



(12) 发明专利

(10) 授权公告号 CN 101951521 B

(45) 授权公告日 2012. 08. 22

(21) 申请号 201010525777. 1

金涛 等. 一种可变块运动估计结构的优化设计. 《微型电脑应用》. 2008, 第 24 卷 (第 2 期), 4-7.

(22) 申请日 2010. 10. 30

(73) 专利权人 上海交通大学

地址 200240 上海市闵行区东川路 800 号

审查员 左恬源

(72) 发明人 陈伟伟 何卫锋 毛志刚

(74) 专利代理机构 上海科盛知识产权代理有限公司 31225

代理人 赵志远

(51) Int. Cl.

H04N 7/26 (2006. 01)

(56) 对比文件

CN 101505427 A, 2009. 08. 12, 全文.

US 2009/0310875 A1, 2009. 12. 17, 全文.

曹伟, 毛志刚. 全搜索运动估计算法的高效 VLSI 结构设计. 《固体电子学研究进展》. 2006, 第 26 卷 (第 2 期), 220-224.

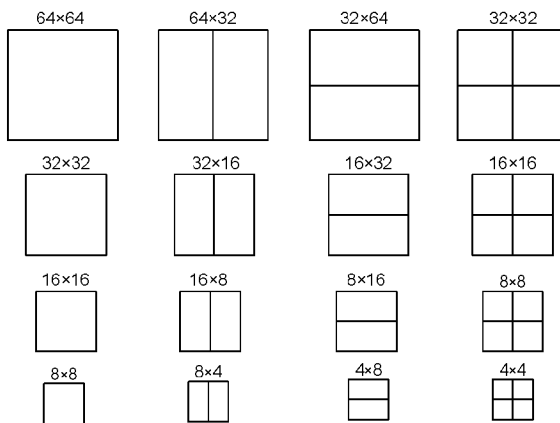
权利要求书 1 页 说明书 7 页 附图 5 页

(54) 发明名称

针对扩展可变块的视频图像运动估计方法

(57) 摘要

一种数字视频技术领域的针对扩展可变块的视频图像运动估计方法, 通过将正方形宏块划分为若干基本块条带进行 SAD 值的计算; 将第一步得到的所有基本块条带的 SAD 值与该基本块条带所属的同一个宏块中的其他基本块条带的 SAD 利用加法树结构进行叠加, 得到正方形宏块的 SAD 值; 利用正方形宏块的 SAD 值得到最小 SAD 值, 即可得到最佳匹配块和其对应的运动向量, 得到匹配块的信息, 从而对其进行编码, 完成运动估计。本发明通过 SAD 模块对输入的搜索区域的基本块条带及参考块的基本块条带的数据的运算, 再将得到的基本块条带的 SAD 值输入 SAD 值叠加模块, 得到大的宏块的 SAD 值。



1. 一种针对扩展可变块的视频图像运动估计方法,其特征在于,包括以下步骤:
 - 第一步、将正方形宏块划分为若干基本块条带进行 SAD 值的计算;
 - 第二步、将第一步得到的每一基本块条带的 SAD 值与该基本块条带所属的同一个正方形宏块中的其他基本块条带的 SAD 利用加法树结构进行叠加,得到正方形宏块的 SAD 值;
 - 第三步、利用正方形宏块的 SAD 值得到最小 SAD 值,即可得到最佳匹配块和其对应的运动向量,得到最佳匹配块的信息,从而对其进行编码,完成运动估计;所述的正方形宏块的尺寸为 64 像素 × 64 像素;
所述的基本块条带是指:宏块被划分为共 16 列,每列尺寸为 64 × 4 像素的图像。
2. 根据权利要求 1 所述的针对扩展可变块的视频图像运动估计方法,其特征是,所述的 SAD 值是指:
$$SAD(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|$$
,其中: $f_k(m, n)$ 为当前参考块的像素值, $f_{k-1}(m+i, n+j)$ 为相对于当前参考块偏移为 (i, j) 的候选块的像素值,即搜索区域的候选块的像素值。
3. 根据权利要求 1 所述的针对扩展可变块的视频图像运动估计方法,其特征是,所述的加法树是指:将多个需要叠加的值分组两两相加后,再将得到的和两两相加,直至得到最后的结果。

针对扩展可变块的视频图像运动估计方法

技术领域

[0001] 本发明涉及的是一种数字视频处理技术领域的设计,具体是一种针对扩展可变块的视频图像运动估计方法。

背景技术

[0002] 数字视频传输的图像数据中往往存在着大量的冗余数据,如果可以将那些冗余数据除去,从而可以大大减少数据的传输量。运动估计 (Motion Estimation, 简称 ME) 通过消除运动图像中时间域的相关性,计算两帧之间的参考帧的绝对差值 (the Sum of Absolute Difference, 简称 SAD), 由于两帧之间的差可以用比帧内编码少得多的比特数来编码,从而达到压缩图像的目的。运动补偿 (Motion Compensation, 简称 MC) 利用先前图像和运动估计的结果来预测、补偿当前的图像。

[0003] 由于视频图像是位于时间轴区间内的一组连续画面,因此相邻帧间的变化量一般很小,仅是运动物体的空间位置略有移动。因此运动估计和运动补偿技术是去除大量冗余的有效方法。

[0004] 运动估计作为视频编码器的核心部件,也是视频压缩中的最复杂的部分,对视频图像的编码压缩质量起到决定性的作用。在运动估计的研究中,人们针对不同应用提出了多种不同的方案;在这些方案中,最为经典且具有实用性的为全搜索块匹配 (Full Search Block Matching Algorithm, 简称 FSBM)。

[0005] 全搜索块匹配 (Full Search Block Matching Algorithm, 简称 FSBM) 是指,将图像中的一帧的搜索区域划分成许多互不重叠的 16×16 像素宏块 (Macro-Block, 简称 MB) 之后,每一宏块与相邻帧搜索区域的所有块的像素值进行比较匹配来得到最佳匹配块,即具有与当前块比较匹配后拥有最小 SAD 值的块。所得的匹配块与当前块的相对位移称为运动矢量 (Motion Vector, 简称 MV)。视频压缩编码时,仅对运动矢量即可。

[0006] 基于宏块的全搜索块匹配运动估计结构简单,易于硬件实现,硬件电路结构整齐,并且可重复性高,易于扩展,并能够得到全局最优的运动矢量,因此被广泛应用于各种视频编码标准中。该中运动估计方法的最大缺点是运算量较大。

[0007] 在基于宏块的运动估计的基础上,人们又引入的基于可变块的运动估计。与基于宏块的运动估计不同,在可变块运动估计中,进行运动估计的块除了包括 16×16 像素的宏块外,还包括 4×4 、 4×8 、 8×4 、 8×8 、 8×16 、 16×8 等不同大小的块。通过采用不同大小的块进行运动估计,可以有效地压缩视频图像的时间冗余,因而可变块运动估计被广泛应用到 H. 264、AVS 等视频压缩标准中。

[0008] 在当前正在制定的 High Performance Coding 标准中,人们提出了扩展可变块运动估计,进行运动估计的块包括 4×4 、 4×8 、 8×4 、 8×8 、 8×16 、 16×8 、 16×16 、 16×32 、 32×16 、 32×32 、 32×64 、 64×32 、 64×64 像素这些大小不同的块。如图 1 所示。

[0009] 扩展可变块运动估计的最大块的大小增加为 64×64 像素,但由于采用可变块的运动估计,既有如 64×64 这样较大宏块 SAD 计算,又有如 4×4 等小的块的 SAD 计算,因此采

用扩展可变块的运动估计后视频压缩质量会有较大提高。但引入扩展可变块运动估计后,进行 SAD 运算的最大块的大小变大且进行 SAD 运算的块的种类增多,会导致计算复杂度增加,硬件电路消耗变大。

发明内容

[0010] 本发明针对现有技术存在的上述不足,提供一种针对扩展可变块的视频图像运动估计方法,将参考块划分为 64×4 的基本块条带,再利用对参考块和搜索区域的数据流的组织,以及搜索窗和扫描窗的特殊计算顺序,并利用加法树结构进行 SAD 值的叠加,达到高效进行运动估计的目的。

[0011] 本发明通过以下技术方案得以实现:本发明包括以下步骤:

[0012] 第一步、将正方形宏块划分为若干基本块条带进行 SAD 值的计算;

[0013] 所述的正方形宏块是指:运动估计中,宏块是指将视频图像中的每一帧划分为一定尺寸的图像,并对宏块大小的图像进行计算以求得最佳匹配块;

[0014] 所述的宏块的尺寸为 $64 \text{ 像素} \times 64 \text{ 像素}$;

[0015] 所述的基本块条带是指:宏块被划分为共 16 列,每列尺寸为 64×4 像素的图像。

所述的 SAD 值是指:
$$\text{SAD}(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|,$$

[0016] 其中: $f_k(m, n)$ 为当前参考块的像素值, $f_{k-1}(m+i, n+j)$ 为相对于当前参考块偏移为 (i, j) 的候选块的像素值,即搜索区域的候选块的像素值。

[0017] 第二步、将第一步得到的所有基本块条带的 SAD 值与该基本块条带所属的同一个宏块中的其他基本块条带的 SAD 利用加法树结构进行叠加,得到正方形宏块的 SAD 值。

[0018] 所述的加法树是指:将多个需要叠加的值分组两两相加后,再将得到的结构两两相加,直至得到最后的结果。

[0019] 第三步、利用正方形宏块的 SAD 值得到最小 SAD 值,即可得到最佳匹配块和其对应的运动向量,得到匹配块的信息,从而对其进行编码,完成运动估计。

[0020] 根据本发明所提出的运动估计方法,所对应的核心电路结构主要有 SAD 运算模块及对应的数据缓存模块,此外还有 SAD 值叠加模块。根据本发明所提出的针对扩展可变块的运动估计,SAD 运算模块的每一次运算仅需负责 64×4 的基本块条带的 SAD 运算,不需进行 64×64 像素的宏块的 SAD 值运算;数据缓存模块负责缓存搜索区域数据;参考块数据在电路结构内部不需缓存,直接从外部读入。并且根据本发明所提出的针对扩展可变块的运动估计方法,在一定的控制信号的控制下在数据缓存模块与外部之间、SAD 运算模块与缓存模块之间传递搜索区域数据,同时在一定的控制信号下控制参考块数据的读入。

[0021] 通过 SAD 模块对输入的搜索区域的基本块条带及参考块的基本块条带的数据的运算,再将得到的基本块条带的 SAD 值输入 SAD 值叠加模块,得到大的宏块的 SAD 值。

附图说明

[0022] 图 1 为扩展宏块运动估计的块大小示意图。

[0023] 图 2 为 64×64 的参考帧的划分示意图。

[0024] 图 3 为搜索窗示意图。

[0025] 图 4 为扫描窗示意图。

- [0026] 图 5 为搜索窗、扫描窗与参考块对应关系示意图。
 [0027] 图 6 为 SAD 值的累加示意图。
 [0028] 图 7 为扫描小块内扫描顺序对应图。
 [0029] 图 8 为第奇数次大扫描的计算顺序示意图。
 [0030] 图 9 为第偶数次大扫描的计算顺序示意图。
 [0031] 图 10 为第奇数次和偶数次大扫描叠加后的计算顺序示意图。

具体实施方式

[0032] 下面对本发明的实施例作详细说明,本实施例在以本发明技术方案为前提下进行实施,给出了详细的实施方式和具体的操作过程,但本发明的保护范围不限于下述的实施例。

[0033] 本实施例通过以下方式实现扩展可变块的视频图像运动估计:

[0034] (1) 参考块的定义及划分

[0035] 参考块:扩展宏块运动估计中宏块大小为 64×64 像素,称为参考块。

[0036] 基本块条带:将参考块均匀划分为 16 列,每列包含 64×4 像素,将其称为的基本块条带,如图 2 所示。其中,将 64×4 像素的基本块条带划分为 16 个 4×4 像素的基本块。

[0037] 这样,对于一个 64×64 像素的宏块来说,它包括 16 个基本块条带,每个基本块条带包括 16 个 4×4 像素的基本块。

[0038] (2) 搜索窗的定义及划分

[0039] 基于块的绝对差值 (SAD) 计算:基于块匹配的运动估计的基本思想是将视频图像序列中的每一帧都划分为大小相同的、互不重叠的 $M \times N$ 的小宏块,并在 M 、 N 均较小的前提下,假设宏块内的所有像素点只做平移运动,而无旋转、拉伸等运动。在一定的搜索范围内,按照一定规则计算当前帧的每一宏块和参考帧中的宏块的 SAD 值。基于块的 SAD 计算定义如下:

$$[0040] \quad \text{SAD}(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|$$

[0041] 其中, $f_k(m, n)$ 为当前参考块的像素值, $f_{k-1}(m+i, n+j)$ 为相对于当前参考块偏移为 (i, j) 的候选块的像素值。

[0042] 搜索区域:以当前参考块在当前帧的位置为坐标,在参考帧中找到相应的位置点,并以该点确定一个半径为 p 的搜索范围进行基于块的 SAD 计算。其中,以半径为 p 的 $[-p, p]$ 搜索范围,称之为搜索区域。

[0043] 搜索窗:在 $[-p, p]$ 的搜索区域中,每一个搜索点对应于一个块的 SAD 计算,因此总共包含 $(2p+1) \times (2p+1)$ 个宏块的 SAD 计算。以搜索点的坐标来表示这些需要进行块的 SAD 计算,由这些搜索点构成的区域称为搜索窗。对于 $[-p, p]$ 的搜索区域,其搜索窗大小为 $(2p+1) \times (2p+1)$ 。

[0044] 搜索小块:在搜索窗中,根据数据流组织需要,将搜索窗划分为 4×4 的小块,称为搜索小块。

[0045] 图 3 给出了以 $p = 16$ 为例的搜索窗的示意图,搜索窗大小为 32×32 ,划分为 64 个搜索小块。

[0046] (3) 扫描窗的定义及划分

[0047] 扫描窗:对于 $(2p+1) \times (2p+1)$ 大小的搜索窗,对于参考块中被划分为 64×4 像素的每一个基本块条带,对应为 $(2p+1) \times 64$ 的区域内的基本块条带的 SAD 值计算,称为扫描窗。扫描窗中,每一个点对应于一个基本块条带的 SAD 计算,以这些扫描点的坐标来表示需要进行 SAD 计算的基本块条带。

[0048] 扫描小块:与搜索小块类似,在扫描窗中,将扫描窗划分为 4×4 的小块,称为扫描小块。一个扫描小块内的 SAD 计算为 16 个不同的当前块内的基本块条带和同一个参考块内的基本块条带之间的计算。

[0049] 图 4 给出了以 $p = 16$ 为例的扫描窗的示意图,扫描窗大小为 32×64 ,划分为 128 个扫描小块。

[0050] 对于一个 $(2p+1) \times (2p+1)$ 大小的搜索窗,总共对应 $(2p+1)/4$ 个扫描窗。图 5 给出了搜索窗、扫描窗和参考块的对应关系。一个搜索小块的 64×64 像素 SAD 值计算对应一大行扫描小块的基本块条带 SAD 计算,一列搜索小块的 64×64 像素 SAD 值计算对应一个扫描窗的基本块条带 SAD 计算。

[0051] 以 $p = 16$ 为例,扫描窗的大小与搜索窗相对应为 32×64 ,划分为 $8 \times 16 = 128$ 个 4×4 的扫描小块。其中,每个扫描小块对应 16 个基本块条带的 SAD 值计算,一大行扫描小块对应一个搜索小块的 16 个 64×64 像素宏块的 SAD 值计算,一个扫描窗对应搜索窗中一大列搜索小块的 128 个宏块的 SAD 值计算。

[0052] 所述的 SAD 值通过以下方式得到:以搜索区域为 $[-16, 16]$,搜索窗大小为 32×32 ,参考块大小为 64×64 为例,具体介绍本发明所设计的数据流组织和 SAD 计算方法。

[0053] 将 64×64 的宏块划分为 64×4 像素的基本块条带进行运动估计,以基本块条带为单位进行 SAD 值计算,在数据初始化之后,对输入的搜索区域及相应的参考块数据进行 SAD 值的计算,每个时钟周期可以输出一个基本块条带的 SAD 值,16 个时钟周期则输出搜索窗中一个 4×4 的搜索小块的对应的 16 个基本块条带的值。然后依次计算接下去 15 个基本块条带的 SAD 值,共需要 $16 \times 16 = 256$ 个时钟周期。如图 6 所示。

[0054] 对于搜索窗内的某一点而言,每隔 16 个时钟周期将输出该点对应的 64×64 像素宏块中的一个基本块条带的 SAD 值。依次类推,经过 $16 \times 16 = 256$ 个时钟周期可以得到搜索窗内的一个搜索小块中的 16 个点所对应的 16 个 64×64 宏块的 SAD 值所需的所有基本块条带的 SAD 值,即经过 256 个时钟周期可以算出搜索窗内一个 4×4 搜索块内的 16 个 64×64 像素的 SAD 值。其中,基本块条带的 SAD 值由 4×4 的小块的 SAD 值利用加法树得到, 64×64 像素宏块的 SAD 值由基本块条带的 SAD 值叠加得到。

[0055] (1) 搜索窗内计算顺序

[0056] 其中,将 32×32 像素的搜索窗划分为 64 个搜索小块后。在搜索窗的小块级别需要满足一定的计算顺序,以和各列的基本块条带的扫描顺序相对应。因此,经过研究和设计,搜索窗内以 4×4 的块为单位的计算顺序如图 3 中标号所示。

[0057] 搜索窗内的搜索小块计算顺序为倒下的之字形扫描路径,两列之间的衔接为相邻的两个 4×4 小块之间的衔接,而非全部从上至下或全部从下至上;例如第一列从 1 号小块扫描至 8 号小块,不是跳回最上方的 16 号小块而是衔接到相邻的 9 号小块。这样的计算顺序设计能够提高数据复用率,减少数据的过大变动,减少数据的等待时间。

[0058] 以搜索窗内的各块为例,1号小块对应与扫描窗内的第一大行(指的是扫描窗内横向的16个扫描小块组成的一大行),具体地讲为扫描窗内第一大行从第一个扫描小块开始从左至右连续16个 4×4 的小块,即对应 $16\times 16=256$ 个基本块条带的SAD值;搜索窗内的2号小块则类似的对应扫描窗内第二大行,依次类推,搜索窗内的8号小块对应扫描窗内第八大行,并将搜索窗内1号至8号对应的扫描窗内的扫描完成记为第一次大扫描。

[0059] 9号小块则对应扫描窗中第八大行从第二个块开始从左至右连续16个 4×4 的小块,10号小块则对应第七大行,依次类推,并将搜索窗内9号至16号对应的扫描窗内的扫描完成记为第二次大扫描;17号小块对应第一大行从第三个块开始从左至右连续的16个 4×4 的小块.....

[0060] (2) 扫描窗内扫描小块级别计算顺序

[0061] 扫描窗同样划分为 4×4 像素的扫描小块,块中各点代表一个基本块条带的 64×4 像素的SAD值计算的当前块;扫描窗内从左至右依次16个 4×4 像素的小块记为扫描窗内的一大行。

[0062] 第一次大扫描:

[0063] 如上所述,以扫描窗中第一大行从第一块左起的16个扫描小块为例:其中,第一个扫描小块的SAD计算为该扫描小块中各点对应的16个当前基本块条带与参考块中第一大列基本块条带的SAD值;其中,第二个扫描小块计算为该小块中各点对应的16个当前基本块条带与参考块中第二大列基本块条带的SAD值;以此类推。

[0064] 经过第一大行的16个扫描小块的SAD值运算后,得到依次与搜索窗中的以1号搜索小块中各点对应的16个 64×64 像素的宏块的各16列基本块条带的SAD值,经过叠加后即得到搜索窗中1号搜索小块中各点对应的16个 64×64 像素的宏块的SAD值。

[0065] 接下来第二大行到第八大行对应关系依次类推。

[0066] 与搜索窗中计算顺序相对应的是,以扫描窗中的扫描小块为单位,则第奇数大行的扫描顺序为从左至右,第偶数大行扫描顺序为从右至左。搜索窗中完成这样一次八大行的SAD计算称之为一次大扫描,一次大扫描完成后得到对应搜索窗内的一列8个搜索小块对应的128个 64×64 宏块的SAD值;则第一次大扫描完成后得到对应搜索窗内的0号小块至8号小块对应的 $16\times 8=128$ 个 64×64 宏块的SAD值。

[0067] 第二次大扫描:

[0068] 接下来第二次大扫描则在扫描窗中整体右移一大列,一大行中扫描小块级别计算顺序依然为从左至右,从倒数第一行依次向上扫描。

[0069] 类似的,在倒数第一行从左至右的第一个扫描小块的SAD计算为该小块中各点对应的16个当前基本块条带与参考块中第一大列基本块条带的SAD值;其中,第二个小块计算对应于扫描小块中各点对应的当前基本块条带与参考块中第二大列基本块条带的SAD值;依次类推。

[0070] 经过倒数第一大行的16个小块的SAD值运算后,得到依次与搜索窗中的以9号搜索小块中各点对应的16个 64×64 像素的宏块的各16列基本块条带的SAD值,经过叠加后即得到搜索窗中9号搜索小块中各点对应的16个 64×64 像素的宏块的SAD值。

[0071] 接下来倒数第二大行到正数第一大行对应关系依次类推。

[0072] 与搜索窗中计算顺序相对应的是,以扫描窗中的扫描小块为单位,则正数第奇数

大行的扫描顺序为从右至左,第偶数大行扫描顺序为从左至右。第二次大扫描完成后得到对应搜索窗内的 9 号小块至 16 号小块对应的 128 个 64×64 宏块的 SAD 值。

[0073] 以扫描窗中的 16 个扫描小块组成的一大行为单位,第奇数次大扫描的计算顺序为从上到下,第偶数次大扫描为从下到上;且所有奇数次大扫描的计算顺序及读写数据等的规律均相同,所有偶数次大扫描的规律也均相同。

[0074] (3) 扫描窗内扫描小块内计算顺序

[0075] 扫描窗内 4×4 的小块的计算顺序大致分为 4 种模式,模式 1、模式 2、模式 3 和模式 4 如图 7 所示。模式 1、2 为以纵向为主的扫描模式,模式 3、4 为以横向为主的扫描模式。

[0076] 数据的扫描设计原则是提高数据复用,并用较少的带宽进行图像数据传输,减少等待时间。运算单元阵列进行 SAD 运算时,以列为单位输入数据,如果扫描顺序出现连续的横向扫描,则每个时钟周期都需要不断地有左右大量数据进入 ($64\text{pel} \times 8\text{bit}/\text{pel}$),数据输入量较大,则每个时钟都需要更新一列数据;如果扫描顺序出现连续的纵向扫描,则每个时钟周期都需要的数据仅为上下数据更新 ($4\text{pel} \times 8\text{bit}/\text{pel}$),四个时钟周期左右移动一列数据,数据输入量较小,则四个时钟仅需更新一列数据,所以可以将输入数据率减少为每个时钟周期 $1/4$ 列数据,可将输入数据带宽大大减少。所以,设计的基本原则是将扫描窗的 4×4 小块内的扫描顺序为保证连续的纵向移动最多,连续的横向移动最少,这样每个时钟的需要更新的数据最少。

[0077] 最终将每一次的大扫描总体的详细扫描顺序如图 8 所示,各个扫描小块内标号代表与图 7 的扫描模式相对应的计算顺序。其中,灰色块为计算起始块。

[0078] 如图 8 所示,在第奇数次大扫描中,经过设计以扫描小块为单位,每一大行最多仅有一个小块采用横向为主的扫描模式,且以横向为主的扫描方式仅可能出现在换行处。

[0079] 如图 9 所示,在第偶数次大扫描中,经过设计以扫描小块为单位,除了最后一行以外的其他行,与第奇数次大扫描类似,每一大行仅有一个以横向扫描为主的扫描模式,不会出现数据来不及更新的现象。但在最后进行扫描的最后一行,由于与第奇数次大扫描的衔接,会出现两个以横向扫描为主的扫描小块。

[0080] 在扫描的最后一行(即第一行)出现两个以横向扫描为主的小块,第一行最右侧的小块因为与第二行的最后一个小块的扫描存在较大的数据复用,同理不会出现数据准备来不及的现象。而第一行的第一个小块又是一次以横向扫描为主的小块,这一小块扫描时,会出现数据来不及准备的现象。这时,则需要等待一定的时钟周期,使数据准备好。经过扫描顺序的设计和研究,需要等待的时钟周期数为 7。

[0081] 为更清楚的看清两次大扫描之间的衔接,两次大扫描叠加后扫描如图 10 所示,每个 4×4 的小块内的第一个数字为第一次大扫描,第二个数字为第二次大扫描,数字 0 表示没有计算。

[0082] 根据以上扫描顺序,经过 8 次大扫描后,完成对一次对一个 64×64 的参考宏块的在 32×32 搜索窗的 SAD 值运算。

[0083] SAD 值叠加:

[0084] 为高效的得到各块的 SAD 值,本发明采用加法树的方案进行 SAD 值的叠加。首先,利用多级加法树将一个 4×4 的小块的 16 个像素的 SAD 值叠加得到小块整体的 SAD 值,然后再次利用多级加法树将 4×4 小块的 SAD 值叠加得到大块的 SAD 值,便完成 SAD 值的叠加

运算。

[0085] 本发明针对扩展可变块数据流的特点,综合考虑运动估计电路的硬件面积、输入端口数、内部缓存等各方面,独特的提出将宏块划分为 64×4 的基本块条带后进行运算和数据组织。

[0086] 通过对搜索窗和扫描窗内的计算顺序和 SAD 值的叠加进行了独特的设计,使得针对扩展块的运动估计能够有效地进行,并使得该运动估计方案利于 VLSI 实现;并在运算单元阵列大小仅为宏块大小的十六分之一,而传统的运动估计电路结构的运算单元阵列大小与宏块大小相同的情况下,保证同样优秀的性能;且通过对搜索区域数据流的设计,大大提高了数据复用率,提高数据运算效率。

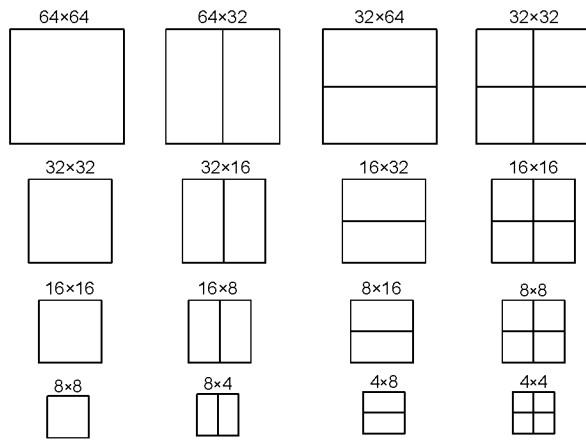


图 1

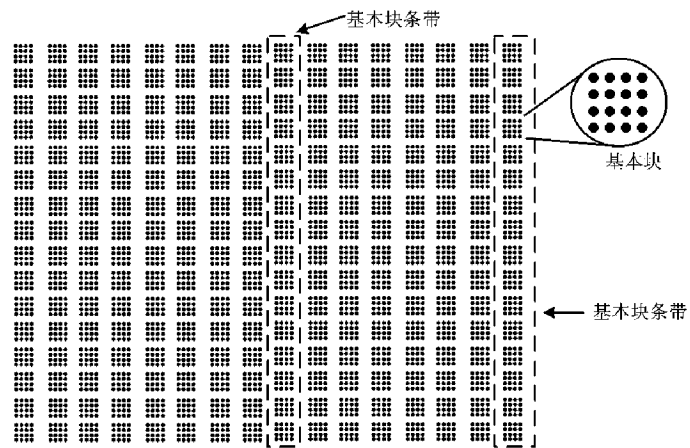


图 2

Figure 3 shows an 8x8 grid with numbers 1 through 64 arranged in a specific pattern. A callout box labeled "搜索小块" (Search Block) points to the top-left corner of the grid.

1	16	17	32	33	48	49	64
2	15	18	31	34	47	50	63
3	14	19	30	35	46	51	62
4	13	20	29	36	45	52	61
5	12	21	28	37	44	53	60
6	11	22	27	38	43	54	59
7	10	23	26	39	42	55	58
8	9	24	25	40	41	56	57

图 3

Figure 4 shows an 8x8 grid where every cell contains an asterisk (*). A callout box labeled "扫描小块" (Scan Block) points to the top-left corner of the grid.

*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*

图 4

搜索窗 1 号块

参考块

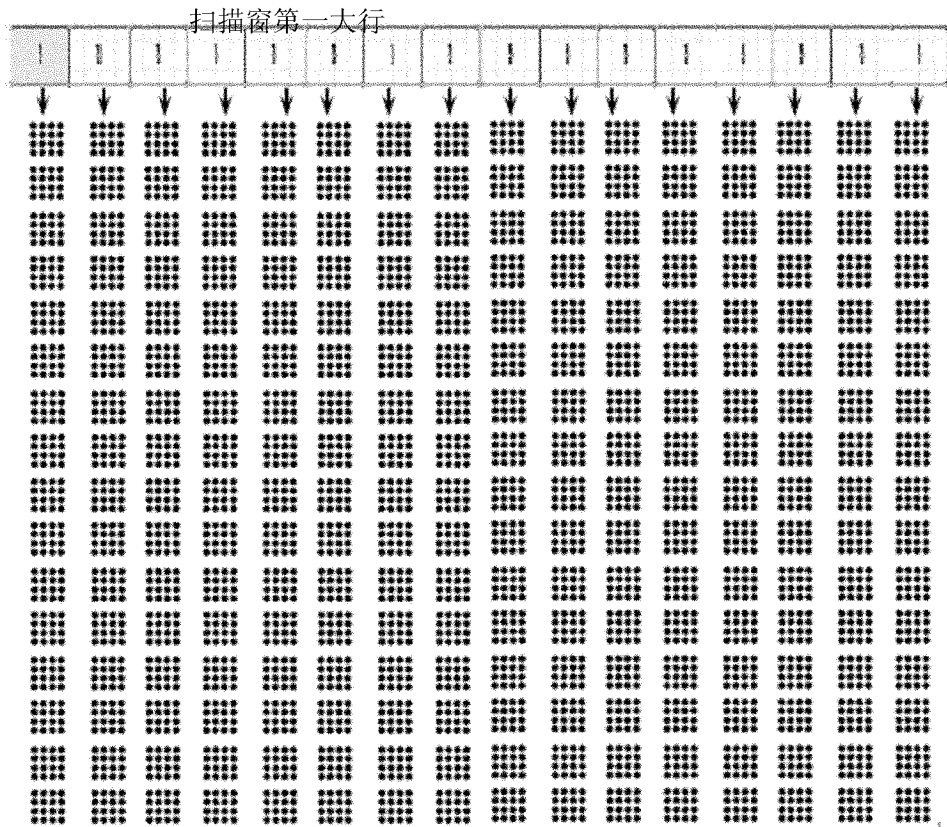


图 5

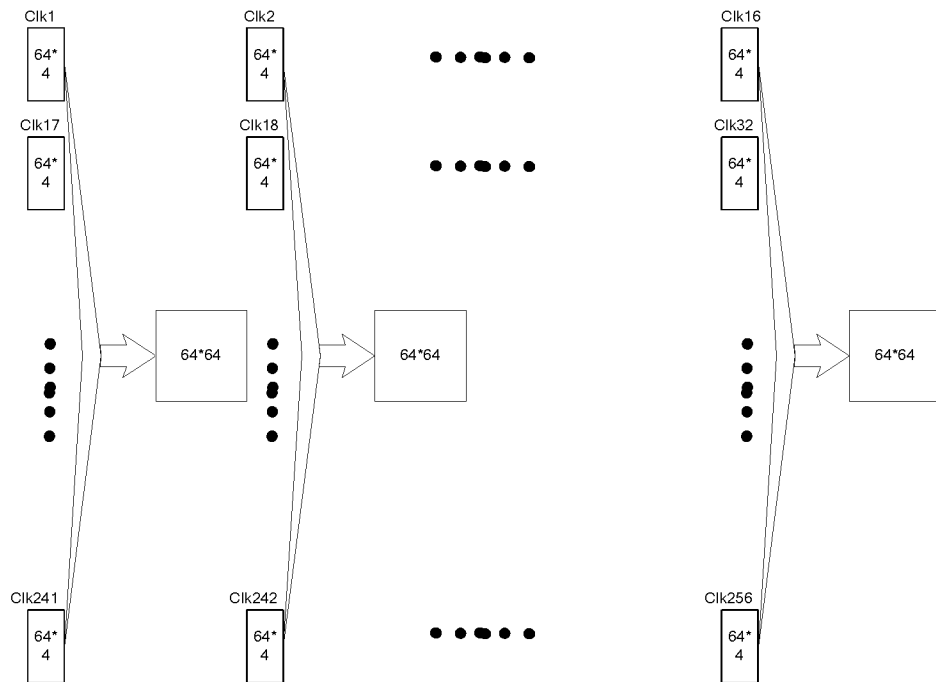
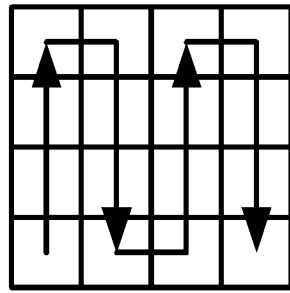
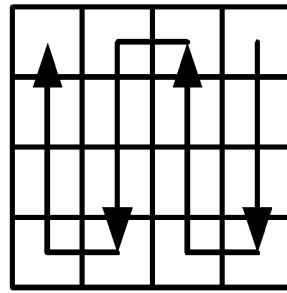


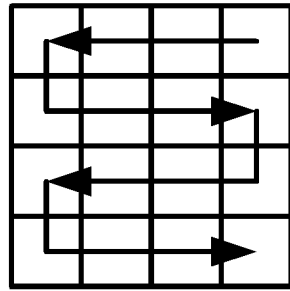
图 6



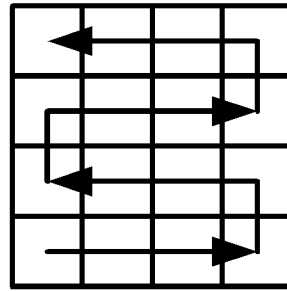
模式1



模式2



模式3



模式4

图 7

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

图 8

3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	4
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	4

图 9

1/0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0/4
3/0	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	0/4
3/0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0/4
3/0	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	0/4
3/0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0/4
3/0	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	0/4
3/0	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	1/2	0/4
3/0	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	2/1	0/4

图 10