

H. 264 运动估计中块模式选择的并行设计

廖永红

(广东轻工职业技术学院计算机工程系 广州 746880)

摘要 为了方便 H. 264/AVC 在低码率高时性应用系统中的实现,须对其编码算法进行优化。本文首先对 H. 264/AVC 编码器进行了深入的分析,然后在此基础上提出了块模式选择的并行设计方案,并对此方案进行了并行设计与实现。最后对此设计方案在多核服务器上进行了试验。试验结果表明:并行后的三步搜索算法在不影响 H. 264 编码质量的前提下,其编码速度平均提高了 2.73 倍以上。

关键词 H. 264/AVC, 运动估计, DCT, 块模式选择, 并行算法

Parallel Design and Implementation of Block Mode Selection in Motion Estimation of H. 264

LIAO Yong-hong

(Department of Computer Engineering, Guangdong Industry Technical College, Guangdong 746880, China)

Abstract To facilitate H. 264/AVC implement in the application system with high low rate, its coding algorithm needs to be optimized. This paper first analyzes the H. 264/AVC encoder deeply, then on the basis of it presents the parallel design program of block mode selection and parallel designs and implements this program. Finally the design program is tested on multi-core server and the testing results show that the encoding speed of the three-step search algorithm after parallel improves by an average of 2.73 times without affecting the quality of H. 264 coding.

Keywords H. 264/AVC, Motion estimation, DCT, Mode select, Parallel algorithms

1 引言

H. 264/AVC 是 ITU-T 和 MPEG 组联合专家组 JVT 制定的视频编码标准,这一编码标准可获得很高的编码效率,尤其是在低码率方面比 MPEG-4 有明显提高,适合低宽带、高质量网络视频应用的需要。但是 H. 264/AVC 为了提高编码效率,采用了许多高计算复杂度的算法,使得编解码计算量很大,成为制约 H. 264/AVC 编码标准应用的一个主要瓶颈。

通过对 H. 264/AVC 编码器各个算法模块的分析可以得出:运动估计模块的计算量占了整个编码器运算量的 70% 以上。在 H. 264/AVC 编码标准中,引入了 7 种帧间候选模式。虽然这种多模式方法能够在一定程度上提高块的匹配效率,但是判断所有模式的算法却极大地增加了计算的复杂度。因此,对块模式选择的优化,对提高 H. 264/AVC 编码器的执行速度有着至关重要的作用。

2 运动估计的块模式选择

2.1 运动估计的基本原理

多数情况下,只是运动图像中的很少一部分图像在运动,同一场景相邻的两幅视频帧之间在内容上的差异不会太大,或者说后一帧的内容与前一帧的重复部分很多,前后两帧存在很强的时间相关性。运动估计的基本思想(如图 1 所示)是将图像序列的每一帧分成许多互不重叠的宏块,并认为宏块内所有像素的位移量都相同,然后对于当前帧中的每一块到前一帧或后一帧某一给定搜索范围内根据一定的匹配准则找出与当前块最相似的块,即匹配块。由匹配块与当前块的相

对位置计算出运动位移,所得即为当前块的运动矢量。利用搜索得到的运动矢量在参考帧上进行运动补偿,残差值(difference)经 DCT 变化、量化、行程编码后与运动矢量共同经熵编码,然后以比特流形式传出去。

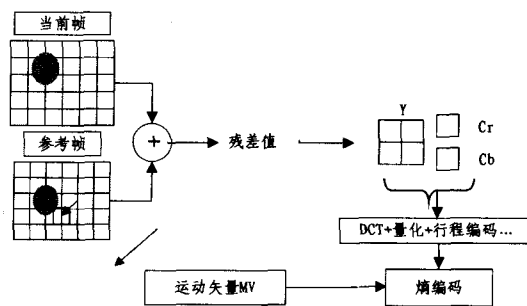


图 1 运动估计的基本思想

显然,残差块的值越小,越有利于压缩。因此,运动估计的主要目标就是使预测块和当前块之间的残差能量值尽量小。搜索窗尺寸(搜索范围)的选取对搜索结果有很大影响:搜索范围越大,得到更小残差块的可能性越大,但这会带来更大的时间开销。通常选择的搜索范围有 $\pm 7, \pm 16, \pm 32$ 等等。由于块匹配时需要在前后帧间逐像素比较,如果采用全搜索算法(即穷尽参考帧搜索区域所有可能的点进行比较,从中找到全局最优块),计算量非常大。

2.2 块的模式选择

块匹配法基于如下假设:同一匹配块的像素运动是一致的。显然这个假设具有一定的片面性,但选择合适的块形状与大小可在一定程度上消除这种片面性。一般说来,块形状

选用正方形是比较自然的选择,这样既便于图像的划分,又有利于块匹配准则函数的计算。但这并非总是最佳选择,因此有的算法采用了其他形状,如三角形,针对视频对象的边缘形状特征,将宏块按不同形状的模板进行分割。

关于块的大小,如果块越小,得到的残差值越小,但是这会引入较多的运动矢量,可能降低编码的效率。在 H. 261, MPEG-1 和 MPEG-2 中,运动矢量是以 16×16 宏块为单位的。由于块尺寸相对较大,可能包含图像中不同的运动部分,造成预测精度的下降;H. 263 和 MPEG-4 标准在宏块运动矢量的基础上加入了以 8×8 块的运动矢量,预测精度得到了一定的提高。而在 H. 264/AVC 编码标准中,在标准的参考代码中,有 7 个帧间模式可供选择,即 $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 4$, 如图 2 所示。一个宏块可以采用上述 7 种模式中的任意一种模式或是多种模式的组合。

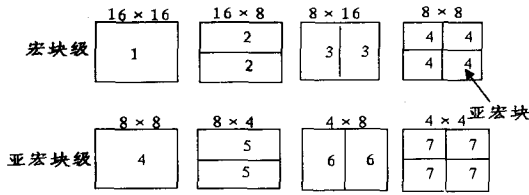


图 2 H. 264/AVC 中块的模式选择

H. 264/AVC 在进行运动估计时,首选对当前块采用所有的模式进行运动估计,然后用率-失真优化(RDO)比较在每种模式下码率的大小,码率小的模式将作为最终的模式。该过程是一个全搜索的过程,它盲目地搜索了所有候选模式,从而造成了极大的计算复杂度。虽然通过这种方法能够得到最优的效果,但是在很多硬件设备中不能支持这种庞大的计算能力要求。因此,在尽可能小的质量损失前提下,过滤一些候选的模式,是对 H. 264/AVC 进行优化的关键。

2.3 块匹配准则

运动估计的准确性依赖于在块匹配过程中所用的匹配准则。其中最主要的是两个相匹配的像素块之间的误差度量(Block Distortion Measure, BDM)。从文献中看,共有三种基于块的误差度量。这里假设物体的帧间运动局限在查找窗口($2w \times 2w$)之内。下面给出这些 BDM 的定义以及相应的匹配准则。其中, m 和 n 是计算 BDM 的像素块的尺度(分别对应于宽度和高度), $U(k, l)$ 是上一帧中的物体块, $U_R(k+p, l+q)$ 是当前帧的匹配块, (p, q) 就是物体的帧间运动矢量(位移向量)。

(1) 归一化互相关函数(Cross-correlation function, CCF)

$$M_1(p, q) = \frac{\sum_{k=l=1}^m \sum_{l=1}^n U(k, l) U_R(k+p, l+q)}{(\sum_{k=l=1}^m \sum_{l=1}^n U^2(k, l))^{\frac{1}{2}} (\sum_{k=l=1}^m \sum_{l=1}^n U_R^2(k+p, l+q))^{\frac{1}{2}}} \quad (1)$$

匹配准则: 查找区域中最小的 $M_1(p, q)$ ($-w \leq p \leq w$ 和 $-w \leq q \leq w$) 所对应的像素块代表最佳的匹配块, $d_h = p$ 和 $d_v = q$ 分别表示运动矢量的水平分量和垂直分量。

(2) 均方误差(Mean Square Error, MSE)

$$MSE(p, q) = \frac{1}{mn} \sum_{k=l=1}^m \sum_{l=1}^n [U(k, l) - U_R(k+p, l+q)]^2 \quad (2)$$

匹配准则: 查找区域中最大的 $MSE(p, q)$ ($-w \leq p \leq w$ 和 $-w \leq q \leq w$) 所对应的像素块代表最佳的匹配块, $d_h = p$ 和 $d_v = q$ 分别表示运动矢量的水平分量和垂直分量。

(3) 绝对平均误差函数(MAE)

$$MAE(p, q) = \frac{1}{mn} \sum_{k=l=1}^m \sum_{l=1}^n |U(k, l) - U_R(k+p, l+q)| \quad (3)$$

匹配准则: 查找区域中最小的 $MAE(p, q)$ ($-w \leq p \leq w$ 和 $-w \leq q \leq w$) 所对应的像素块代表最佳的匹配块, $d_h = p$ 和 $d_v = q$ 分别表示运动矢量的水平分量和垂直分量。这里, 最优匹配相应于找 NCCF 的最大点。

求和绝对误差 SAD (Sum of Absolute Differences) 准则出现后, 迅速取代 MAE 且被各种运动估计算法采用, 因为它与 MAE 的匹配效果等价, 却去掉了不必要的除法, 计算量大大降低:

$$SAD(i, j) = \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} |I_2(m, n) - I_1(i+m, j+n)| \quad (4)$$

这是因为, 对采用相同形状大小的块来说, N_2 都是相同的, 除以它已经没有必要。另一方面, 在计算 SAD 的过程中, 当发现块的部分 SAD 已经大于当前最小 SAD 时, 可以中途退出, 从而使计算量大大减少。因此在本文中采用 SAD 作为块的匹配准则。

3 块模式选择的并行设计

3.1 块模式选择并行设计的理论依据

对视频帧空间区域的块模式进行选择的步骤如下。

Step1 先顺序选择帧中的未进行宏块划分的空间区域(如 16×16 像素大小的宏块), 然后对此宏块区域依次地选择上述 7 种块模式对此区域进行划分。

Step2 对于上述的 7 种宏块区域的划分, 分别以划分块为单位, 在多个参考中寻找其最佳的匹配块(即与当前块的能量残差值最小的匹配块)。

Step3 分别计算当前块与其对应的匹配块的残差能量值, 然后以块模式为单位, 将求得的各个块的残差能量值进行汇总, 求得最终的 7 个 SAD 值, 即 $SAD_1, SAD_2, \dots, SAD_7$ 。

Step4 将上述求得的 7 个 SAD 进行大小比较, 选出最小的 SAD。最小的 SAD 对应的块模式变为此宏块空间的最佳块划分模式。

Step5 再依次按照上面的步骤计算下一个未进行宏块划分的空间区域, 直到当前帧的空间区域都被块划分完毕为止。

通过上述块模式选择的步骤可知: 对同一个宏块空间, 上述 7 种模式的 SAD 的求值过程是完全独立的, 不存在前后的必然联系。故可以采用串行算法直接并行化的方法将其改造为并行算法, 以提高其运算速度, 其通信方式为共享内存的通信方式。

3.2 块模式选择的并行设计

通过以上 3.1 节的理论分析可知, 可以将块模式算法进行如下的并行设计。

块模式选择的并行设计:

```

Begin
  For each i par-do { //1 ≤ i ≤ 7
    for(int j=1; j ≤ num(i); j++) { //num(i)为模式 i 的
      块的数目
        for(int k=1; k ≤ num(f); k++) { //num(f)为参考
          帧的数目
            在帧 k 中寻找块 j 的最佳匹配块;
            求出当前块 j 与 k 帧最佳匹配块的残差能量值 E
            (k);
          }
          for(k=1; k ≤ num(f); k++)
            找出最小的 E(k), 放入变量 SAD(i, j) 中;
        }
      }
    }
  }
  SAD(i) = 0;

```

```

for(j=1;j<=num(i);j++)
    SAD(i)+=SAD(i,j);
}
for(i=1;i<=7;i++)
    找出最小的SAD(i);
最小的SAD(i)对应的块模式i变为当前宏块的最佳块模式。
End

```

3.3 块模式选择并行算法的时间复杂度分析

通过对块模式串、并行算法描述的分析可知,块模式串行算法的时间复杂度为: $O(s) = 7 \times (\text{num}(i) \times (\text{num}(f) \times 2) + \text{num}(i)) + 7 = 7 \times (\text{num}(i) \times (2\text{num}(f) + 1) + 1)$;块模式并行算法的时间复杂度为: $O(p) = \text{num}(i) \times (\text{num}(i) \times 2) + \text{num}(i) + 7 = \text{num}(i) \times (2\text{num}(f) + 1) + 7$, $O(s) / O(p) \approx 7$ 。通过对块模式串、并行算法时间复杂度的分析可知:当块模式并行算法的硬件条件满足时(即存在足够多的CPU个数),块模式的并行算法比其串行算法编码时间提高了接近7倍。

3.4 块模式选择并行设计的多核服务器实现

将块模式选择并行化完成以后,可以利用多线程技术来模拟,在多核服务器上实现。利用多线程技术实现块模式并行化的思想为:在程序中开辟7个线程,即thread1,thread2,...,thread7。用线程thread1来计算SAD₁,thread2来计算SAD₂,...,thread7来计算SAD₇。然后将各个线程计算完的数据进行比较,从而求得最佳的块模式。

4 实验结果与分析

4.1 实验环境

试验环境为:实验平台,JVT的JM9.3;编码序列,IBPBP格式;参考帧数5;率失真优化选项设置1;编码速度30fps;机器类型,惠普服务器;主频800MHZ;CPU个数4;操作系统Windows 2000;编译环境VC6.0;块搜索起始位置(0,0);条带划分规则编码序列中的所有帧都只划分一个条带;测试序列QCIF(176×144)格式的“Foreman”,“Stefan”,“Akiyo”。

表1 串行与并行的块模式选择的试验结果

(a) Foreman										
量化参数	串行算法			并行算法			优化比较			时间比值
	PSNR	码率	时间	PSNR	码率	时间	PSNR	码率	时间	
16	37.48	16022079	50.86	37.48	16022079	18.04	0	0	0	2.82
20	33.89	10875559	51.82	33.89	10875559	18.31	0	0	0	2.83
24	30.35	7027231	51.97	30.35	7027231	18.63	0	0	0	2.79
28	26.95	4288159	51.01	26.95	4288159	18.15	0	0	0	2.81
32	23.35	2373191	51.55	23.35	2373191	18.61	0	0	0	2.77
36	20.15	1263455	53.27	20.15	1263455	19.09	0	0	0	2.79

(b) Stefan										
量化参数	串行算法			并行算法			优化比较			时间比值
	PSNR	码率	时间	PSNR	码率	时间	PSNR	码率	时间	
16	37.89	7250180	41.77	37.89	7250180	14.97	0	0	0	2.79
20	34.52	4181804	42.46	34.52	4181804	15.11	0	0	0	2.81
24	31.33	2384252	44.44	31.33	2384252	16.28	0	0	0	2.73
28	28.51	1352196	45.81	28.51	1352196	16.54	0	0	0	2.77
32	25.61	778916	46.54	25.61	778916	16.62	0	0	0	2.80
36	22.95	480804	46.25	22.95	480804	16.70	0	0	0	2.77

(c) Akiyo

量化参数	串行算法			并行算法			优化比较			时间比值
	PSNR	码率	时间	PSNR	码率	时间	PSNR	码率	时间	
16	39.75	1560993	25.07	39.75	1560993	8.92	0	0	0	2.81
20	35.78	841657	24.96	35.78	841657	9.01	0	0	0	2.77
24	33.76	466697	27.08	33.76	466697	9.71	0	0	0	2.79
28	30.98	252689	26.86	30.98	252689	9.63	0	0	0	2.79
32	27.94	144241	27.72	27.94	144241	10.08	0	0	0	2.75
36	25.23	88305	25.78	25.23	88305	9.27	0	0	0	2.78

本实验进行两组测试,一组是串行的块模式搜索算法对全部预测块模式进行编码,另一组是采取本文提出的并行的块模式搜索算法的编码时间,分别对测试序列的复杂度、图像质量、码率与编码时间进行比较,如表1所示。

通过分析表1的实验数据,不难发现:块模式选择并行后与以前的块模式串行选择相比,峰值信噪比(PSNR)、码率没有发生变化,但是编码所用的时间有了显著缩短,缩短的时间幅度均在2.73倍以上。从而也进一步验证了块模式算法的并行化,对提高H.264/AVC编码器的编码速度效果显著。

结束语 本文对影响H.264/AVC编码速度的运动估计部分进行了优化,即对块模式选择算法进行了并行化。通过实验数据分析可知:优化后的运动估计在硬件条件允许的情形下(存在多个运算单元),其性能有了明显的提高。

随着计算机网络技术和计算机通信技术的迅猛发展,对于提供高质量网络音、视频等多媒体服务的需求越来越大。实时视频传输由于其独特的应用环境,使得传统的编码方法不再适用。通过对H.264/AVC编码器编码速度的优化,H.264/AVC编码器可以很好地满足市场需求。因而对H.264/AVC编码速度的改进将继续成为今后学术研究的一个热点,同时它的研究成果也必将对人们未来的生活产生重大影响。

参考文献

- [1] 韩立强,杨波.一种快速实现H.264运动估计的方法.光电工程,2006,33(8):94-97
- [2] 黄军浩,毕笃彦,许悦雷,等.基于H.264的多参考帧运动估计快速算法.计算机工程,2006(32):238-240
- [3] 丁永林,沈海滨.H.264/AVC中的运动估计算法对比及改进. Digital TV & digital video,2005,17(2):17-18
- [4] Draft ITU-T Recommendation H.264, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-TVCEq Document JVT C167, May 2002:45-49
- [5] ITU-T Recommendation H.261, Video Codec for Audiovisual Services at px64 k bit/s, Mar. 1993:19-21
- [6] Richardson I E G. H.264和MPRG-4视频压缩新一代多媒体的视频编码技术.欧阳合,韩军,译.长沙:国防科技大学出版社,2004:99-107
- [7] 沈兰荪,卓力,等.视频编码与低速率传输.北京:电子工业出版社,2001:27-31
- [8] 王文生,杨铭,崔会娟,等.H.264运动估计的自适应提前退出算法.清华大学学报(自然科学版),2004,44(10):28-31