

适合硬件实现的运动估计方法及其 VLSI 实现

宋 锐, 赵 波, 肖 嵩

(西安电子科技大学 综合业务网理论与关键技术国家重点实验室 陕西 西安 710071)

摘要: 根据视频编码器硬件实现的需求,给出了一种新的近似金字塔运动估计算法,通过对搜索窗中的匹配位置进行金字塔抽样,减少了每一层的搜索运算量。在此基础上,在最后一层的整像素搜索中,直接进行小块的匹配计算,去除了 MPEG-4 和 H. 264 中宏块运动估计之后对 8×8 或 4×4 小块估计的冗余计算。在硬件实现中改进了数据流的输入方式,利用类全搜索估计中数据流的规则性提高了运算的并行度。位置抽样后的搜索中每一层的实现有很强的相似性,通过复用硬件结构大大节省了硬件资源。在 MPEG-4 SP 编码器上的测试说明,这种方法的 PSNR 值相对于全搜索平均只有 0.02 dB 的损失,但它的运算量只有全搜索的 7%,占用的硬件资源仅为全搜索的 1/4,并且可满足实时性的需求。

关键词: 运动估计;新金字塔算法;硬件复用;实时处理

中图分类号: TN919.18 **文献标识码:** A **文章编号:** 1001-2400(2006)02-0257-05

A new motion estimation algorithm based on hardware implementation and its VLSI design

SONG Rui, ZHAO Bo, XIAO Song

(State Key Lab. of Integrated Service Networks, Xidian Univ., Xi'an 710071, China)

Abstract: A new pyramid motion estimation algorithm and its VLSI architecture are proposed based on the demand for hardware implementation of a video encoder. The computational load is reduced by the method of a pyramidal sample in searching positions. Redundant computation of 8×8 or 4×4 blocks is removed after macro blocks for the Sum of Absolute Difference (SAD) of sub-block are calculated directly in the last layer of integral search. During hardware implementation processing, data stream access mode is revised, and parallel process efficiency is improved as well. After positions sampling, each layer has a similar architecture, and hardware resource is saved significantly by use of the common module. Simulation results on MPEG-4 SP show that although the new pyramid algorithm is 0.02 dB lower than full search in PSNR, the process load and hardware resource occupancy are 7% and 25%, respectively, that of full search, which will meet the requirements for real time applications.

Key Words: motion estimation; new pyramid algorithm; hardware reuse; real time process

基于块的运动估计与补偿 (ME/MC) 是视频编解码系统中的重要组成部分,在当今的视频编码标准中得到了广泛的应用,如 H. 261, H. 263, MPEG-1/2/4 等^[1~4]。在 H. 264 编码标准中将运动估计进一步细化,采用树型结构的运动估计,更大限度地去除帧间冗余性^[5]。然而,运动估计的复杂性和巨大的运算量是视频编码实时实现的瓶颈。为此,人们提出了多种快速算法,如三步法 (TSSA)^[6], 钻石搜索法 (DSA)^[7], 矢量场自适应搜索法 (MVFAST) 以及带预测的矢量场自适应搜索法 (PMVFAST)^[8] 等等。这些算法加速了运动估计,但它们数据流不规则,每次估计运算的时间不确定,所以,硬件实现中多采用类全搜索的搜索算法。

收稿日期 2005-06-14

基金项目 国家自然科学基金资助项目(60372043)

作者简介 宋 锐(1982-),男,西安电子科技大学博士研究生。

全搜索算法的数据流规则,可充分利用相邻匹配位置数据的相关性提高并行度,但它的运算时间长,在很多应用中难以达到实时性需求。笔者在全搜索算法硬件实现基础上给出了一种新的金字塔运动估计加速算法,将运动估计在不同的精度层上分步完成,既利用了全搜索方式数据流的规则性,同时又可满足 MPEG-4 和 H. 264 中对更小块在小范围内的精细匹配。算法在每一步上的运算量比较小,不同步骤的搜索可复用硬件资源,对内部存储器的 I/O 要求也比较低,因此易于硬件实现。

1 新的金字塔运动估计算法(NPA)

在块匹配运动估计中,常用最小绝对差函数(MAD)作为失真度量,假设块大小为 $M \times N$, f_{k-1} 和 f_k 分别是参考帧和当前帧的灰度值,则在 (i, j) 位置的 MAD 定义为

$$S_{\text{MAD}}(i, j) = (1/(MN)) \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|$$

对应最小 S_{MAD} 的 (i, j) 即为当前块的运动矢量。全搜索算法对搜索窗中每个位置 (i, j) 都要进行匹配计算,运算量巨大。结合硬件实现的有效性,后来提出了一种块和金字塔(BSPA)^[9],这种方法采用预淘汰技术大大加速了估计速度,但这种方法有固有的不足,在硬件的实现中比较突出的是它需要额外的运算单元和存储空间,计算这部分数据也需要单独的时间开销,不利于硬件的实时实现。

BSPA 中和金字塔的运算相当于对图像像素进行下采样,降低空间分辨率,对这一方法进行改进,不再进行图像下采样计算,而进行搜索位置的采样,在不同的层对搜索窗中的不同范围内的位置进行采样,先进行 n ($n > 1$) 像素层搜索,即在搜索窗中行列方向每隔 $n-1$ 个位置对一个位置进行匹配计算,在得到的最佳匹配位置 $\pm n/2$ 范围内做 $n = n/k$ ($k > 1$) 层搜索,直至整像素层搜索。这一方法相对于 BSPA 方法有两个优点:(1)不需要进行块和金字塔计算;(2)在前几层的搜索中以宏块为单位进行计算可快速缩小范围,在最后一层整像素搜索中直接以 8×8 或 4×4 的小块进行匹配,可以去掉小块估计的冗余计算。

算法实现过程为:假设宏块大小为 $M \times N$,宏块在搜索窗中的中心位置为 (p, q) ,搜索范围为 $[-2^{f_{\text{code}}+3}, 2^{f_{\text{code}}+3}-1]$,那么搜索窗口大小为 $(M+2 \times 2^{f_{\text{code}}+3}) \times (N+2 \times 2^{f_{\text{code}}+3})$,在第 n 层的搜索中,对参考帧中搜索窗内 $(p \pm tn, q \pm tn)$, $t = 0, 1, \dots$ 的位置进行匹配运算,得到最佳匹配位置。然后根据这一位置更新搜索中心 (p, q) 和 n 做下一层搜索,下面以 $n = 4, k = 4, f_{\text{code}} = 1$ 的情况为例对 NPA 算法进行详细描述。在这种情况下搜索范围为 $[-16, 15]$,运动估计两层完成整像素运动估计,如图 1 所示。

搜索第一步以 (p, q) 为中心,做 $(\pm 4t, \pm 4t)$, $t = 0, 1, \dots$ 位置的匹配,如图 1 中实心圆圈所示的位置,得到 4 像素层运动矢量 (m_x, m_y) 后,在以 (m_x, m_y) 为中心 ± 2 的范围内每个位置进行匹配,如图 1 中空心的方块所示,得到整像素层运动矢量。在具体的实现中,在 ± 2 范围内搜索时,可直接以 8×8 的块为单位进行匹配,将 8×8 的块在每个位置的 SAD 值保存,4 个 8×8 块对应位置 SAD 值的加和即是 16×16 宏块的 SAD 值。

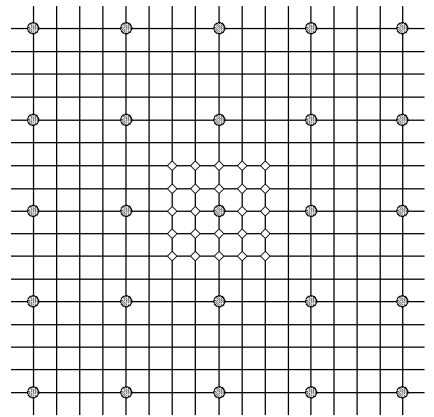


图 1 $n = 4, k = 4$ 时分级搜索步骤示意图

2 算法的硬件实现

2.1 不同层搜索的硬件复用

NPA 搜索法充分利用全搜索方法中数据流的规则性,便于硬件实现。在文献 [10] 提出的全搜索运动估计硬件结构基础上做一些修改,应用于 NPA 搜索法。NPA 搜索法每一层的运算有很强的相似性,可复用硬件结构。它的硬件实现示意图如图 2 所示。

图 2 中,在每一层硬件实现时,NPA 算法的数据流与全搜索结构的数据流^[10]有所不同,将搜索窗中每一

行的数据顺序读出,搜索窗数据在每个 PE 中不经过延迟直接接入,通过运算单元的使能信号选择数据输入 PE 中.每个 PE 中需要的当前宏块数据相同,所以当前宏块的数据经过延迟单元输入到 PE 中进行运算.在 PE 运算单元中完成绝对差累加得到每个位置的 SAD 值.虚线框内的部分在每一层的运算中复用,在每一层运算得到的 SAD 值经过比较器得到最佳匹配位置,然后将最佳匹配位置信息反馈到前端地址产生单元,计算下一层运算的数据读地址,计算完成之后启动下一层的运算.

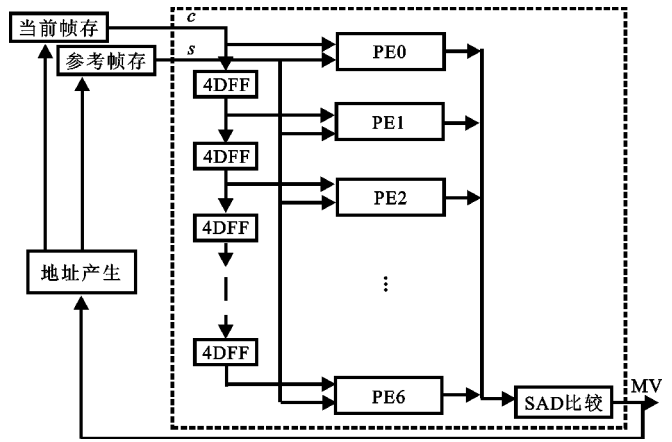


图 2 搜索单元结构复用示意图

如果搜索范围为 $[-16, +16]$ 在全搜索的硬件结构中需要用 32 个 PE 单元来并行处理完成运动估计,而在 NPA 算法的硬件结构中只需要 7 个 PE 单元,加上一些硬件复用的控制开销,所用的硬件资源只有全搜索硬件结构的 1/4,在 VLSI 实现中相当可观.

2.2 精细块搜索的高效实现

采用这一算法的另一个优点在于能更加高效地完成精细块的搜索.在 MPEG-4 中,加入了 4MV 技术,不仅要得到宏块的运动矢量,而且要在宏块最佳匹配位置周围做块的运动估计,通过一定的准则选择编码模式.在这其中,有些运算是冗余的,可从 SAD 的计算公式得出

$$S_{SAD}(i, j) = \sum_{m=0}^{15} \sum_{n=0}^{15} |f_k(m, n) - f_{k-1}(m+i, n+j)| = \sum_{m=0}^7 \sum_{n=0}^7 |f_k(m, n) - f_{k-1}(m+i, n+j)| + \sum_{m=0}^7 \sum_{n=8}^{15} |f_k(m, n) - f_{k-1}(m+i, n+j)| + \sum_{m=8}^{15} \sum_{n=0}^7 |f_k(m, n) - f_{k-1}(m+i, n+j)| + \sum_{m=8}^{15} \sum_{n=8}^{15} |f_k(m, n) - f_{k-1}(m+i, n+j)|$$

由上式可看出,宏块在某一位置的 SAD 值计算可分成 4 部分,相应地可将宏块分成 4 个 8×8 的块,分别为 a, b, c, d ,如图 3 所示.那么,宏块在该位置的 SAD 值 $S_{SAD_MB}(i, j)$ 就可通过 4 个小块在同一位置的 SAD 值 $S_{SAD,a}(i, j), S_{SAD,b}(i, j), S_{SAD,c}(i, j), S_{SAD,d}(i, j)$ 得到,即 $S_{SAD_MB}(i, j) = S_{SAD,a}(i, j) + S_{SAD,b}(i, j) + S_{SAD,c}(i, j) + S_{SAD,d}(i, j)$.

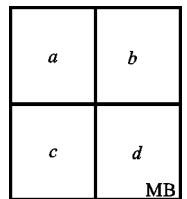


图 3 宏块划分

NPA 算法在整像素最后一层搜索时,搜索范围已经缩小到很小,在这一范围内做 8×8 的块的搜索运算量并不大,而相应位置宏块的 SAD 值也可通过对 4 个 8×8 的块对应位置的 SAD 值进行运算直接得到.在硬件实现中 4 个块的运动估计顺序完成,在做完每个块的每个位置的 SAD 计算之后,将计算结果 SAD 值保存起来,进行累加.4 个块的运动估计全部做完之后,对应位置的 SAD 值累加结果就是宏块的 SAD.4 个块的运动估计硬件可复用图 2 中的硬件资源,只需要加入一些配置参数输入.

H. 264 中引入了树型运动估计,运动估计可以在更小的 4×4 块上完成,一般的运算方法是根据树型结构分别进行宏块, 16×8 和 8×16 的块的估计,根据判据,在需要的情况下做 4×4 的运动估计,但这种从上而下的估计方法在硬件上实现时很浪费时间,硬件实现需要足够的并行度.对 SAD 的计算公式进一步分解,可得知,宏块的 SAD 和 $16 \times 8, 8 \times 16$ 的块的 SAD 都可由对应位置的 4×4 的小块的 SAD 运算得到.因此,在 H. 264 中也可采用同样的硬件结构,在最后一个整像素层,先对每个 4×4 的小块进行估计并保存每个位置的 SAD,通过计算得到对应位置宏块和 $16 \times 8, 8 \times 16$ 的块的 SAD 值.

2.3 运算复杂度分析

这里以 $n = 4, k = 4$ 的配置为例进行复杂度分析.假定一次减法或一次加法的运算量为 1,一次绝对值计算的运算量为 1.5,那么在搜索范围为 $[-16, 15]$ 时,整像素全搜索的总运算量为 916 480,新金字塔搜索法的总运算量为 66 230,新金字塔搜索法的运算量仅为全搜索运算量的 0.072 3 倍.如果将 8×8 的计算考虑

在内,全搜索的运算量为 938 780,而新金字塔搜索法的运算量仍为 66 230,这时新金字塔搜索法的运算量仅为全搜索运算量的 0.070 5 倍.通过计算可看出,新金字塔搜索法大大减少了估计运算量.算法的复杂度与硬件的功耗有密切联系,运算复杂度的降低会大大减少硬件的功耗.

3 性能分析

为了将新的金字塔算法与全搜索和原金字塔算法进行性能比较,对 Stefan(CIF),Akiyo(CIF),Mother(CIF)序列 100 帧进行了测试,测试中使用的配置为 $n = 4, k = 4$,搜索窗口大小为 $[-16, 15]$,用 MPEG-4 SP 编码器做仿真,设定 I 帧 $Q_p = 16$,P 帧 $Q_p = 16$,每隔 15 帧插入一个 I 帧.图 4 是对全搜索算法、金字塔算法和新金字塔算法的仿真结果,从仿真的结果可看出,采用新金字塔算法做运动估计得到的解码图像 PSNR 值与全搜索和金字塔算法相差很小.仿真统计数据如表 1 所示.

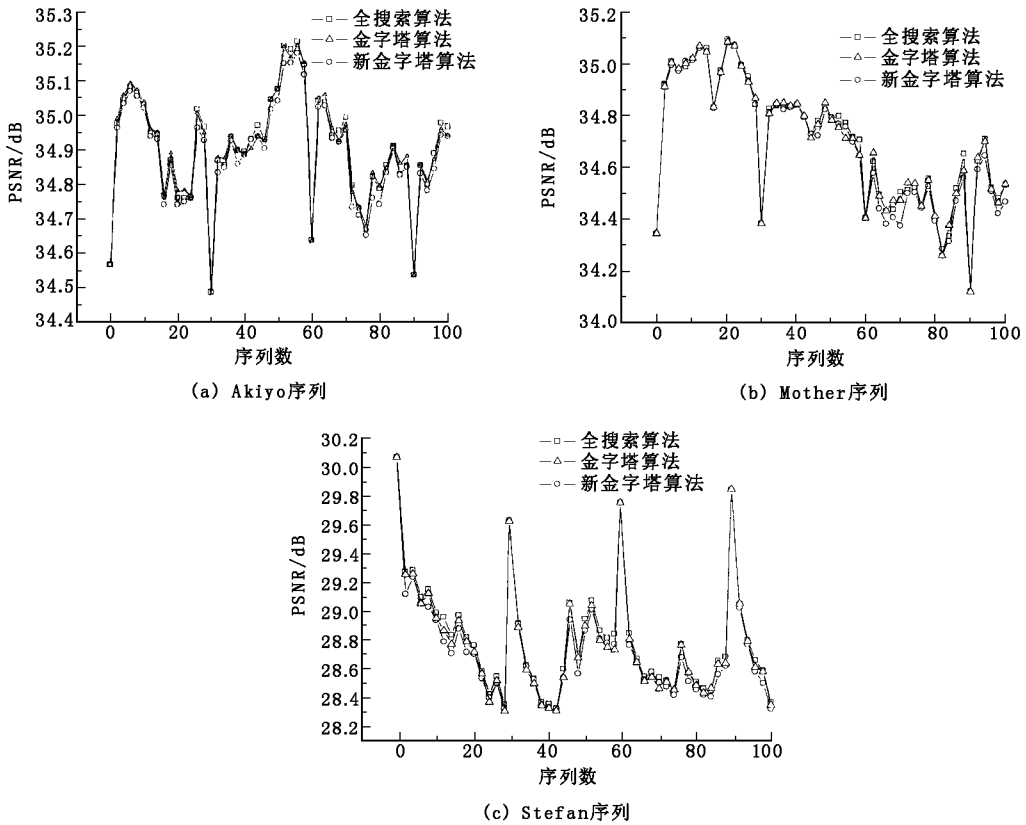


图 4 $I_{Q_p} = 16, P_{Q_p} = 16$,序列重建图像 PSNR 值比较

表 1 图像序列仿真统计数据

图像序列	搜索算法	平均 PSNR			平均每帧 bit 数
		PSNR_Y	PSNR_U	PSNR_V	
Akiyo	全搜索	34.909 206	37.606 327	40.295 216	4 013.306 885
	金字塔搜索	34.910 915	37.622 810	40.314 293	4 023.366 455
	新金字塔搜索	34.887 844	37.612 583	40.294 312	4 027.722 656
Mother	全搜索	34.713 390	40.082 664	41.790 440	4 421.861 328
	金字塔搜索	34.710 392	40.108 089	41.795 624	4 480.791 992
	新金字塔搜索	34.693 954	40.088 390	41.770 962	4 462.020 020
Stefan	全搜索	28.785 049	34.363 991	33.856 678	21 134.654 300
	金字塔搜索	28.754 442	34.329 117	33.829 189	21 632.158 200
	新金字塔搜索	28.728 741	34.278 297	33.760 700	22 898.296 880

从仿真结果可看出,新的金字塔算法在性能上与全搜索和金字塔算法相差不大,在物体运动比较剧烈的情况下,新金字塔算法在 PSNR 值上相对于全搜索有平均约 0.05 dB 的损失,需要的比特数为全搜索的 1.08 倍,而在图像运动比较平缓的情况下,新金字塔算法的 PSNR 值相对于全搜索只有平均约 0.02 dB 的损失,需要的比特数仅为全搜索的 1.009 倍.图 5 给出了 Stefan 序列采用 3 种估计算法的第 44 帧重建图像.



图 5 Stefan 序列第 44 帧重建图像

4 结束语

根据硬件资源和实时性的需要,给出了一种新的适合硬件实现的金字塔估计算法.这一算法将搜索位置分层,减少每一层的运算量,在图像质量损失很小的情况下,大大减少了整个运动估计的运算时间.在硬件实现中,利用了数据流的规则性和每一层搜索的相似性,通过复用搜索单元,节省了硬件资源.算法通过搜索位置分层,逐步缩小搜索范围,在最后一步的整像素搜索中直接计算小块的 SAD 值,不仅达到了计算 4MV 的效果,同时又得到了宏块的 SAD,节省了冗余计算.应用于 H.264 中,在最后一层整像素搜索的小范围内直接进行 4×4 小块的计算,可通过简单运算同时得到不同块大小的 SAD 值,达到节省时间和硬件资源的效果.

参考文献:

- [1] ISO/IEC 14469-2. Coding of Audio-visual Objects—Part 2: Visual[S]. 1999.
- [2] Sikora T. Trends and Perspectives in Image and Video Coding[J]. Proceedings of the IEEE, 2005, 93(1):6-17.
- [3] 王养利,吴成柯.基于三焦距张量及 BMA 的运动估计和补偿[J].航空学报,2001,22(2):117-120.
- [4] Yan Yaoping, Lu Zhaoyang, Wu Chengke. Motion Estimation and Compensation Based on the DT Mode[J]. Journal of Xidian University, 1997, 24(3):357-362.
- [5] ITU-T/ISO/IEC. H.264 Advanced Video Coding for Generic Audiovisual Services[S]. 2005.
- [6] Li R X, Zeng B, Liou M L. A New Three-step Search Algorithm for Block Motion Estimation[J]. IEEE Trans on Circuits and Systems for Video Technology, 1994, 4(4):38-42.
- [7] Zhu S, Ma K K. A New Diamond Search Algorithm for Fast Block-matching Motion Estimation[J]. IEEE Trans on Image Processing, 2000, 9(2):287-290.
- [8] Tourapis A M, Au O C, Liou M L. Fast Block-matching Motion Estimation Using Predictive Motion Vector Field Adaptive Search Technique(PMVFAST) [DB/OL]. ISO/IEC/JTC1/SC29/WG11 M5866, 2000-04-08.
- [9] Han T H, Hwang S H. A Novel Hierarchical Search Block Matching Algorithm and VLSI Architecture Considering the Spatial Complexity of the Macroblock[J]. IEEE Trans on Consumer Electronics, 1998, 44(2):337-342.
- [10] Yang K M, Sun M T, Wu L. A Family of VLSI Designs of the Motion Compensation Block Matching Algorithm[J]. IEEE Trans on Circuits and Systems for Video Technology, 1989, 36(10):1317-1325.

(编辑:齐淑娟)

