

Journal of Image
and Graphics

中国图象图形学报



ISSN1006-8961
CN11-3758/TB

2012 8
Vol.17 No.

中国科学院遥感应用研究所
中国图象图形学学会主办
北京应用物理与计算数学研究所

中国图象图形学报

Zhongguo Tuxiang Tuxing Xuebao

2012 年 8 月 第 17 卷 第 8 期(总第 196 期)

目 次

综述

可计算图像美学研究进展 王伟凝, 蚁静斌, 贺前华(893)

图像处理和编码

利用混沌 PSO 或分解的 2 维 Tsallis 灰度熵阈值分割 吴一全, 吴诗嫒, 张晓杰(902)

H. 264 并行编码中负载平衡方法 侯兴松, 刘大齐, 盛凯, 顿玉洁(911)

提升细节捕捉能力的非下采样轮廓波变换 郭洪, 李雪军(919)

SVD 域的图像高斯噪声强度估计 柳薇(923)

图像扩散去噪模型的分析与改进 傅绪加, 刘峰, 王信松(934)

小波域最小嵌入失真函数设计及其在隐写中的应用 李军, 潘峰, 杨晓元, 刘镇(946)

图像分析和识别

抗压缩的照片图像与真实感计算机图形识别 张荣, 王让定(953)

云的自适应阈值检测方法 黄茜, 王少龙, 徐道磊, 廖婷婷(960)

煤矸区分中的间隔灰度压缩扩阶共生矩阵 于国防(966)

离群样本划分的半监督模糊学习策略 宋晓宁, 杨静宇, 杨习贝(971)

多相 Chan-Vese 模型的直接对偶方法 刘存良, 潘振宽, 郑永果, 王国栋, 丁洁玉(979)

利用梯度信息快速提取直线边缘特征 夏军营, 徐小泉, 熊九龙(987)

2 维双树复小波不确定度加权融合的人脸识别 王仕民,叶继华,邓涛,王明文(995)

图像理解和计算机视觉

流形学习与非线性回归结合的头部姿态估计 范进富,陈锻生(1002)

计算机图形学

角点距离矩阵和同心圆划分的曲线描述与匹配 曾接贤,刘秀朋,符祥(1011)

弹性二次曲面模型下的 3 维形状表现与平滑 郝世杰,蒋建国,郭艳蓉,詹曙(1021)

医学图像处理

主动学习的白细胞图像自动分割 崔凤,潘晨,吴向平,徐军(1029)

遥感图像处理

Bregman 交替迭代遥感图像复原方法 徐焕宇,孙权森,罗楠,夏德深(1035)

基于各向异性 Markov 随机场的遥感影像亚像元尺度建筑物提取 李晓冬,凌峰,杜耘(1042)

中国图象图形学报

刊名题字: 宋 健 月刊(1996 年创刊) 第 17 卷 第 8 期 2012 年 8 月 16 日出版

主管单位	中国科学院	Superintended by	Chinese Academy of Sciences
主 办	中国科学院遥感应应用研究所 中国图象图形学学会 北京应用物理与计算数学研究所	Sponsored by	Institute of Remote Sensing Application, CAS China Society of Image and Graphics Institute of Applied Physics and Computational Mathematics
主 编	李小文	Chief editor	LI Xiaowen
编辑出版	《中国图象图形学报》编辑出版委员会 北京 9718 信箱 邮编 100101 电子信箱:jig@irsa. ac. cn 电话:010-64807995 010-82614429 网 址:www. cjig. cn	Editor, Publisher	Editorial and Publishing Board of Journal of Image and Graphics (P. O. Box 9718, Beijing 100101, China) E-mail: jig@irsa. ac. cn
印刷装订	北京北林印刷厂	Distributed by	Beijing Bureau for Distribution of Newspapers and Journals
广告经营许可证	京朝工商广字第 0346 号	Domestic	All Local Post Offices in China
总 发 行	北京报刊发行局	Foreign	China International Book Trading Corporation (P. O. Box 399, Beijing 100044, China)
订 购	全国各地邮局	Printed by	Beijing Beilin Printing House
国外发行	中国国际图书贸易总公司 (中国国际书店) (北京 399 信箱 邮编 100044)		

ISSN 1006-8961 CN11-3758/TB CODE ZTTFXZ 国内邮发代号: 82-831 国外发行代号: M1406 国内定价: 45.00 元

Journal of Image and Graphics

(Monthly, Started in 1996)

Vol. 17 No. 8 August 2012

Contents

Review

Review for computational image aesthetics Wang Weining, Yi Jingjian, He Qianhua (893)

Image Processing and Coding

Two-dimensional Tsallis gray entropy image thresholding using chaotic particle swarm optimization or decomposition
..... Wu Yiquan, Wu Shihua, Zhang Xiaojie (902)

Load balancing for H. 264 parallel encoding Hou Xingsong, Liu Daqi, Sheng Kai, Dun Yujie (911)

Non-subsampled contourlet transform algorithm to promote detail information capturing ability Guo Hong, Li Xuejun (919)

Gaussian noise level estimation in SVD domain for images Liu Wei (923)

Analysis and improvement of image diffusion denoising models Fu Xujia, Liu Feng, Wang Xinsong (934)

Design of minimizing embedding impact function in DWT domain and its applications in steganography
..... Li Jun, Pan Feng, Yang Xiaoyuan, Liu Zhen (946)

Image Analysis and Recognition

Anti-compression approach to distinguishing photographic images and photorealistic computer graphics
..... Zhang Rong, Wang Rangding (953)

Adaptive threshold approach for night-sky cloud inspection Huang Qian, Wang Shaolong, Xu Dao lei, Liao Tingting (960)

Expanded order co-occurrence matrix to differentiate between coal and gangue based on interval grayscale compression
..... Yu Guofang (966)

Semi-supervised fuzzy learning strategy by using a way of partitioning the outlier instances
..... Song Xiaoning, Yang Jingyu, Yang Xibei (971)

Direct dual method of Chan-Vese model for multiphase segmentation
..... Liu Cunliang, Pan Zhenkuan, Zheng Yongguo, Wang Guodong, Ding Jieyu (979)

Fast line edge feature extraction using image gradients Xia Junying, Xu Xiaoquan, Xiong Jiulong (987)

Two-dimensional dual-tree complex wavelet transform uncertainty weighted fusion in face recognition
..... Wang Shimin, Ye Jihua, Deng Tao, Wang Mingwen (995)

Image Understanding and Computer Vision

Combining manifold learning and nonlinear regression for head pose estimation Fan Jinfu, Chen Duansheng (1002)

Computer Graphics

Representation and matching for planar curve based on corner distance matrix and concentric circles
..... Zeng Jiexian, Liu Xiupeng, Fu Xiang (1011)

Three-dimensional shape representation and smoothing with elastic quadratic patch model
..... Hao Shijie, Jiang Jianguo, Guo Yanrong, Zhan Shu (1021)

Medical Image Processing

White blood cell image segmentation based on active learning Cui Feng, Pan Chen, Wu Xiangping, Xu Jun (1029)

Remote Sensing Image Processing

Bregman alternating iterative method for remote sensing image restoration Xu Huanyu, Sun Quansen, Luo Nan, Xia Deshen (1035)

Building extraction at the sub-pixel scale from remotely sensed images based on anisotropic Markov random field
..... Li Xiaodong, Ling Feng, Du Yun (1042)

中图法分类号: TP301.6 文献标识码: A 文章编号: 1006-8961(2012)08-0911-08

论文引用格式: 侯兴松, 刘大齐, 盛凯, 顿玉洁. H. 264 并行编码中负载均衡方法[J]. 中国图象图形学报, 2012, 17(8): 911-918.

H. 264 并行编码中负载均衡方法

侯兴松, 刘大齐, 盛凯, 顿玉洁

西安交通大学信息与通信工程系, 西安 710049

摘要: 针对在多核处理器上 Slice 并行编码 H. 264 高清视频中的负载不平衡问题, 首先利用已编码帧的编码统计信息, 根据帧间时间相关性预测下一帧各宏块的编码负载, 然后据此预测的编码负载划分 Slice, 使各个处理器核上编码的 Slice 具有相接近的计算负载, 从而达到动态负载均衡目的。在 Tile64 多核平台上的实际测试结果表明, 与传统的基于宏块区域的动态数据分配算法相比, 该方法可以将编码并行加速比和并行效率提高 5% 左右。

关键词: H. 264; 并行编码; 多核处理器; 负载均衡

Load balancing for H. 264 parallel encoding

Hou Xingsong, Liu Daqi, Sheng Kai, Dun Yujie

Department of Information and Communication Engineering Xi'an JiaoTong University, Xi'an 710049, China

Abstract: To reduce the load imbalancing problem of the slice parallel of H. 264 encoding for high-definition videos with multicore processors, we firstly, predict the encoding load of each MB in the current frame based on the temporal dependence with the previous encoded frame statistics. Then the current frame is divided into slices with the predicted encoding load, so that each processor core has the relative similar computation load for each slice. Experiments using the Tile64 multicore processor platform show that it can increase the parallel encoding speedup and efficiency by about 5% with this method, when compared with the dynamic data partition algorithm based on macroblock regions.

Key words: H. 264; parallel encoding; multicore processor; load balance

0 引言

近年来,随着超高分辨率视频、3D 电视和实时高分辨率视频编码等新兴应用的出现,市场对于高计算性能的需求与日俱增。以往的办法是不断地提高时钟频率以及扩展相应的并行指令集,但是由于芯片热耗瓶颈以及开发并行指令集的难度越来越大,单纯使用高性能单核处理器已经远远不能满足需要。图1所示为 AMD Phenom 9900 (CPU 规格: 65 nm, 2600 MHz, 2 MB L2 Cache, 2 MB L3 Cache) 单核到 4 核处理器的 H. 264 编码器性能比较示意图。

图中所测的 4 个视频序列格式都为 HDTV (高清晰度电视, 1920 × 1080), 序列长度为 720 帧。图1时间单位为 min:s。

从图1中,可以明显看到多核编码性能要比单

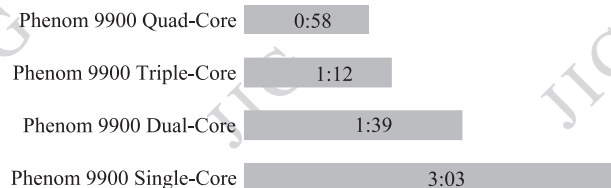


图1 不同核数下编码时间示意图

Fig. 1 The encoding time of different cores

收稿日期: 2011-05-19; 修回日期: 2012-04-17

基金项目: 国家自然科学基金项目(60602024); 西安交通大学校内科研基金项目(xjj2009042, xjj2012023)

第一作者简介: 侯兴松(1972—), 男, 副教授, 2005 年于西安交通大学电信学院信息与通信工程系获博士学位, 主要研究方向为图像和视频编码、图像的稀疏表示等。E-mail: houxs@mail.xjtu.edu.cn

核编码性能优越许多。4 核编码性能大约为单核的 3 倍左右。

多核处理器较之单核处理器,能带来更多的性能和计算优势,降低单个问题的求解时间,增加问题求解的规模。但是绝大多数的算法在设计时都是为了利用速度越来越快的处理器的处理能力,当处理器频率提升时,可以获得更高的性能,多核处理器核心数的增加,并不能使算法直接获得加倍的性能提升。因此在多核处理器智能化的并行编译优化和任务调度出现之前,对已有的算法必须首先进行任务的划分和调度从而获得并行能力。

在多核处理器上进行 H. 264 并行编码可以使用 GOP(group of pictures)、Frame(帧)、Slice(片)和 MB(宏块)等不同的数据结构进行编码任务的划分,但是 GOP、Frame 并行只能使用较小的并行规模,性能提升有限,MB 并行由于各宏块之间的数据参考关系,必须花费大量的时间进行核间数据的通信和同步,同样会限制并行性能的提高^[1-2]。相比较而言,由于 Slice 并行相对独立,具有大小适度的数据颗粒,并且具有比较好的可扩展性,因此 Slice 并行在实际中得到了广泛应用^[3-5]。文献[6]中提出了一种基于宏块自适应分配的 Slice 分割方案,该方法的基本思路是在没有对一帧进行编码之前先利用快速帧间模式选择的方法为该帧中的各个宏块选择合适的编码模式,然后利用它所提出的自适应 Slice 分割方法来均衡计算负载到每个处理器核上,但该

方法只是基于 JM9.6 来进行仿真,它并没有真正的在多核平台上实现,而且该方法过分强调了帧间模式选择的计算复杂度。文献[7]提出了一种基于宏块帧内预测的动态数据分配算法,该算法在 4 核 Intel Pentium 处理器上完成不同分辨率序列的编码任务,然而 Slice 并行编码在小规模的 2 个处理器核、4 个处理器核上应用时可以取的比较好的结果,但是当并行的核数大于 4 个时,会由于处理器核间计算负载分配的不均匀导致计算性能的下降。而且该文仅使用了帧内宏块预测来进行并行加速,并没有考虑到帧间相关性。针对上面所提到的这些问题,提出一种新的动态 Slice 任务负载均衡的方法来进行处理器核间的负载平衡,提高了并行性能。

1 Slice 并行编码负载平衡

1.1 Slice 并行编码

使用 X264 编码器在 Tile64 多核处理器上实现 Slice 并行编码。并行编码编码流程如图 2 所示,输入的帧数据首先进行 Slice 划分。划分的 N 个 Slice 对应于 N 个处理器核,由这 N 个处理器核同时编码相应的 Slice,当使用其他 Slice 的数据时,使用核间通信交换数据,编码完成后进行同步操作合并所有 Slice 码流,写入 Nal(network abstraction layer)单元。然后编码下一帧,直到所有帧编码完成后结束。

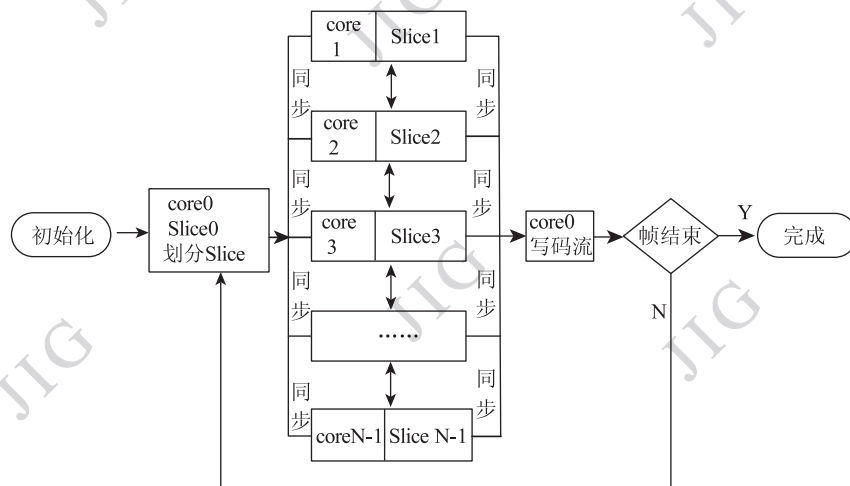


图 2 Slice 并行编码流程图

Fig. 2 Slice parallel encoding flow chart

衡量并行算法的标准有并行加速比和并行效率; S_p 为并行加速比,用来衡量并行算法的提升性

能; E_p 为加速效率,用来衡量执行并行程序时,每个参与并行运算的处理器的使用效率。

$$S_p = \frac{T_1}{T_p} \quad (1)$$

$$E_p = \frac{S_p}{P} \quad (2)$$

式中, P 为处理器核数, T_1 为串行算法在单处理机上的运行时间; T_p 为并行算法在 P 个处理器核上的运行时间。图 3 和图 4 给出了使用不同处理器核 Slice 并行编码的并行性能, 可以看到编码并行加速比虽然随处理器核数的增加而增加, 但并行效率随着核数的增多不断下降, 8 个核时达到的并行效率在 80% 左右, 32 个核时并行效率在 80% 以下。因此有必要分析造成并行效率下降的原因, 提高多处理器核的利用效率。

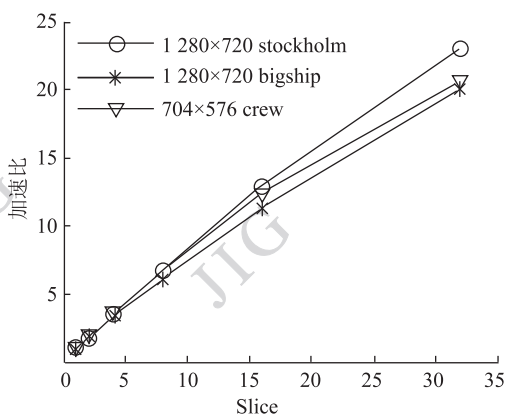


图3 Slice 并行加速比

Fig. 3 Slice parallel speedup ratio

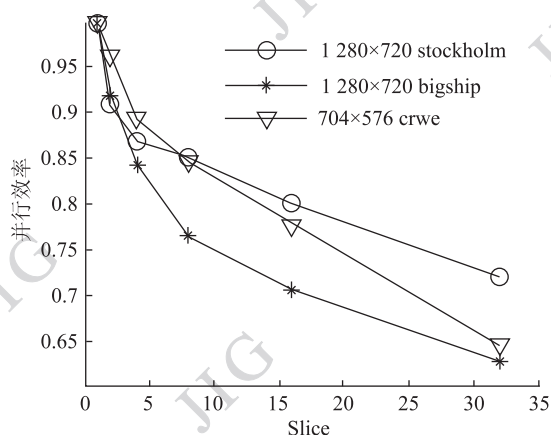


图4 Slice 并行效率

Fig. 4 Slice parallel efficiency

1.2 并行编码中的负载平衡

在 Slice 并行编码中通过平均划分每个 Slice 包含的宏块数目, 实现了简单的静态负载平衡。

但是对于实际的视频序列来说, 静态负载平衡往往不能很好地均等划分各 Slice 的负载。如图 5 所示的视频序列, 对图像背景进行预测和编码显然要比对中间的人物前景进行预测和编码运算量小得多。因此静态平均划分宏块的方法虽然分配给每个 Slice 的宏块数目相同, 但由于不同的 Slice 区域编码复杂度并不相同, 实际上并没有平均划分各 Slice 的计算任务量, 这就造成了各 Slice 编码时间的不同, 图 6 所示为 football 序列划分 4 个 Slice 时各处理器核编码对应的 Slice 的时间统计。统计表明, 各 Slice 编码的最大时间和最小时间相差有 25% 左右。因此各 Slice 编码计算负载之间的不平衡造成了并行编码效率的下降。对于并行系统来说, 各处理器核上的负载分配状况决定了整个系统可以取得的并行加速比。对于 Slice 并行的编码任务来说, 总的编码负载可以由各 Slice 的编码负载相加得到, 即



图5 claire 序列中的一帧图像

Fig. 5 One frame in claire sequence

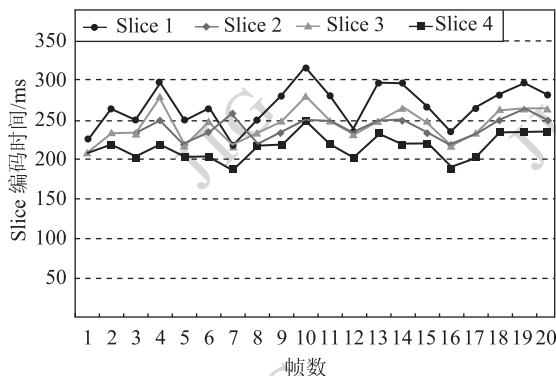


图6 静态数据分割各 Slice 编码时间

Fig. 6 Slice encoding time for static data partition

$$C_f = \sum_{i=0}^{N-1} C_i \quad (3)$$

式中, N 为多核的核数, C_f 为整帧的编码负载; C_i 为第 i 个 Slice 的编码负载计算量, 可以由编码时间衡量。多核执行相对于单核执行可以获得的并行加速比为

$$S_p = \frac{C_f}{\text{Max}\{C_i\}} \quad (4)$$

为了达到并行加速比的最大化和处理器资源的最大利用, 必须使 $\text{Max}\{C_i\}$ 最小, 即各个 Slice 拥有相对较接近的编码负载。

$$C_i \approx C_j, \forall i, j \in \{0, 1, \dots, N\} \quad (5)$$

由于编码过程中各宏块之间运算量大不相同, 所以简单的静态划分不能对总的计算复杂度有一个很好的平均作用。针对这一问题, 提出在不改变 X264 宏块预测提前终止策略的前提下, 在一帧计算过程中动态统计和预估下一帧的编码计算量, 作为平均分配核间计算复杂度的依据, 从而近似地达到动态负载平衡。

2 Slice 动态负载分配

2.1 编码宏块复杂度分析及统计

编码复杂度不平衡主要体现在宏块复杂度的差别上。宏块在预测过程中, 有的使用大块 16×16 分割, 有的使用小块 4×4 分割, 这就造成了预测和编码计算量的差异。因此有必要使用宏块来作为复杂度估计和 Slice 负载分配的依据。对多个序列 (cif football、cif forman、4 cif crew、720p bigship、720p stockholm) 的各模块编码时间统计表明, 在编码参数相同的情况下, 各模块的计算时间占总计算时间的比例大致相同, 如表 1 所示。可以看出宏块分析和预测模块占用了计算的绝大多数时间。

宏块分析和预测模块是计算的主要部分。其主要任务是找到帧间模式的最优编码代价, 并保存最优编码模式的分块方式和预测运动矢量。对于帧间宏块可以采用的宏块编码方式有 Skip、 16×16 、 16×8 、 8×16 、 8×8 、 8×4 、 4×8 和 4×4 共 8 种模式, 帧间预测后还要和帧内预测比较, 因此每个宏块都要对 9 种编码模式进行编码代价最优化的比较。

表 2 是宏块分析预测中各模块所需计算周期统计 (测试序列和表 1 相同)。

表 1 Slice 层编码各模块所需时间统计

Table 1 Load statistics for each module in slice coding

模块名称	占总计算比例/%
X264_macroblock_analyse	60.60
X264_macroblock_write_cabac	18.60
X264_fdec_filter	8.70
X264_macroblock_encode	8.70
X264_macroblock_cache_load	1.80
X264_macroblock_cache_save	1.30
X264_cabac_mb_skip	0.20
X264_cabac_encode_terminal	0.10

表 2 宏块预测分析各模块计算负载统计

Table 2 Load statistics for each module in MB analysis

宏块	Mode_time /周期	$\frac{\text{Mode_time}}{\sum \text{Mode_time}} / \%$
Skip	1 367	1.96
16×16	7 312	10.50
8×16	6 874	9.87
16×8	7 515	10.79
8×8	8 266	11.87
4×4	12 800	18.37
4×8	8 000	11.48
8×4	8 800	12.63
Intra	8 733	12.53

2.2 X264 宏块预测提前终止策略

在对 P 帧作宏块预测时需要比较所有的 9 个编码模式的最小代价。但是实际应用中很少有算法把这 9 种编码模式全部的编码代价得到后才选择编码代价最小的最优模式。X264 对帧间模式的预测使用了提前终止的策略降低编码复杂度。图 7 所示为帧间宏块预测编码模式选择流程图。

如果当前宏块的左方、上方、左上方、右上方中有一个是 Skip 模式, 则当前宏块也有可能是 Skip 模式, 采用 Skip 模式经过变换、量化、zigzag 重排序后的 DCT (离散余弦变换) 系数分布足够稀疏, 则当前宏块直接采用 Skip 模式而不再计算其他的 8 种预测方式, 此时 Test1 成立。如果 Test1 不成立, 继续计算帧间 16×16 和 8×8 分割模式的编码代价, 如果 4 个 8×8 子块的残差编码代价和大于 16×16 的

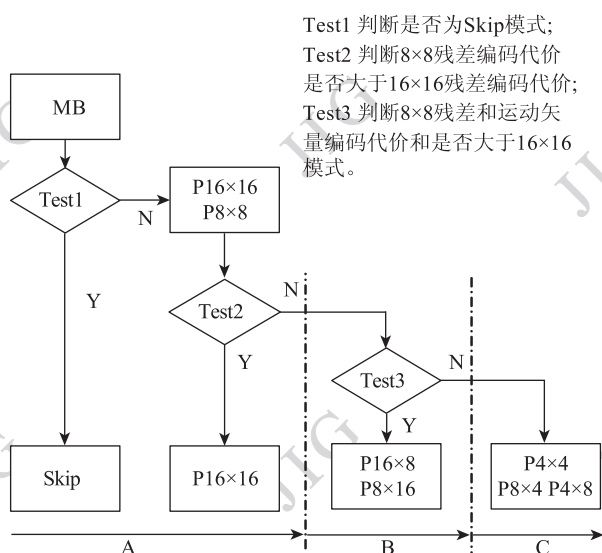


图7 帧间宏块预测编码模式选择流程图

Fig. 7 MB inter mode selection flow

残差编码代价 (Test2) 成立, 则当前宏块不再计算 16×8 、 8×16 和 8×8 子宏块模式, 选择 16×16 作为最优预测模式。Test2 不成立的话, 继续比较 4 个 8×8 分块模式的残差和运动矢量编码代价是否大于 16×16 模式的编码代价 (Test3), 如果 Test3 成立则只需再计算 16×8 、 8×16 的编码代价和 16×16 的编码代价比较得到最优编码代价。如果 Test3 不成立, 则要计算所有的 8×8 子块划分模式、 8×8 、 4×8 、 8×4 和 4×4 , 和前面计算的 16×16 、 16×8 、 8×16 编码代价比较得到最优预测模式。

从宏块预测模式选择的流程图上可以看出不同模式的最终选择, 其计算量是不同的, 如果选择了 Skip 模式, 其计算只经过了 Skip 模式预测这一步; 选择 16×16 模式意味着计算过程包括了 Skip、 16×16 和 8×8 这 3 种模式预测, 需要经过 A 的计算过程; 选择 16×8 和 8×16 意味着计算过程包括了 Skip、 16×16 、 8×8 、 16×8 和 8×16 这 5 种模式预测, 需要经过 A + B 的计算过程; 如果选择了 8×8 子宏块模式, 那么其前面肯定已经对所有的帧间模式 Skip、 16×16 、 8×8 、 16×8 、 8×16 、 4×8 、 8×4 和 4×4 共 8 种模式都进行了计算, 需要经过 A + B + C 的计算过程。预测模式计算复杂度从大到小依次为 8×8 子宏块模式、 16×8 或 8×16 模式、 16×16 模式、Skip 模式。

2.3 编码宏块预测模式的选择

宏块采用不同的预测模式, 计算量有很大的不同, 这是在宏块级造成负载不均衡性的主要原因。针对帧间模式选择的研究表明, 已编码帧对后续帧

宏块编码模式具有非常明显的时间相关性, 可以采用快速帧间模式选择缩小待编码宏块的预测模式范围^[8-10]。相邻 P 帧之间对应宏块预测模式的时间相关性统计表明^[11], 如果前一帧宏块使用了 Skip 模式, 则下一帧对应宏块使用 Skip 模式或 16×16 的概率大于 80%; 如果前一帧宏块使用了大分块模式, 则下一帧对应宏块使用大块分割的概率和也超过了 75%。

在 X264 的基础上, 不改变 X264 宏块预测提前终止策略, 引入时间相关性的思想, 由前一帧编码宏块模式估计下一帧宏块预测模式, 将估计的结果作为计算复杂度的依据。

根据编码模式时间相关性和图 7 帧间宏块预测编码模式选择流程, 将前一帧编码模式和后续的估计宏块预测模式划分为对应的 3 组 (表 3)。前一帧的 3 组已编码模式为 G1 {Skip}、G2 { 16×16 , 16×8 , 8×16 } 和 G3 {sub 8×8 } ; 对应的 3 组预估宏块预测模式 g1 {Skip, 16×16 , 16×8 }、g2 {Skip, 16×16 , 16×8 , 16×8 , 8×16 , 8×16 } 和 g3 {Skip, 16×16 , 16×8 , 8×16 , sub 8×8 } , sub 8×8 包括了 8×8 、 4×8 、 8×4 和 4×4 模式。

表3 f-1 帧宏块模式和 f 帧宏块模式对应表

Table 3 MB mode in frame f corresponding to MB mode in frame f-1

MB_{ij}^{f-1} _mode	MB_{ij}^f _mode
G1 {Skip}	g1 {Skip, 16×16 , 16×8 }
G2 { 16×16 , 16×8 , 8×16 }	g2 {Skip, 16×16 , 16×8 , 16×8 , 8×16 , 8×16 }
G3 {sub 8×8 }	g3 {Skip, 16×16 , 16×8 , 8×16 , sub 8×8 }

帧间模式预测完之后, 还要预测帧内模式的编码代价, 在帧间最优模式和帧内最优模式间确定最终预测模式, 所以对于 f 帧宏块来说, g1、g2、g3 计算完成后还要加上帧内模式计算才是 f 帧宏块预测的负载。由 Slice 层编码各模块所需时间统计可以得出 g1、g2、g3 和 Intra 模式的计算负载, 如表 4 所示。

表 4 是只计算宏块预测得到的复杂度, 由表 2 知宏块预测部分占用了 60% 左右, 则一个宏块的总编码复杂度可由 MB_{ij}^f _load = MB_{ij}^f _load' $\times$ 0.6 + 0.4 得到, 式中 MB_{ij}^f _load' 为预测的宏块分析模块代价。如表 5 所示为 f-1 帧宏块模式和 f 帧宏块负载对应表。

表 4 3 组估计模式宏块预测复杂度

Table 4 Predicted MB mode encoding load for 3 groups

MB _{ij} ^f _mode	MB _{ij} ^f _load'
g1 { Skip, P16 × 16, P8 × 8 }, Intra	0.369
g2 { Skip, P16 × 16, P16 × 8, P8 × 16, P8 × 8 }, Intra	0.575
g3 { Skip, P16 × 16, P16 × 8, P8 × 16, sub8 × 8 }, Intra	1

表 5 f-1 帧宏块模式和 f 帧宏块负载对应表

Table 5 MB load in frame f corresponding to MB mode in frame f-1

MB _{ij} ^{f-1} _mode	MB _{ij} ^f _load
G1 { Skip }	0.62
G2 { P16 × 16, P8 × 16, P16 × 8 }	0.745
G3 { sub8 × 8 }	1

2.4 本文 Slice 动态负载均衡方法

由上面的分析可知,有了 f 帧所有宏块的编码复杂度估计,就可以计算 f 帧所有宏块的预估负载总和

$$Fload = \sum MBload_{ij} \quad (6)$$

式中, $Fload$ 为 f 帧宏块预估负载和, $MBload_{ij}$ 为第 i 个 Slice 的第 j 个宏块的计算负载预估。

有了一帧的计算负载,就可以将总的计算负载在 Slice 之间平均分配,划分新的 Slice 边界:

$$Sload_i = \frac{Fload}{N} \quad (7)$$

式中, $Sload_i$ 为第 i 个 Slice 分配到的负载, N 为划分的 Slice 数目。

由每个 Slice 分配的负载 $Sload_i$ 和所有的宏块负载 $MBload_{ij}$, 可以计算得到 Slice 划分边界, 并进而划分 f 帧 Slice 边界。

3 实验结果

对 4 组运动场景复杂度均不相同的序列在 Tile64 多核处理器平台上进行 Slice 并行编码, 其中一组使用静态负载均衡, 另一组使用动态负载均衡, 并且对在不同的核数情况下各个方法的加速比及并行效率都进行了相应的统计。其次将文献[7]中的动态负载均衡算法在 Tile64 多核处理器平台上做了相应的实现, 并与本文算法进行了对比。实验测试参数如下: 测试序列帧中只有第一帧为 I 帧, 其余帧均

为 P 帧, 即帧结构为 IPPP; 熵编码选择 CABAC (context-based adaptive binary arithmetic coding); 参考帧数为 1; 不采用码率控制模式, 量化步长 qp 设为 28。

采用 4 组运动场景复杂度不同的序列进行测试, 其中运动场景变化较为平缓的序列 bigship 和序列 shileds 可以达到较高的并行加速比。与之相反的运动场景变化较为剧烈的序列 stockholm 和序列 parkjoy 的并行加速比则相应较低。运动场景变化剧烈意味着视频序列中前后两帧之间的相关性会变差, 即由前一帧预测后一帧的难度将加大, 所以运动场景变换剧烈的序列其并行加速比性能较差。

表 6—表 9 是 4 组序列使用静态负载均衡和宏块负载动态分配的编码性能比较, 使用宏块负载动态分配后, 其并行性能较静态分配最高可以提高 16.4%。图 8—图 11 是理想的完全负载均衡, 本文所提宏块负载动态分配方法, 静态负载均衡和文献[7]方法的加速比性能比较图。

表 6 720p bigship 序列使用动态负载分配并行性能

Table 6 Parallel performance for dynamic load assignment of 720p bigship sequence

不同负载均衡 方法编码帧率	Slice 数目					
	1	2	4	8	16	32
静态负载/(帧/s)	1.68	3.10	5.67	10.29	19.02	33.78
动态分配/(帧/s)	1.68	3.27	6.29	11.98	21.4	37.2
提高百分比/%	0	5.5	10.9	16.4	12.5	10.1

表 7 720p stockholm 序列动态负载分配并行性能

Table 7 Parallel performance for dynamic load assignment of 720p stockholm sequence

不同负载均衡 方法编码帧率	Slice 数目					
	1	2	4	8	16	32
静态负载/(帧/s)	1.55	2.82	5.38	10.56	19.86	35.72
动态负载/(帧/s)	1.55	2.96	5.76	11.29	21.3	38.8
提高百分比/%	0	5.0	7.1	7.0	7.3	8.6

表 8 720p parkjoy 序列动态负载分配并行性能

Table 8 Parallel performance for dynamic load assignment of 720p parkjoy sequence

不同负载均衡 方法编码帧率	Slice 数目					
	1	2	4	8	16	32
静态负载/(帧/s)	1.61	2.98	5.52	10.38	19.35	34.43
动态负载/(帧/s)	1.61	3.16	5.93	11.29	20.95	37.21
提高百分比/%	0	6.0	7.4	8.7	8.3	8.1

表9 720p shileds 序列动态负载分配并行性能
Table 9 Parallel performance for dynamic load assignment of 720p shileds sequence

不同负载平衡 方法编码帧率	Slice 数目					
	1	2	4	8	16	32
静态负载/(帧/s)	1.58	2.86	5.27	10.41	20.01	35.64
动态负载/(帧/s)	1.58	2.98	5.71	11.18	21.52	39.13
提高百分比/%	0	4.2	8.3	7.4	7.6	9.5

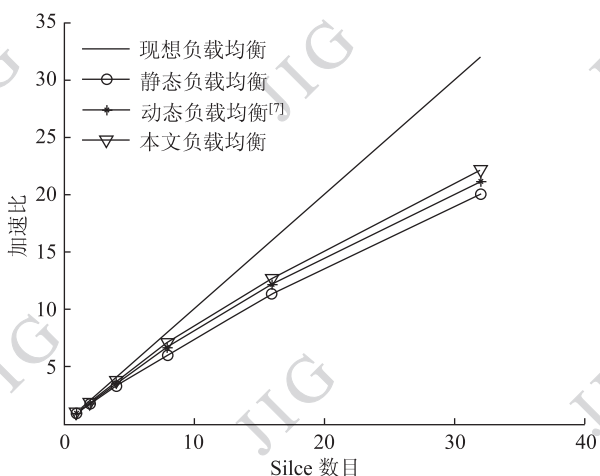


图8 720p bigship 序列动态负载分配并行性能
Fig. 8 Parallel performance for dynamic load assignment of 720p bigship sequence

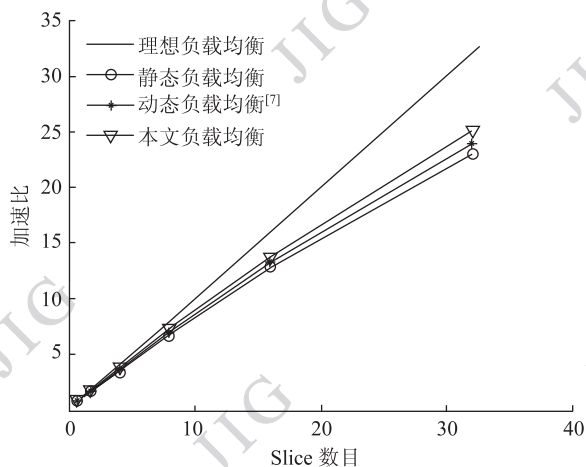


图9 720p stockholm 序列使用动态负载分配并行性能
Fig. 9 Parallel performance for dynamic load assignment of 720p stockholm sequence

从以上实验结果图中可看到加速比随着核数的增加类似于线性增加,而并行效率在不断下降。在加速比性能示意图中,对角线代表就是理想的负载

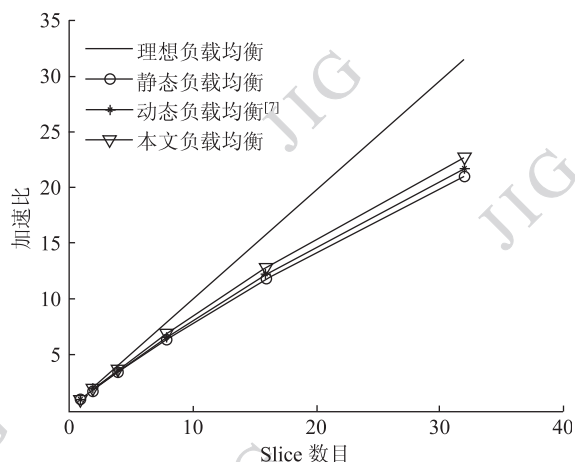


图10 720p parkjoy 序列动态负载分配并行性能
Fig. 10 Parallel performance for dynamic load assignment of 720p parkjoy sequence

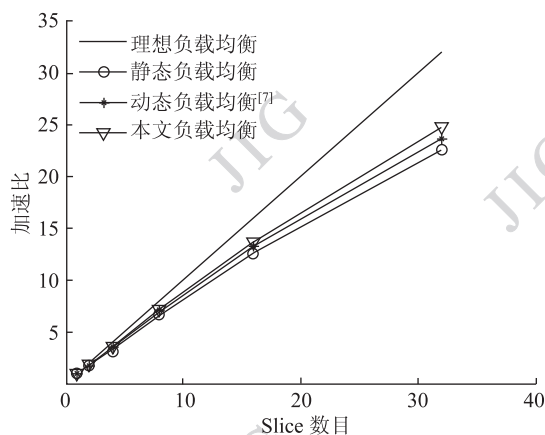


图11 720p shileds 序列动态负载分配并行性能
Fig. 11 Parallel performance for dynamic load assignment of 720p shileds sequence

均衡下的加速比的情况。可以看出在核数较少时,动态负载平衡曲线十分接近对角线,随着核数的不断增加,加速比曲线不断偏离对角线,这也就是引言当中提到在核数较少情况下(4核以下)并行计算性能较高,而核数越往上走性能提升越难的缘故。究其原因在于随着核数增加,核间通信以及数据交换的负荷越来越大,处理器核间计算负载分配的越来越不容易均匀分配。

再者,从各个实验结果图中可看出本文所提的动态负载均衡算法所得到加速比曲线要比静态算法及文献[7]算法的加速比曲线更接近对角线,这说明了本文所提算法通过在编码过程中对下一待编码帧的运算复杂度的估计及分配,使得分配到各个Slice上的计算工作量较静态分配及文献[7]算法更

为接近理想的负载均衡情况(与静态分配方法相比,所提方法可以将编码并行加速比和并行效率提高 10% 左右;与传统的基于宏块区域的动态数据分配算法相比,所提方法可以将编码并行加速比和并行效率提高 5% 左右),从而性能上有了可观的提高。

4 结 论

由于在基于 Slice 级的多核系统并行编码过程中各个 Slice 之间负载的不均衡性,造成了 Slice 并行编码的性能指标并行加速比和并行效率的降低。本文通过预先估计待编码帧总的宏块编码计算复杂度,即预先估计 Slice 编码计算负载度的方法,将计算负载动态均衡分配。测试结果表明,这种方法相比于固定宏块数目的静态负载平衡,并行加速比和并行效率都有了一定的提高。

参考文献 (References)

- [1] Luo C Y, Sun J G, Tao Z Y. The research of H. 264/AVC video encoding parallel algorithm [J]. 2008 Second International Symposium on Intelligent Information Technology Application, 2008, 10(1): 201-205.
- [2] Rodriguez A, Gonzalez A, Malumbres M P. Hierarchical parallelization of an H. 264/AVC video encoder[C]//Parelec'06 Proceedings of the International Symposium on Parallel Computing in Electrical Engineering. Washington, DC, USA: IEEE Computer Society, 2006: 363-368.
- [3] Jacobs T R, Choularas V A, Mulvaney D J. Thread-parallel MPEG-2, MPEG-4 and H. 264 video encoders for SoC multi-processor architectures [J]. Consumer Electronics, IEEE Transactions on, 2006, 52(1): 269-275.
- [4] Zhao Z, Liang P. A highly efficient parallel algorithm for h. 264 video encoder [C]//Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. Toulouse, France: IEEE Signal Processing Society, 2006: 489-492.
- [5] Jonghan P, Soonhoi H. Performance analysis of parallel execution of H. 264 encoder on the cell processor [C]// Embedded Systems for Real-Time Multimedia. Salzburg, Austria: CODES-ISSS, 2007: 27-32.
- [6] Jung B, Jeon B. Adaptive slice-level parallelism for H. 264/AVC encoding using premacroblock mode selection [J]. Journal of Visual Communication and Image Representation, 2008, 19(8): 558-572.
- [7] Juntae K, Jaeyoung P, Kyoungkun L, et al. Dynamic data partition algorithm for a parallel H. 264 encoder [J]. World Academy of Science, Engineering and Technology, 2010, 72(68): 350-353.
- [8] Wu D, Pan F, Lim KP, et al. Fast intermode decision in H. 264/AVC video coding [J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2005, 15(7): 953-958.
- [9] Hanli W, Sam K, Chi-Wah K. An efficient mode decision algorithm for H. 264/AVC encoding optimization [J]. Multimedia, IEEE Transactions on, 2007, 9(4): 882-888.
- [10] Jianfeng R, Kehtarnavaz N, Budagavi M. Computationally efficient mode selection in H. 264/AVC video coding [J]. Consumer Electronics, IEEE Transactions on, 2008, 54(2): 877-886.
- [11] Byung-Gyu K. Novel inter-mode decision algorithm based on macroblock (MB) tracking for the p-slice in H. 264/AVC video coding [J]. Circuits and Systems for Video Technology, IEEE Transactions on, 2008, 18(2): 273-279.