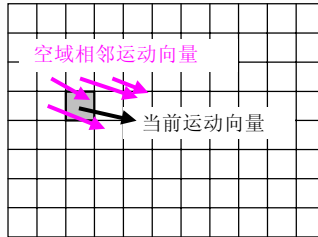


Fig.6 Motion vectors of neighboring macro-blocks

图 6 相邻宏块运动向量

当完成对 MVD 的计算后,可以通过 MVD 预测出 J-C slope.每个已经进行了多模式决策的宏块,都有一组对应的 J-C slope 和 MVD:

$$(s_{(i,j)}^n, MVD_{(i,j)}^n) \quad (5)$$

下面根据当前宏块的 MVD 选出两组与该 MVD 最接近的数值.当前宏块的 J-C slope 通过对这两组数值进行线性插值得出:

$$\hat{s}_{(i,j)}^n = \frac{(s_1^{n-1} - s_2^{n-1})}{(MVD_1^{n-1} - MVD_2^{n-1})} \times (MVD_{(i,j)}^n - MVD_1^{n-1}) + s_1^{n-1} \quad (6)$$

其中,  $s_1^{n-1}$ ,  $MVD_1^{n-1}$ ,  $s_2^{n-1}$  和  $MVD_2^{n-1}$  分别是前一帧两组与当前宏块的  $MVD_{(i,j)}^n$  最接近的 J-C slope 和 MVD.当前宏块的 J-C slope 就是通过对这两组数的插值计算出来.

当宏块的 J-C slope 都计算出来以后,计算资源的分配便按照该值的大小进行.J-C slope 大的宏块获取更多的计算资源;J-C slope 小的宏块则获取较少的计算资源.这样就保证了计算资源以优化的率失真性能为目标进行分配.不同的快速多模式算法只要配合本文方法进行了改进和资源合理分配后,在同样计算资源的约束下,要比单纯采用快速算法在率失真性能方面有显著的提高.

## 2 J-C slope 阈值的确定

到目前为止,前面所提到的方法已经完成了对 J-C slope 的预测.接下来要解决的问题就是:如何满足任意给定的计算复杂度约束,达到最优化的编码性能.对于该问题,我们通过为每帧自动设定一个 J-C slope 阈值确定参与模式决策宏块的数目,从而达到控制整体算法计算复杂度的目的:若当前宏块的预测 J-C slope 大于该阈值,则进行多模式决策;否则只进行  $16 \times 16$  模式的运动估计.恰当地设置该阈值,就是确定算法实际计算复杂度大小的关键.下面将把 J-C slope 阈值作为模式决策的控制参数,把当前帧需要进行多模式决策的宏块数目作为模式决策的计算复杂度,并给出具体算法来调整 J-C slope 阈值,令当前帧需要进行多模式决策的宏块数目能够准确达到所给定的要求.该方法仅运用简单的代数运算,无需使用 Akyol 等人<sup>[11]</sup>所提到的机器学习和训练过程,极大地简化了使用难度.

我们用  $N(t)$  表示当前帧多模式决策的实际计算复杂度(需要进行多模式决策的宏块数目), $T(t)$  表示控制参数(J-C slope 阈值),两者都是时间  $t$  的函数.而目标计算复杂度是事先给定的常数,用  $N_{target}$  表示.

复杂度控制的目标就是通过调整  $T(t)$ , 令  $N(t)$  尽可能地逼近  $N_{target}$ . 因为  $T(t)$  与  $N(t)$  的关系是动态变化的,所以  $N(t)$  难以 100% 地吻合  $N_{target}$ , 总会出现或多或少的偏差.为此,需要根据  $N_{target}$  和  $N(t)$  的差值来调整下一时刻的计算复杂度. $N_{target}-N(t)$  越大, $N(t)$  的变化率应该越快;反之,则应该越慢.如果  $N_{target}-N(t)$  为 0, 则  $N(t)$  不需要改变.根据以上描述,可以用下面的常微分方程表示  $N(t)$  向  $N_{target}$  逼近的过程:

$$\frac{dN}{dt} = (N_{target} - N(t)) \quad (7)$$

该方程表明, $N(t)$  的变化率(一阶导数)与  $N_{target}-N(t)$  成正比.图 7 是该方程在时刻  $t$  的图示.

由于前一帧多模式决策的计算复杂度  $N(t-1)$  是已知的,因此  $N(t)$  的求解可以转化为下面的常微分方程初值问题:

$$\begin{cases} \frac{dN}{dt} = N_{target} - N(t) \\ N(t-1) = N_{t-1} \end{cases} \quad (8)$$

其中, $N_{t-1}$  为  $t-1$  时刻多模式决策的计算复杂度,可以在上一帧多模式决策完成后计算出来.方程(8)的解可以表示为

$$N(t) = N_{target} - (N_{target} - N_{t-1}) \cdot e^{-1} \quad (9)$$

公式(9)表示在理想状态下, $N(t)$  向  $N_{target}$  逼近的轨迹是一条如图 7 所示的指数曲线.

但实际上, $N(t)$  是无法直接控制的,只能通过调整  $T(t)$  间接影响.因此,必须建立  $N(t)$  和  $T(t)$  的关系模型.由于  $T(t)$  与  $N(t)$  呈单调递增关系,即,1 帧中需要进行多模式决策的宏块越多,多模式决策的计算复杂度也会越高,所以本文使用一个简单的线性模型描述  $T(t)$  与  $N(t)$  的关系:

$$N(t) = k_t \times T(t) \quad (10)$$

其中, $k_t$  表示对时刻  $t$  的图像编码时,宏块多模式决策所消耗的平均计算复杂度.把公式(10)代入公式(9),可以得出当前时刻  $t$  应该设置的参数:

$$T(t) = \frac{N_{target}}{k_t} - \frac{(N_{target} - N_{t-1})}{k_t} \cdot e^{-1} \quad (11)$$

其中, $k_t$  只能在完成了整帧图像运动估计后算出.但调整  $N(t)$  时必须用到  $k_t$ , 因此,本算法使用前一帧图像的  $k_{t-1}$  代替  $k_t$ , 即

$$k_t \approx k_{t-1} = \frac{T(t-1)}{N(t-1)} \tag{12}$$

虽然用  $k_{t-1}$  代替  $k_t$  会使最终的复杂度计算产生误差,但由于本文的逼近模型已经考虑到这个问题,所以在下一时刻的计算中会自动对其修正.公式(11)对  $N(t)$  的计算分成两项:第 1 项  $N_{target}/k_t$  由线性模型公式(10)导出;第 2 项  $(N_{target}-N_{t-1})/k_t \cdot e^{-1}$  修正上一时刻所产生的误差  $N_{target}-N_{t-1}$ .在实际计算过程中,多模式决策计算复杂度  $N(t)$  的变化轨迹如图 8 所示,是以  $N_{target}$  为中心上下波动的.同时,下一节的实验结果也表明,公式(11)的方法能够很准确地控制运动估计的计算复杂度.

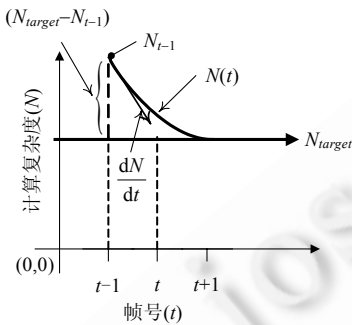


Fig.7 Theoretical curve of computational complexity  $N(t)$

图 7 理想的计算复杂度曲线  $N(t)$

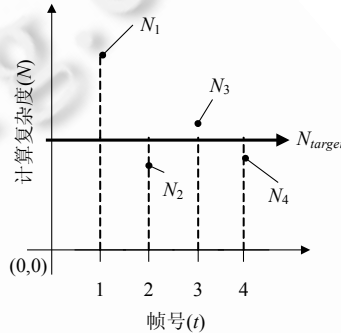


Fig.8 Actual curve of computational complexity  $N(t)$

图 8 实际的计算复杂度曲线  $N(t)$

### 3 实验结果

为了评估本算法的性能,本文配合联合模型(joint model 18.1, JM)对样本视频序列进行了测试.其中开启了高精度率失真优化选项,每个序列只对其中 200 帧编码,第 1 帧是 I 帧,其余都是 P 帧.因为篇幅有限,下面只列出量化参数分别是 16,20,24 和 28 的部分测试结果.实验在 Intel 酷睿双核 2.4 GHz 的台式电脑上完成.

实验结果表明,本文的算法能够在准确控制多模式决策计算复杂度的同时提供优化的率失真性能.本文使用平均实施多模式决策宏块数目作为多模式决策计算复杂度的量度,拉格朗日代价( $J$ )作为多模式决策后的率失真性能,PSNR 作为视频重建的质量,码率表示视频的编码效率.平均宏块数目和对应的  $J$  越低,说明算法越有效率.目标复杂度  $N_{target}$  表示当前帧需要进行多模式决策的宏块数目.由于在不同分辨率下视频的宏块数目会相差很大,其  $N_{target}$  难以统一比较,所以在实验部分,本文使用另一个控制参数  $K$  表示给定的计算约束. $K$  表示在 1 帧中应用多模式决策宏块的百分比.

图 9 给出算法对编码率失真性能的影响.从图中可以看出:随着  $K$  的增加,编码的率失真性能逐步上升.图 9(a)中的率失真曲线间隔明显比图 9(b)中的曲线间隔要大.这说明在高清序列中,增加多模式决策所能获得的率失真收益比在低分辨率序列中的要小.其原因是,采用小分块的运动补偿在低分辨率序列上比在高分辨序列上更有效.此外,无论是 CIF 序列还是高清序列,在高码率的情况下,增加不同计算复杂度所得到的收益也大于低码率的情况.

图 9 的结果同时也表示了本文方法用在 CIF 和高清序列上的率失真性能.当参数  $K$  设置成为 25%时,表示多模式决策只应用在 25%的宏块上.从图中可看出:这与只采用  $16 \times 16$  模式相比,信噪比有着大幅度的提高.此外,当参与多模式决策宏块的数目增加时,本算法的率失真性能将随之而升高.在  $K=75\%$  的情况下,本算法的信噪比与完全实施多模式决策相比仅相差小于 0.01 dB.这主要由于本算法根据 J-C slope 值从大到小分配计算资源,保证在花费最小计算资源的情况下获得最优的率失真性能.

通过上面的实验分析可知,本算法提供了可分级的计算复杂度和率失真性能.另外,不同复杂度可调整算法的比较结果在图 10 中给出.两种通用的方法参与比较:一种是基准方法(Baseline),就是通过增量地开启了所有



不同的模式(8×8,16×8,8×16,4×4,8×4,4×8,4×4);另一种是 Su 等人的方法<sup>[10]</sup>.在图 10 中,横坐标表示平均多模式决策的计算复杂度,纵坐标表示平均的拉格朗日代价.因为计算复杂度越高导致的预测失真度越低,所以全部的曲线都是单调递减的.从这些曲线可以看出,本文算法的下降速度明显比另外两种方法要快.也就是说,本算法在相同计算约束的条件下,总能获得比其他算法更低的预测失真度和更好的率失真性能.

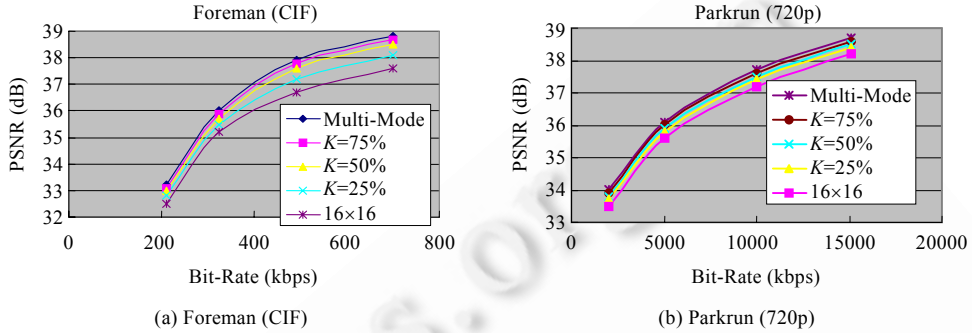


Fig.9 Rate-Distortion curves of the proposed algorithm

图 9 本文方法的率失真曲线

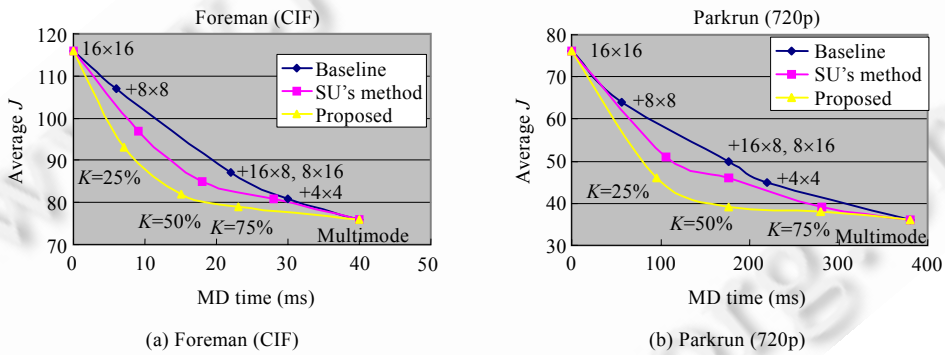


Fig.10 Complexity-Distortion curves of different CAMD algorithms

图 10 不同 CAMD 方法的复杂度-失真度曲线

此外,总体实验分别在表 1 所示的 22 个不同分辨率和不同运动情况的序列上完成.这些序列按照分辨率的不同被划分成 A~F 这 6 个组.

Table 1 Different groups of test sequences

表 1 测试序列的分组情况

Group	Format	Sequences
A	QCIF	News, Bus, Husky, Intros
B	SIF	Stenfan, Mobile, Garden, Tt, Tennis
C	CIF	Foreman, Sign_irene, Silent, Students
D	4SIF	Vtclnw, Galleon, Washdc
E	720p	Ducks_take_off, Parkrun, Old_town_cross
F	1080p	Park_joy, Riverbed, Pedestrian_area

表 2 是本算法作用在不同样本序列编码后耗费的计算复杂度和所达到的效果,K 表示本文方法所设置的目标计算复杂度参数.虽然目前 K 是一个固定值,但也可以对其动态地设置,以适应多变的实际应用需求.表 2 显示了本算法在 A~F 各组序列上采用不同 K 值的性能,ΔL 和 ΔTime 的计算公式如下:

$$\begin{cases} \Delta L = (J_{16 \times 16} - J_K) / (J_{16 \times 16} - J_{multi-mode}) \\ \Delta Time = (Time_{16 \times 16} - Time_K) / (Time_{16 \times 16} - Time_{multi-mode}) \end{cases} \quad (13)$$

其中,  $\Delta L$  是本算法与单一  $16 \times 16$  模式的平均拉格朗日代价之比,  $\Delta Time$  表示这两者的计算复杂度之比. 从表中数据可以看出:  $\Delta L$  和  $\Delta Time$  都会随着  $K$  的增大而变大,  $\Delta Time$  基本上正比于  $K$ . 然而,  $\Delta L$  在  $K$  值较低时会增长得更快. 这缘于本文按照 J-C slope 的大小顺序进行多模式决策,  $K$  值越小, 每增加单位  $K$  值所能得到的  $\Delta L$  就越大. 该特性也正是本文所期望的.

**Table 2** Rate-Distortion performance comparison of the proposed algorithm with different QPs

表 2 不同量化参数下, 算法率失真性能的比较

Sequence	K (%)	QP=16		QP=20		QP=24		QP=28	
		$\Delta L$ (%)	$\Delta Time$ (%)	$\Delta L$ (%)	$\Delta Time$ (%)	$\Delta L$ (%)	$\Delta Time$ (%)	$\Delta L$ (%)	$\Delta Time$ (%)
A	25	46	26	45	29	50	26	59	28
	50	77	51	81	51	81	52	84	53
	75	88	77	91	76	92	77	93	77
B	25	41	25	45	26	43	29	61	29
	50	69	53	82	45	77	54	83	54
	75	90	86	95	82	89	79	94	79
C	25	45	29	43	27	51	28	58	27
	50	79	48	79	51	84	57	82	51
	75	89	81	91	83	91	80	93	74
D	25	42	28	45	24	61	29	51	28
	50	81	48	81	52	84	55	81	50
	75	87	78	93	79	96	79	91	72
E	25	39	26	45	26	59	29	49	26
	50	76	52	79	54	81	53	83	53
	75	88	71	89	74	92	82	94	77
F	25	46	27	46	30	50	26	53	27
	50	77	53	78	49	74	52	75	51
	75	88	78	91	79	94	81	93	74

表 3 给出了不同多模式决策算法作用在不同视频序列上的平均计算性能. 为了方便比较, 基准算法 (Baseline) 的计算复杂度都统一标识为 100%, 其他算法的计算复杂度都是与基准算法的相对值. 从表 3 中可以看出, 本文算法的计算性能与另外两种算法 (Baseline 和 Su's method) 相比都具有明显的优势.

**Table 3** Percentage of time cost of different multi-mode decision algorithms (%)

表 3 不同多模式决策算法所耗费的时间比例 (%)

视频序列	基准算法	Su 等人的算法	本文方法
A	100	86	72
B	100	81	68
C	100	82	73
D	100	79	71
E	100	73	61
F	100	79	62

上面给出了不同算法在率失真性能上的比较, 本文的平均计算复杂度则明显低于 Su 等人所提出的算法. 下面我们按不同  $K$  值分别给出本算法的计算复杂度和率失真性能, 其中, 码率和视频重建质量的计算方式如下:

$$\begin{cases} \Delta PSNR = PSNR_K - PSNR_{16 \times 16} \\ \Delta Bit-rate = (Bit-rate_{16 \times 16} - Bit-rate_K) / Bit-rate_{16 \times 16} \end{cases} \quad (14)$$

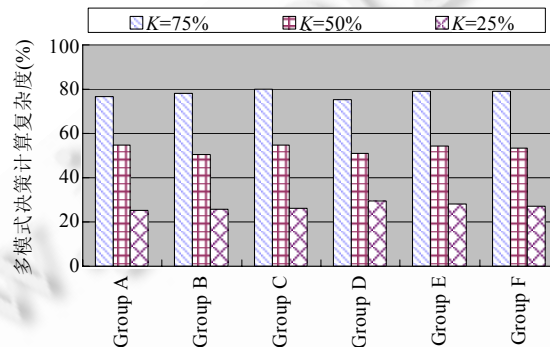
其中,  $PSNR_K$  和  $Bit-rate_K$  分别是当参数  $K$  被设定为不同数值时, 算法的信噪比和码率, 而  $PSNR_{16 \times 16}$  和  $Bit-rate_{16 \times 16}$  分别是仅采用  $16 \times 16$  单一分块模式时的信噪比和码率.

表 4 是本算法作用在不同样本序列后的信噪比和码率. 由于本文采用了固定量化参数的编码方式, 所以表中的信噪比相差并不远, 基本上都在 0.01 dB 以下. 而率失真性能的可伸缩性主要体现在码率上, 当  $K=25\%$  和  $K=50\%$  的时候, 基本上已经能够获取主要的码率收益. 这也恰恰表示了本算法按照率失真收益最大化的原则分配计算资源.

**Table 4** Rate-Distortion performance of the proposed algorithm with different  $K$ 表 4 不同  $K$  值下,本文算法的率失真性能

Sequences	Average $\Delta PSNR$ (dB)			Average $\Delta Bit-rate$ (%)		
	$K=25\%$	$K=50\%$	$K=75\%$	$K=25\%$	$K=50\%$	$K=75\%$
Group A	0.004	0.01	-0.007	5.5	6.3	6.7
Group B	0	0.001	-0.008	4.6	5.1	6.1
Group C	0.009	0.012	-0.02	6.4	7.3	7.5
Group D	0.01	0.009	0	7.2	8.5	8.9
Group E	0.009	0.011	-0.01	6.1	7.6	7.8
Group F	0.01	0.012	-0.01	5.9	6.1	7.0

从图 11 可知:不同的  $K$  值与不同的计算复杂度成等比例关系;随着  $K$  值的增大,算法的计算复杂度也相应地增加.这说明本文 J-C slope 阈值的自适应计算方法是十分准确和有效的.

Fig.11 Comparison of computational performance of the proposed algorithm with different  $K$ 图 11 不同  $K$  值下,本文算法计算性能的比较

#### 4 结束语

本文针对视频编码中的多模式决策过程,提出了一种可以准确控制计算复杂度的率失真优化算法.首先,本文给出了分别基于时域和空域相关性的 J-C slope 预测方法,继而在计算资源受限的情况下,按照预测 J-C slope 从大到小的顺序分配多模式决策的计算资源;其次,本文还给出了一种按照所给定的计算复杂度约束动态地调整 J-C slope 阈值的自适应方法,以准确地控制算法对计算资源的消耗.由于本文的方法仅在部分的宏块上实施多模式决策,所以该算法能够十分方便地与现有的快速多模式决策算法配合使用,进一步达到优化的效果.此外,本文所给出的自适应 J-C slope 阈值调整方法建立起任意计算复杂度度量参数与 J-C slope 阈值之间的关系模型,从而达到了准确控制计算复杂度的目的.根据本文的实验结果,本算法能够在相同计算约束下获得比其他复杂度可调整多模式决策方法更为优化的率失真性能.

#### References:

- [1] Lu J, Zhang P, Chao H, Fisher PS. On combining fractional-pixel interpolation and motion estimation: A cost-effective approach. IEEE Trans. on Circuits and Systems for Video Technology, 2011,21(6):717-728. [doi: 10.1109/TCSVT.2011.2129830]
- [2] Schwarz H, Marpe D, Wiegand T. Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans. on Circuits and Systems for Video Technology, 2007,17(9):1103-1120. [doi: 10.1109/TCSVT.2007.905532]
- [3] Sullivan GJ, Ohm J, Han WJ, Wiegand T. Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. on Circuits and Systems for Video Technology, 2012,22(12):1649-1668. [doi: 10.1109/TCSVT.2012.2221191]
- [4] Kim C, Kuo CCJ. Feature-Based intra-/intercoding mode selection for H.264/AVC. IEEE Trans. on Circuits and Systems for Video Technology, 2007,17(4):441-453. [doi: 10.1109/TCSVT.2006.888829]
- [5] Ri SH, Vatis Y, Ostermann J. Fast inter-mode decision in an H.264/AVC encoder using mode and Lagrangian cost correlation. IEEE Trans. on Circuits and Systems for Video Technology, 2009,19(2):302-306. [doi: 10.1109/TCSVT.2008.2009257]

- [6] Vatis Y, Lu L, Jagmohan A. Inter mode selection for H.264/AVC using time-efficient learning-theoretic algorithms. In: Bayoumi M, ed. Proc. of the 16th IEEE Int'l Conf. on Image Processing. New York: IEEE, 2009. 3413–3416. [doi: 10.1109/ICIP.2009.5413863]
- [7] Zeng H, Cai C, Ma KK. Fast mode decision for H.264/AVC based on macroblock motion activity. IEEE Trans. on Circuits and Systems for Video Technology, 2009,19(4):491–499. [doi: 10.1109/TCSVT.2009.2014014]
- [8] Lee J, Jun D, Park H. An efficient and fast mode decision method for inter slice of H.264/AVC. In: Bayoumi M, ed. Proc. of the 16th IEEE Int'l Conf. on Image Processing. New York: IEEE, 2009. 3405–3408. [doi: 10.1109/ICIP.2009.5413861]
- [9] Shen L, Liu Z, Zhang Z, Shi X. Fast inter mode decision using spatial property of motion field. IEEE Trans. on Multimedia, 2008, 10(6):1208–1214. [doi: 10.1109/TMM.2008.2001358]
- [10] Su L, Lu Y, Wu F, Li S, Gao W. Complexity-Constrained H.264 video encoding. IEEE Trans. on Circuits and Systems for Video Technology, 2009,19(4):477–490. [doi: 10.1109/ICIP.2007.4378895]
- [11] Akyol E, Mukherjee D, Liu Y. Complexity control for real-time video coding. In: Rodriguez J, ed. Proc. of the 14th IEEE Int'l Conf. on Image Processing. New York: IEEE, 2007. 77–80. [doi: 10.1109/ICIP.2007.4378895]
- [12] Luo J, Ahmad I, Liang Y, Swaminathan V. Motion estimation for content adaptive video compression. IEEE Trans. on Circuits and Systems for Video Technology, 2008,18(7):900–909. [doi: 10.1109/TCSVT.2008.923423]
- [13] Zhang XG, Guo ZM. Rate-Distortion optimized multi-path selection for video streaming over wireless multi-hop networks. Ruan Jian Xue Bao/Journal of Software, 2011,22(10):2412–2424 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3884.htm> [doi: 10.3724/SP.J.1001.2011.03884]
- [14] Zhou C, Zhang XG, Guo ZM. Optimization scheme for multi-user video transmission over MIMO multi-hop wireless networks. Ruan Jian Xue Bao/Journal of Software, 2013,24(2):279–294 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4201.htm> [doi: 10.3724/SP.J.1001.2013.04201]
- [15] Grecos C, Yang M. Fast inter mode prediction for P slices in the H264 video coding standard. IEEE Trans. on Broadcasting, 2005, 51(2):256–263. [doi: 10.1109/TBC.2005.846192]

#### 附中文参考文献:

- [13] 张行功,郭宗明.率失真优化的无线多跳网络多路径选择算法.软件学报,2011,22(10):2412–2424. <http://www.jos.org.cn/1000-9825/3884.htm> [doi: 10.3724/SP.J.1001.2011.03884]
- [14] 周超,张行功,郭宗明.面向MIMO多跳无线网络的多用户视频传输优化方法.软件学报,2013,24(2):279–294. <http://www.jos.org.cn/1000-9825/4201.htm> [doi: 10.3724/SP.J.1001.2013.04201]



陆寄远(1976—),男,广东广州人,博士,系统分析师,CCF 学生会员,主要研究领域为视频编码,图像处理.

E-mail: dtc005001@163com



刘宇熏(1971—),男,高级工程师,主要研究领域为多媒体网络通信,信息安全.

E-mail: wnzliuyx@gdudf.edu.cn



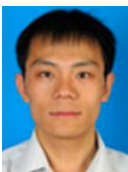
侯昉(1975—),男,博士,讲师,CCF 学生会员,主要研究领域为视频网络,图像通信.

E-mail: thf@gdudf.edu.cn



朝红阳(1957—),女,博士,教授,博士生导师,主要研究领域为视频编码,图像压缩.

E-mail: isschhy@sysu.edu.cn



黄承慧(1976—),男,博士,系统分析师,主要研究领域为视频信息检索,多媒体数据挖掘.

E-mail: hch.gdudf@163.com