

基于望远镜搜索的块匹配运动估值的 低功耗 VLSI 结构

张武健¹ 邱晓海¹ 周润德¹ 陈弘毅¹ Kondo Toshio² Nakashima Takayoshi² Ishitani Tsunehachi²

(1 清华大学微电子学研究所, 北京 100084, 中国)

(2 NTT 电子公司 (NEL 株式会社), 海老名市, 日本)

摘要: 在一种基于望远镜搜索的块匹配运动估值的 VLSI 实现中, 对用于加速搜索的传统心动阵列引擎进行了结构上的改进, 从而能够显著地降低功耗. 方法是使用一种新的块匹配误差计算的提早跳出技术, 并通过在阵列处理单元中屏蔽操作数来避免不必要的计算操作. 基于算法模拟结果的简单估计表明: 使用新结构搜索引擎的运动估值, 功耗可降低到原来的 40% 左右, 而仍然保持着相同的处理速度和相似的视频解码图质量.

关键词: 望远镜搜索; 运动估值; 低功耗; VLSI 结构; 提早跳出

EEACC: 6140C

中图分类号: TN912.3

文献标识码: A

文章编号: 0253-4177(2001)05-0670-07

1 引言

视频压缩是实现交互式视频通讯, 数字存储媒体等应用的最重要技术之一. 由于高度的计算复杂性, 其实时实现必须要求硬件的支持. 为面向移动式应用, 以及为降低芯片封装和散热的成本, 迫切要求低功耗的视频编码器^[1].

块匹配运动估值被广泛地应用于各种视频编码器中. 这一过程要求进行运动矢量的搜索, 即在参考图一定的范围内找到与当前待编码宏块最佳匹配的宏块, 相应的宏块位移就是运动矢量. 最小绝对差值和(SAD)是最有效的, 也是最适合于 VLSI 实现的块匹配准则. 最直接的全局搜索方法是通过检测每一个可能的候选宏块能够得到最优解, 却要求很大的运算量, 因而会消耗很高的功耗.

为降低运动估值的计算复杂性和功耗, 出现了许多快速的搜索方法, 如二维对数搜索^[2], 三步搜索^[3]等. 其中有一类算法的基本思想是尽可能早地判断出不可能是最佳匹配的候选参考图宏块, 从而避免不必要的宏块匹配误差(宏块中所有像素的匹

配误差和) 计算. 预先筛选技术(Successive Elimination)^[4]利用不等式的放缩得到宏块匹配误差的保守近似, 在计算候选宏块的匹配误差之前先计算要求较小运算量的保守近似. 如果保守近似表明该候选宏块不可能成为最佳匹配宏块, 将不必要进行严格的宏块匹配误差计算. 该技术已被用于低功耗的运动估值器设计^[5], 但是为计算宏块匹配误差的保守近似需要多个加法器和一个加法树, 额外的硬件开销较大. 与预先筛选不同, 提早跳出技术(Early Jump Out)是在计算宏块匹配误差的过程中判断候选宏块能否成为最佳匹配宏块. 简单地说, 如果发现某候选宏块中部分像素的匹配误差和大于已经检测出的最小宏块匹配误差, 显然该宏块不可能是最佳匹配, 于是即可中止宏块匹配误差的计算过程. 文献[6]提出了一种自适应的提早跳出方法. 传统的提早跳出技术虽然简单、直观, 却存在着一个致命的缺点: 针对各个候选宏块的匹配误差计算必须是串行的, 即在一个候选宏块的匹配误差计算完成之后才能进行下一个候选宏块的匹配误差计算. 只有这样, 在宏块匹配误差的计算过程中才能利用到前面检测出的最小宏块匹配误差. 这一缺点破坏了搜索的并

张武健 男, 1974 年出生, 博士研究生, 现从事视频图象压缩方面的研究.

周润德 男, 1945 年出生, 教授, 博士生导师, 从事数字信号处理、低功耗超大规模集成电路设计等研究.

2000-04-15 收到, 2000-7-14 定稿

行性,使得该技术只限于在基于简单树结构^[7]的 VLSI 实现中能够得到应用.而另一方面,为降低运动估值对数据存储器带宽的要求,心动阵列 (Systolic Array) 才是最常用的结构.显然传统的提早跳出技术无法被应用于要求并行计算的心动阵列实现中,到目前为止也还没有发现有文献报道它被用于心动阵列实现中.

本文提出了一种适合于心动阵列结构的新提早跳出技术,其基本思想是并行计算各个候选宏块的匹配误差,并实时比较它们的部分匹配误差.如果提早判断出某候选宏块不可能成为最佳匹配宏块,在传统心动阵列搜索引擎的处理单元中通过屏蔽操作数来避免不必要的计算.提早跳出判断要求的运算量非常小,要求的额外硬件开销也很小.基于算法模拟结果的简单估计表明:使用新结构搜索引擎的运动估值,功耗可降低到原来的 40% 左右,而仍然保持着相同的处理速度和相似的视频解码图质量.

2 多分辨率望远镜搜索算法

为保证视频质量,运动矢量的搜索要求在大范围内进行.尤其是在面向数字电视广播等应用时,几乎没有实际的视频编码器采用了真正意义上的全局搜索,然而很多快速算法仍然要求进行局部的“全搜索”,如多分辨率望远镜搜索 (MRTLcS) 算法^[8](图 1).

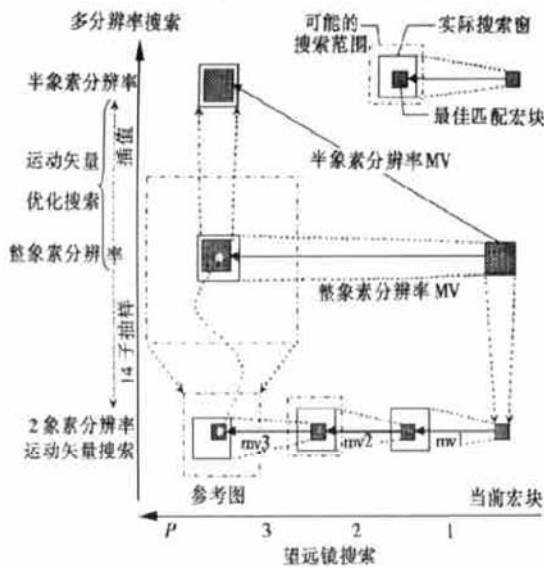


图 1 多分辨率望远镜搜索

FIG. 1 Multiresolution Telescopic Search

望远镜搜索方法^[9]被用于多分辨率搜索的粗搜

索过程中,它借助当前待编码图和参考图之间的中间图通过逐步跟踪找到最终的运动矢量.设中间图数目为 D , 每一步实际搜索窗的大小均为 $SX \times SY$, 则最终可能的搜索范围可扩大到 $(D+1)SX \times (D+1)SY$. 如果水平和垂直方向的子采样因子均为 2:1, 则 MRTLcS 算法的运算量仅为全局搜索算法的 $1/(16(D+1))$. 由于 MRTLcS 算法在性能和计算复杂性两方面有较好的折衷,它已经被用于实际的 MPEG2 视频编码器中^[10,11].

很显然望远镜搜索实质上是一个多步的局部“全搜索”过程.设当前宏块的大小为 $M \times N$, 在 SAD 准则下,第 P 步 ($1 \leq P \leq D+1$) 的搜索如下表示:

$$SAD_P(u, v) = \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} |T(i, j) - R_P(i + u + mv_{xP-1}, j + v + mv_{yP-1})|$$

$$(mv_{xP}, mv_{yP}) = (u + mv_{xP-1}, v + mv_{yP-1}) \Big|_{SAD_P(u, v) = \min}$$

$$u \in [0, SX - 1], v \in [0, SY - 1]$$

$$(mv_{x0}, mv_{y0}) = (0, 0) \quad (1)$$

这里 $T(i, j)$ 和 $R_P(i + u + mv_{xP-1}, j + v + mv_{yP-1})$ 分别表示当前宏块中 (i, j) 位置处的象素亮度值, 和第 P 幅参考图(为简单计,我们把中间图也称作参考图)在可能的搜索范围 $(P \cdot SX) \times (P \cdot SY)$ 中, $(i + u + mv_{xP-1}, j + v + mv_{yP-1})$ 位置处的象素亮度值.望远镜搜索共分 $D+1$ 步进行,其中每一步都是 $SX \times SY$ 范围内的全搜索,进行 $SX \times SY$ 个候选宏块的匹配误差即 SAD 值计算,得到该步的运动矢量 (mv_{xP}, mv_{yP}) .降低每一步局部全搜索的运算量能够减小整个望远镜搜索的运算量,从而能够减小运动估值的功耗.

3 新提早跳出技术

3.1 部分和 PSAD 的实时比较

对一个全搜索过程,新提早跳出技术的基本思想是并行地计算各个候选宏块的匹配误差即绝对差值和 SAD 值,并且实时地比较部分和 PSAD 值.如果某个候选宏块的部分和比已经检测出的最小部分和超出很多,则认为该候选宏块不可能成为最佳的匹配宏块,此时便中止对该候选宏块匹配误差的计算.下面是算法过程:

```

for v= 0 to SY- 1 do
  for u= 0 to SX- 1 do
    {SAD(u, v)= 0; vld(u, v)= 1;}
  for j= 0 to N- 1 do {
    dminj= ∞;
    for v= 0 to SY- 1 do
      for u= 0 to SX- 1 do {
        if (vld(u, v)= 1) then {
          SAD(u, v)= SAD(u, v) +  $\sum_{i=0}^{M-1} |T(i, j) - R(i + u, j + v)|$ ;
          if (SAD(u, v) < dminj) then {
            dminj= SAD(u, v);
            if(j = N- 1) then
              {mvx= u; mvy= v;}
            else if (SAD(u, v) > f(dminj) && j < N- 1) then
              vld(u, v)= 0;
          }}}
  }

```

(2)

其中 f 是 $dmin_j$ 的增函数.

通常的全搜索只是在累加完候选宏块中所有象素的匹配误差之后, 即当 $j = N - 1$ 时才进行 SAD 值的比较, 最小 SAD 值 $dmin_{N-1}$ 对应的宏块位移 (u, v) 就是运动矢量 (mvx, mvy) . 但是新提早跳出技术要求每累加完一行象素的匹配误差之后, 都要进行部分累加和的比较, 以实时地更新当前最小的部分累加和 $dmin_j (j = 0, 1, \dots, N - 2)$. 并且当某一候选宏块的部分累加和大于函数值 $f(dmin_j)$ 时, 认为该宏块不可能成为当前宏块的最佳匹配宏块, 设对应的标志 $vld = 0$, 不再继续进行剩余象素匹配误差的累加.

3.2 比较方法

本文给出两种部分绝对差值和 PSAD 的比较方法, 它们是比例比较法和增量比较法, 其区别体现于不同的 f 函数形式.

比例比较法:

在比例比较法中,

$$f(dmin_j) = k_j dmin_j \quad 0 \leq j \leq N - 2 \quad (3)$$

其中 k_j 是仅与宏块行序号相关的常系数. 显然 SAD 值计算中断时对应的行序号越小, 可节省的运算量就越多. 然而, 此时也因为只有较少的象素参与了实际的比较, 使得误判断的风险增大. 因而, 越是小行序号对应的系数 k_j 对算法的性能和计算复杂

性影响越大. 在进行系数值选取时, 我们按行序号 j 由小至大的顺序依次优化 k_j , 并在优化 k_j 时令 $k_{j+1} = k_{j+2} = \dots = k_{N-2} = \infty$. 最终有 $k_0 > k_1 > \dots > k_{N-2}$, 行序号越小, 对应的比较条件越严格.

增量比较法:

在增量比较法中,

$$f(dmin_j) = dmin_j + th \quad 0 \leq j \leq N - 2 \quad (4)$$

其中 th 是随图象内容自适应的阈值. 设当前宏块所在行的上一行所有宏块的最小匹配误差的平均值为 $MSAD_{pre-slice}$, 在当前宏块的望远镜搜索过程中第 $P - 1$ 步得到的最小宏块匹配误差为 $MSAD_{P-1}$, 则进行第 P 步的搜索时, th 按(5)式确定:

if (当前宏块在首行) then

th = ∞;

else

if ($P = 1$) then

th = $MSAD_{pre-slice}/4$;

else

$$th = (MSAD_{pre-slice} + MSAD_{P-1})/8; \quad (5)$$

表面看来, 在增量比较法中阈值 th 与行序号 j 无关. 但是因为随着行序号的增加部分累加和是逐渐增大的, 因而对于小行序号时的部分和, th 值相对较大; 而对于大行序号的部分和, th 值就相对较小了. 这隐含着比较条件是随着行序号的增加逐渐宽松的.

4 低功耗 VLSI 结构

本文将新的提早跳出技术应用于传统的心动阵列结构搜索引擎^[8, 10, 11]. 当判断出某个候选宏块的部分匹配误差 PSAD 大于函数值 $f(dmin_j)$ 时, 我们将禁止第 $j+1, j+2, \dots, N-1$ 行象素匹配误差的计算和累加. 只要禁止的运算消耗的功耗能够补偿因比较和计算函数值 $f(dmin_j)$ 消耗的功耗, 就能够降低最终的功耗消耗.

在提早跳出判断时, 计算函数值 $f(dmin_j)$ 只需要用到目前最小的部分累加和 PSAD, 而并不需要使用完全 SAD 值的信息, 这使得各个候选宏块 SAD 值的并行计算成为可能. 给出的低功耗结构在传统的心动阵列基础上增加了提早跳出判断单元 EJOD_{*j*} ($j = 0, 1, \dots, N-2$), 以及处理单元使能信号控制. 如图 2 所示.

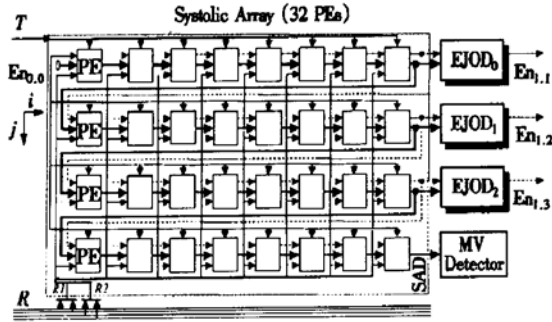


图 2 搜索引擎的低功耗 VLSI 结构

FIG. 2 Low Power Consumption VLSI Structure of Search Engine

4.1 处理单元阵列

图 2 所示的处理单元阵列采用了面向宏块象素的一维心动结构, 由 32 个处理单元(PE) 组成, 对应 4:1 子采样后当前宏块一场的 8×4 个象素(在文献[8, 10, 11]中搜索是基于场预测模式进行的). 这些象素数据 T 通过一条全局广播的总线在心动阵列开始计算的同时串行写入到对应 PE 单元的锁存器中. 而参考图数据 R 是通过梳状总线供给各个 PE 单元的.

PE_{*i,j*} 的功能是计算 T 与对应 R 的差值绝对值

$$AD_{(u,v)}(i,j) = |T(i,j) - R(i+u, j+v)|,$$

然后累加前面处理单元传递过来的部分绝对差值和 PS(u, v).

$$PS_{(u,v)}(i,j) = PS_{(u,v)}(i-1,j) + AD_{(u,v)}(i,j)$$

$$i = 1, \dots, M-1,$$

$$PS_{(u,v)}(i,j) = PS_{(u,v)}(M-1, j-1) + AD_{(u,v)}(i,j)$$

$$i = 0, \text{ 且 } PS_{(u,v)}(M-1, -1) = 0.$$

图 3 给出了对应阵列中首行 PE 单元的数据流, 为简单计我们只列出对应的求象素差值运算. 对于一个候选宏块来说, 其所有象素匹配误差的计算和累加是分布在 32 个 PE 单元中的, 并且是串行的. 而各个候选宏块的 SAD 值计算按照行扫描的顺序依次相隔一个时钟周期, 因此是准并行的.

PE00	PE10	PE20	PE60	PE70
R0, 0-T0, 0					
R1, 0-T0, 0	R1, 0-T1, 0				
R2, 0-T0, 0	R2, 0-T1, 0	R2, 0-T2, 0			
R3, 0-T0, 0	R3, 0-T1, 0	R3, 0-T2, 0			
R4, 0-T0, 0	R4, 0-T1, 0	R4, 0-T2, 0			
R5, 0-T0, 0	R5, 0-T1, 0	R5, 0-T2, 0			
R6, 0-T0, 0	R6, 0-T1, 0	R6, 0-T2, 0		R6, 0-T6, 0	
R7, 0-T0, 0	R7, 0-T1, 0	R7, 0-T2, 0		R7, 0-T6, 0	R7, 0-T7, 0
R0, 1-T0, 0	R8, 0-T1, 0	R8, 0-T2, 0		R8, 0-T6, 0	R8, 0-T7, 0
R1, 1-T0, 0	R1, 1-T1, 0	R9, 0-T2, 0		R9, 0-T6, 0	R9, 0-T7, 0
R2, 1-T0, 0	R2, 1-T1, 0	R2, 1-T2, 0		R10, 0-T6, 0	R10, 0-T7, 0
R3, 1-T0, 0	T3, 1-T1, 0	R3, 1-T2, 0		R11, 0-T6, 0	R11, 0-T7, 0
R4, 1-T0, 0	T4, 1-T1, 0	R4, 1-T2, 0		R12, 0-T6, 0	R12, 0-T7, 0
R5, 1-T0, 0	T5, 1-T1, 0	R5, 1-T2, 0		R13, 0-T6, 0	R13, 0-T7, 0
R6, 1-T0, 0	T6, 1-T1, 0	R6, 1-T2, 0		R6, 1-T6, 0	R14, 0-T7, 0
R7, 1-T0, 0	T7, 1-T1, 0	R7, 1-T2, 0		R7, 1-T6, 0	R7, 1-T7, 0

图 3 对应首先 PE 单元的数据流

FIG. 3 Data Corresponding First Line Cell PE

时钟周期 1 中, PE_{0,0} 计算 $AD_{(0,0)}(0,0) = PS_{(0,0)}(0,0) = |T(0,0) - R(0,0)|$;

时钟周期 2 中, PE_{1,0} 计算 $AD_{(0,0)}(1,0) = |T(1,0) - R(1,0)|$, $PS_{(0,0)}(1,0) = PS_{(0,0)}(0,0) + AD_{(0,0)}(1,0)$,

PE_{0,0} 计算 $AD_{(1,0)}(0,0) = PS_{(1,0)}(0,0) = |T(0,0) - R(1,0)|$;

.....

时钟周期 32 中, PE_{7,3} 计算 $AD_{(0,0)}(7,3) = |T(7,3) - R(7,3)|$, $PS_{(0,0)}(7,3) = PS_{(0,0)}(6,3) + AD_{(0,0)}(7,3)$,

.....

经过 32 个时钟周期在 PE_{7,3} 中得到位置 $(u, v) = (0, 0)$ 处候选宏块的匹配误差即 SAD(0, 0), 然后依次得到 SAD(1, 0), SAD(2, 0), ... 运动矢量检测单元比较 SAD 值, 最小 SAD 值对应的 (u, v) 即运动矢量. 该阵列完成一次 $SX \times SY$ ($SX = mM, m \geq 1$) 范

围内的全搜索共需要 $(32 + SX \times SY)$ 个时钟周期. 实际的 PE 单元在 AD 值计算和累加之间插入了一级流水线(寄存器), 其结构如图 4 所示.

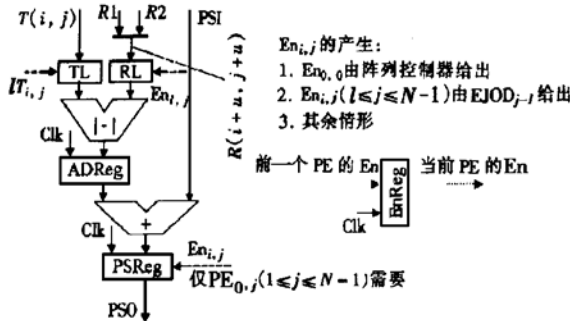


图 4 处理单元 $PE_{i,j}$

FIG. 4 Processing Cell $PE_{i,j}$

为避免不必要的匹配误差计算, 我们在每个 PE 单元的 R 输入端都插入了锁存器 RL, 如图 3 所示. 通常 RL 是透明的, 但当 $EJOD_{j-1}$ 单元判断出不可能成为最佳匹配宏块的候选宏块时, 有效计算标志 $vld_j = 0$, 于是第 j 行及其后面行(除第 j 行首列) PE 单元的 RL 依次变为不透明. 锁存器 RL 关闭时, 能够避免写入新的 R 值, 于是计算 AD 值的电路不会动作. 又因为第 j 行首列处理单元 $PE_{0,j}$ 的部分和寄存器 PSReg 被 $En_{1,j}$ 禁止, 从而保持着旧的累加和, 因而后面单元的部分和累加电路也不会动作.

另一方面, 在充满和排空流水线时 PE 单元中会存在无意义操作. 事实上在完成一次全搜索的 $(32 + SX \times SY)$ 个时钟周期中, 对每一个 PE 单元来说仅有 $SX \times SY$ 个时钟周期处于有效计算状态, 对应于候选宏块的总数目. 为了避免其余 32 个时钟周期中的无意义操作, 使能信号 $En_{0,0}$ 只是在最初的 $SX \times SY$ 个时钟周期中有效.

与 SAD 计算的流水线相适应, 使能信号 En 的产生也是流水线的. 通过禁止 AD 值计算和累加避免了电路的不必要动作, 从而能够节省功耗.

4.2 提早跳出判断单元 EJOD

提早跳出判断需要执行以下三个操作, 即实时更新部分和的最小值 $dmin_j$, 更新函数值 $f(dmin_j)$, 以及实时比较部分和与函数值, 产生有效计算标志信号 vld_j . 其硬件实现如图 5 所示.

在增量比较法中, 使用一个加法器就能够方便

地实现函数 f . 阈值 th 由控制阵列的 RISC 处理器计算提供. 在比例比较法中, 函数 f 是一次乘法运算. 不过由于 k_j 值是预先优化设定的常数值, 变量与常数的相乘可以分解为简单的移位和加法运算, 因此复杂的乘法运算是可以避免的. 经过模拟优化, 我们选择了 $k_0 = 3, k_1 = 2, k_2 = 1.5$, 对应的 f 实现如图 6 所示.

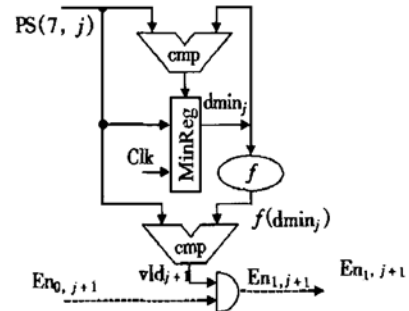


图 5 提早跳出判断单元 $EJOD_j$

FIG. 5 Judgement Cell $EJOD_j$ Ahead of Jump

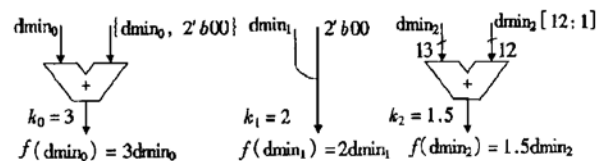


图 6 比例法中函数 f 的实现

FIG. 6 Archivement of Function f

由此可见, $EJOD$ 单元的实现代价大致为 2 个比较器和 1 个加法器. 这与整个处理阵列的硬件量相比是很小的.

5 模拟结果

本文对 5 种具有不同运动特点的视频序列进行了测试, 它们是 “bus”, “cheer girl”, “flower garden”, “football”, 和 “mobile calendar”, 所有这些序列均为隔行帧图, 30 帧/秒, 图像尺寸为 720×480 行, 亮色度格式为 $4:2:0$. 采用 MPEG2 标准^[12], 基于 MP@ML 和场预测模式进行编码, 相应参数为, $N = 15, M = 3$ (即一个图组 15 帧图像, 但第一个图组只有 13 帧图像, 在两个 P 或 I 帧之间存在 2 帧 B 图), 编码速率为 5Mbit/s. 规定望远镜单步搜索范围为 32×17 , 实际由于进行了 $4:1$ 的子采样, 在进行 2 像素分辨率的搜索时, 搜索窗大小为

16×9. 这里纵向距离是以帧图格式表示的, 表示为场图格式搜索窗大小为 16×5.

在上述参数条件下, 我们对各序列的两个图组共 28 帧图象进行模拟后的统计结果如表 1. 为进行对比, 我们也给出了多分辨率望远镜搜索算法的性能指标. 表中算法性能是以解码图的峰-峰信噪比 (PSNR) 来衡量的, 而运算量是参照 MRTlcS 算法的, 设定 MRTlcS 算法的运算量为 1 (全局搜索算法的运算量为 123). 我们把 AD 值运算、累加运算、比较运算都等效为基本加法运算, 而当 $dmin_j$ 确实被更新时, 计算 $f(dmin_j)$ 也视为一次加法运算. 对照表 1, 采用了新提早跳出技术后的算法运算量仅为传统 MRTlcS 算法的 50%—60% 左右, 而平均的 PSNR 衰减在 0.1dB 以内.

表 1 本文算法与 MRTlcS 算法的性能和计算复杂性比较
Table 1 Comparison of Performance and Calculating Complexity Between Algorithm of the Paper and One of MKTlcS

视频序列	新 EJO 技术	PSNR(dB)		运算量
		本文算法	MRTlcS	
bus	M1	31.23	31.31	0.60
	M2	31.20		0.53
cheer girl	M1	29.65	29.67	0.59
	M2	29.64		0.50
flower garden	M1	29.88	29.91	0.58
	M2	29.82		0.50
football	M1	36.83	36.88	0.64
	M2	36.86		0.63
calendar	M1	27.08	27.11	0.59
	M2	27.06		0.51
average	M1	30.93	30.98	0.60
	M2	30.92		0.53

M1: 比例比较法; M2: 增量比较法

基于上述的算法模拟结果, 我们对改进结构和传统结构的功耗作一个简单的比较估计. 设功耗与运算操作的次数即算法要求的运算量成正比. 改进结构的功耗为传统结构的 $53\% \times (16 \times 5) / (16 \times 5 + 32) = 38\%$ (增量比较法), 或 $60\% \times (16 \times 5) / (16 \times 5 + 32) = 43\%$ (比例比较法). 其中 16×5 是不考虑提早跳出时改进阵列中每一个处理单元 PE 的实际运算时钟周期数目.

6 结论

在基于多分辨率望远镜搜索的块匹配运动估值中, 本文针对多步的局部全搜索提出了一种新的提早跳出技术. 它与现有的提早跳出技术相比最突出的优点是不会破坏搜索算法的并行性. 本文以较小的硬件代价将新的提早跳出技术应用于传统的心动阵列结构搜索引擎中. 基于算法模拟结果的简单估计表明: 改进的运动估值器只消耗原来 40% 左右的功耗, 而且不会牺牲处理速度. 视频解码图的平均 PSNR 衰减在 0.1dB 以内.

参考文献

- [1] T. Nishitani, IEICE Trans. Fundamentals, 1999, **E82-A**(2): 184—196.
- [2] J. R. JAIN and A. K. JAIN, IEEE Trans. Communications, 1981, **COM-29**: 1799—1808.
- [3] R. Li, B. Zeng and M. L. Liou, IEEE Trans. Circuits and Systems for Video Technology, 1994, **4**(4): 438—442.
- [4] W. Li and E. Salari, IEEE Trans. Image Processing, 1995, **4**(1): 105—107.
- [5] H. C. Huang and Y. P. Hung, Graphical Models and Image Processing, 1997, **59**(6): 388—394.
- [6] V. L. Do and K. Y. Yun, IEEE Trans. Circuits and Systems for Video Technology, 1998, **8**(4): 393—398.
- [7] Y. S. Jehng, L. G. Chen and T. D. Chiueh, IEEE Trans. Signal Processing, 1993, **41**(2): 889—899.
- [8] K. Suguri, T. Minami, H. Matsuda *et al.*, IEEE Trans. Solid-State Circuits, 1996, **31**(11): 1733—1741.
- [9] K. Matsuda, T. Tsuda, T. Ito *et al.*, IEEE Int. Communications Conf., North-Holland, 1984, 234—237.
- [10] T. Kondo, K. Suguri, M. Ikeda *et al.*, IEEE Micro, 1996, 51—58.
- [11] M. Ikeda, T. Kondo, K. Nitta *et al.*, IEEE Micro, 1999, 56—65.
- [12] ISO/IEC 13818-2, Information Technology—Generic Coding of Moving Pictures and Associated Audio ISO/IEC 13818-2 International Standard (Video), Nov., 1994.

A Low-Power VLSI Architecture for Telescopic Search Block-Matching Motion Estimation

ZHANG Wu-jian¹, QIU Xiao-hai¹, ZHOU Run-de¹, CHEN Hong-yi¹,
Kondo Toshio², Nakashima Takayoshi² and Ishitani Tsunehachi²

(1 *Institute of Microelectronics, Tsinghua University, Beijing 100084, China*)

(2 *NTT Electronics Corporation, Ebina, Japan*)

Abstract: An architectural enhancement is presented to reduce the power consumption of the telescopic search block-matching motion estimation based on a novel early-jump-out technique for computing block-matching errors. Augmenting it to the conventional systolic-architecture-based VLSI engine for the telescopic search can avoid unnecessary computations. A simple estimation based on the algorithm simulation results shows that, our enhanced architecture consumes 40% as much power as the conventional architecture. Meanwhile, the same throughput and a similar image quality are maintained.

Key words: telescopic search; motion estimation; low power; VLSI architecture; early jump out

EEACC: 6140C

Article ID: 0253-4177(2001)05-0670-07

ZHANG Wu-jian male, Ph. D candidate. He is engaged in the reaserch on video compression.

ZHOU Run-de male, professor. He is engaged in the reaserch on DSP, low power VLSI design.

Received 15 April 2000, revised manuscript received 14 July 2000

©2001 The Chinese Institute of Electronics