

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

H04N 7/26 (2006.01)

H04N 7/32 (2006.01)



# [12] 发明专利申请公布说明书

[21] 申请号 200610064777.X

[43] 公开日 2007年12月19日

[11] 公开号 CN 101090491A

[22] 申请日 2006.12.15

[21] 申请号 200610064777.X

[30] 优先权

[32] 2006. 6. 16 [33] US [31] 11/455,009

[71] 申请人 香港科技大学

地址 中国香港九龙清水湾

[72] 发明人 区子廉 黄海明

[74] 专利代理机构 永新专利商标代理有限公司  
代理人 林锦辉

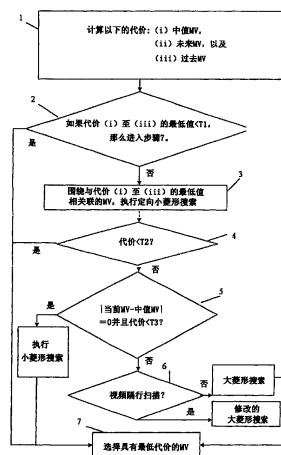
权利要求书 7 页 说明书 20 页 附图 4 页

## [54] 发明名称

用于视频压缩的增强的基于块的运动估计算法

## [57] 摘要

提出了一种方法、系统和软件，用于获得与第二图像(“参考图像”)的块相类似的第一图像的块。对第一图像的块顺序地进行处理，为每个块得到在第二图像中的多个候选位置，并且为每个候选位置评估代价函数。在第二图像中的每个候选位置是由来自第一图像的块的各自运动矢量来替换的。在本发明的第一方面中，所述代价函数是第一图像的未来块(即还没有被处理的第一图像的块)的预测运动矢量的函数。在本发明第二方面中，所述运动矢量由位置值给出，所述位置值并非全部是全像素空间、半像素空间、或 1/4 像素空间的。



1、一种为第一图像中一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括，根据对于每个候选位置的各自的代价函数的值，为所述第一图像的所述块的当前块，在所述第二图像中的多个候选位置之间进行选择，所述代价函数包括：

(A) 在所述第一图像中的当前像素块与在所述第二图像中的候选位置处的像素块之间的相似度测量，以及

(B) 与所述候选位置相对应的候选运动矢量、与所述当前块相关的预测运动矢量、以及在所述系列块中在所述当前块之后的所述图像的至少一个块的未来预测运动矢量的函数。

2、如权利要求 1 所述的方法，其中，所述函数包括第一项，其表示对所述候选运动矢量进行编码所必需的位数，以及第二项，其表示如果将在所述候选位置处的像素块选择作为所述第二图像的相似块，对所述未来预测运动矢量进行编码所必需的位数。

3、一种为按照光栅扫描顺序的第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括，为所述第一图像的所述块的当前块：

(i) 定义所述当前块的当前预测运动矢量；

(ii) 在所述第二图像中定义搜索区域作为所述第二图像中的多个候选位置，每个候选位置与各自的候选运动矢量 (MV) 相关联，所述候选运动矢量是所述第二图像中候选位置与所述第一图像中当前块的位置之间的相对位移；

(iii) 对于每个候选运动矢量以及对于在所述系列块中在所述当前块之后的所述第一图像的至少一个未来块：

(a) 定义所述未来块的未来预测运动矢量；以及

(b) 计算代价函数，其包括：

(A) 在所述第一图像中的当前块与在所述第二图像中的候

选位置处的块之间的相似度测量，以及

(B) 所述候选运动矢量、所述当前预测运动矢量、以及所述未来预测运动矢量的函数；以及

(iv) 选择与具有最低的所计算代价函数的候选位置相对应的相似块。

4、如权利要求 3 所述的方法，其中，所述函数包括第一项，其表示对所述候选运动矢量进行编码所必需的位数，以及第二项，其表示如果将在所述候选位置处的块选择作为所述相似像素块，对所述未来预测运动矢量进行编码所必需的位数。

5、如权利要求 4 所述的方法，其中，所述第二项表示对所述候选运动矢量与每一个所述未来预测运动矢量之间差异进行编码所需要的位数。

6、如权利要求 4 所述的方法，其中，所述函数是根据加权参数的值的所述第一项和所述第二项的加权和。

7、如权利要求 6 所述的方法，其包括得到所述加权参数的步骤。

8、如权利要求 3 所述的方法，其中，所述当前预测运动矢量是所述第一图像的至少一个块的运动矢量的函数，其中，对于所述至少一个块，已经预先得到了所述第二图像中的相似块，并且所述至少一个块在空间上邻近所述当前块。

9、如权利要求 3 所述的方法，其中，所述当前预测运动矢量是所述第一图像之外的一个帧中的与所述当前块相搭配的至少一个块的运动矢量的函数。

10、如权利要求 3 所述的方法，其中，所述当前预测运动矢量是

帧中除了与所述当前块空间上相邻的第一帧块之外的至少一个块的运动矢量的函数。

11、如权利要求 3 所述的方法，其中，在 (iii) 中的所述未来块是根据光栅扫描顺序的下一个块，并且在 (iii) (a) 中的所述未来预测运动矢量是所述候选运动矢量、右上方和右右上方的运动矢量的中值。

12、如权利要求 11 所述的方法，其中，当在步骤 (iii) 中确定所述候选运动矢量中的第一个的代价小于第一阈值时，不计算余下的所述候选运动矢量的代价函数，并且生成与所述第一候选运动矢量具有预定义的空间关系的一组第二候选运动矢量，并且在步骤 (iv) 之前，为所述第二候选运动矢量中的每一个计算代价函数。

13、如权利要求 12 所述的方法，其中，将所述第二候选运动矢量组选择为比所述第一候选运动矢量更接近所述当前预测运动矢量，其中，在矢量对之间距离的测量是根据对所述矢量之间差异进行编码所需要的位数来定义的。

14、如权利要求 3 所述的方法，其中，在步骤 (ii) 中，所述候选矢量中的至少一个是根据所述光栅扫描顺序的所述图像的未来块的运动矢量。

15、如权利要求 3 所述的方法，其中，在步骤 (ii) 中，所述候选矢量中的一个是使用基于使得所选择的候选矢量与所述当前预测运动矢量之间的距离最大化的准则，而从一组可能的候选矢量中选择出的。

16、如权利要求 3 所述的方法，其中，存在多个第二图像，所述多个候选位置包括每一个所述第二图像中的至少一个候选位置。

17、如权利要求3所述的方法，其中，所述块是所述第一图像的较大区域的子块，所述子块是根据由编码数定义的编码顺序来编码的。

18、一种用于为第一图像中的一系列像素块中的每一个选择第二图像的各自相似的像素块的系统，所述系统包括处理器，所述处理器配置成顺序地处理所述第一图像的块，并且根据代价函数，为所述第一图像的所述系列块中的每个当前块，在所述第二图像中的多个候选位置之间进行选择，所述代价函数包括：

(A) 在所述第一图像中的当前像素块与在所述第二图像中的候选位置处的像素块之间的相似度测量，以及

(B) 与所述候选位置相对应的候选运动矢量、与所述当前块相关的预测运动矢量、以及在所述系列块中在所述当前块之后的所述图像的至少一个块的未来预测运动矢量的函数。

19、一种计算机系统可读的软件产品，其使得所述计算机系统为第一图像中的一系列像素块中的每一个选择第二图像的各自相似的像素块，这是通过根据代价函数，为所述第一图像的所述系列块中的当前块而在所述第二图像中的多个候选位置之间进行选择来实现的，所述代价函数包括：

(A) 在所述第一图像中的当前像素块与在所述第二图像中的候选位置处的像素块之间的相似度测量，以及

(B) 与所述候选位置相对应的候选运动矢量、与所述当前块相关的预测运动矢量、以及在所述系列块中在所述当前块之后的所述图像的至少一个块的未来预测运动矢量的函数。

20、一种为按照光栅扫描顺序的第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括为所述第一图像的所述块中的当前块，在所述第二图像中的多个候选位置

之间根据每个候选位置的各自的代价函数的值来进行选择，所述候选矢量中的至少一个是按照光栅扫描顺序的所述图像的未来块的预测运动矢量。

21、一种为第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括为所述第一图像的所述块中的当前块，在所述第二图像中的多个候选位置之间根据每个候选位置的各自的代价函数的值来进行选择，其中，所述候选矢量中的至少一个是使用基于使得所选择的候选矢量与所述块中的所述当前块的当前预测运动矢量之间的距离最大化的准则，而从一组可能的候选矢量中选择出的。

22、一种为第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括为所述第一图像的所述块中的当前块，在所述第二图像中的多个候选位置之间根据每个候选位置的各自的代价函数的值来进行选择，其中，所述第二候选运动矢量组是通过以下过程来选择的，所述过程包括：

定义所述块中的所述当前块的当前预测运动；

定义第一候选运动矢量，以及

定义一组第二候选运动矢量，其比所述第一候选运动矢量更接近所述当前预测运动矢量，其中，在矢量对之间距离的测量是根据对所述矢量之间差异进行编码所需要的位数来定义的。

23、一种为第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括：为所述第一图像的所述系列块中当前块，在所述第二图像中的多个候选位置之间进行选择，所述多个候选位置包括由在各自横向方向上的两个位置值所定义的一组候选位置，所述方向中的至少一个的位置值为使得其无法全部由  $Lm$  表示，其中， $m$  是整数， $L$  是常量。

24、如权利要求 23 所述的方法，其中，所述候选位置组中的每一个通过相应位置值与各自的运动矢量相关联，所述运动矢量不同于预测运动矢量。

25、如权利要求 23 所述的方法，其中，所述候选位置组中的每一个通过相应位置值与各自的运动矢量相关联，所述运动矢量不同于 (0,0) 运动矢量。

26、一种为第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的方法，所述方法包括：为所述第一图像的所述系列块中的当前块，在所述第二图像中的多个候选位置之间进行选择，所述多个所述候选位置包括由两个位置值所定义的一组候选位置，所述候选位置能够写作  $DA_m m / n$ ，其中  $m = -\dots, 2, -1, 0, 1, 2\dots$ ， $n$  是 1 或 2 的幂， $D$  是常量，而对于  $m$  的至少一个的值而言  $A_m$  是小于 1 的值，并且对于  $m$  的所有值而言其是在 0.75 至 1.0 的范围内。

27、如权利要求 26 所述的方法，其中，所述候选位置组中的每一个通过相应位置值与各自的运动矢量相关联，所述运动矢量不同于预测运动矢量。

28、如权利要求 26 所述的方法，其中，所述候选位置组中的每一个通过相应位置值与各自的运动矢量相关联，所述运动矢量不同于 (0,0) 运动矢量。

29、如权利要求 26 所述的方法，其中，对于  $m$  的所有值而言， $A_m$  的值至少为 0.85。

30、如权利要求 26 所述的方法，其中，对于  $m < -1$  以及  $m > 1$  而言， $A_m$  等于 1。

31、一种为第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的系统，所述系统包括处理器，所述处理器配置为，为所述第一图像的所述系列块中当前块，在所述第二图像中的多个候选位置之间进行选择，所述多个候选位置包括由在各自横向方向上的两个位置值所定义的一组候选位置，所述方向中的至少一个的位置值为使得其无法全部由  $Lm$  表示，其中， $m$  是整数， $L$  是常量。

32、一种计算机系统可读的软件产品，其使得所述计算机系统为第一图像中的一系列像素块中的每一个选择第二图像的各自的相似像素块的系统，这是通过在所述第二图像中的多个候选位置之间进行选择来实现的，所述多个候选位置包括由在各自横向方向上的两个位置值所定义的一组候选位置，所述方向中的至少一个的位置值为使得其无法全部由  $Lm$  表示，其中， $m$  是整数， $L$  是常量。



## 用于视频压缩的增强的基于块的运动估计算法

### 相关申请

本申请请求了于2005年6月17日申请的美国临时专利申请60/691,181的优先权，在此将其整体引入作为参考。

### 技术领域

本发明通常涉及用于数字信号压缩、编码和表示的方法和系统，并且更加确切地说，本发明涉及使用多帧运动估计（ME）的方法和系统。本发明进一步涉及一种计算机程序产品，诸如记录媒体，承载可以由计算设备读取的程序指令，以使得所述计算设备执行根据本发明的一种方法。

### 背景技术

由于由现代多媒体应用使用的原始数字视频数据（或者图像序列）的巨大尺寸，必须对这种数据进行压缩以便可以传输和存储这些数据。存在许多重要视频压缩标准，包括ISO/IEC MPEG-1、MPEG-2、MPEG-4标准和ITU-T H.261、H.263、H.264标准。ISO/IEC MPEG-1/2/4标准广泛地运用于娱乐业以发行电影、包括视频压缩光盘或者VCD（MPEG-1）的数字视频广播、数字视频光盘或者数字多用途光盘或者DVD（MPEG-2）、可记录DVD（MPEG-2）、数字视频广播、数字视频广播或者DVB（MPEG-2）、视频点播或者VOD（MPEG-2）、在US中的高清晰度电视或者HDTV（MPEG-2）等等。MPEG-4标准比MPEG-2更加先进，可以在较低的比特率下实现高质量视频，这使得其非常适合于因特网、数字无线网络（例如3G网络）、多媒体信息服务（来自3GPP的MMS标准）等等上的视频信息流。MPEG-4被下一代高清晰度DVD（HD-DVD）标准和MMS标准所接受。ITU-T H.261/3/4标准设计用于低延迟电视电话和视频会议系统。早

期的 H.261 标准设计为在  $p \cdot 64\text{ kbit/s}$  下工作,  $p = 1, 2, \dots, 31$ 。后期的 H.263 标准非常成功, 被广泛地用于现代电视会议系统, 并且用于宽带网络和无线网络中的视频信息流, 其中无线网络包括在 2.5G 和 3G 网络以及其他网络中的多媒体信息服务 (MMS)。最新标准, H.264 (也称作 MPEG-4 版本 10, 或者 MPEG-4 AVC) 是当前最新技术水平的视频压缩标准。它如此强大以至 MPEG 决定与 Joint Video Team (JVT) 的框架中的 ITU - T 联合开发。新的标准在 ITU-T 中称为的 H.264, 并且被称作 MPEG-4 高级视频编码 (MPEG-4 AVC), 或者 MPEG-4 版本 10。H.264 用于 HD-DVD 标准、直接视频广播 (DVB) 标准并且可能用于 MMS 标准。基于 H.264, 当前在中国正在开发称作视听标准 (AVS) 的相关标准。AVS 1.0 设计用于高清晰度电视 (HDTV)。AVS-M 设计用于移动应用。H.264 具有超过 MPEG-1/2/4 及 H.261 / 3 标准的目标和主观的视频质量。除使用整数  $4 \times 4$  离散余弦变换 (DCT) 代替传统的  $8 \times 8$  DCT 以外, H.264 [1] 的基本编码算法类似于 H.263 或者 MPEG-4, 并且还有额外的特点, 包括 I 帧的帧间预测模式、用于运动估计 / 补偿的多种块大小以及多种参考坐标系、用于运动估计的四分之一像素精度、回路内去块效应 (in-loop deblocking) 滤波器, 内容自适应二进制算术编码 (context adaptive binary arithmetic coding), 等等。

运动估计是大多数视频压缩标准(诸如 MPEG-1/2/4 和 H.261/3/4) 的核心部分, 其充分利用时间冗余度, 因此其性能直接影响视频编码系统的压缩效率、主观视频质量以及编码速度。

在块匹配运动估计 (BMME) 中, 在 ME 中, 对当前块和参考块之间的失真的最通用的测量是绝对差值的和 (SAD), 对于一个  $N \times N$  块而言, 定义为

$$SAD(mvx, mvy) = \sum_{m=0, n=0}^{N-1} |F_t(x+m, y+n) - F_{t-1}(x+m+mvx, y+n+mvy)|$$

其中  $F_t$  是当前帧,  $F_{t-1}$  是标准帧,  $(mvx, mvy)$  表示当前运动矢量(MV)。

对于宽度 = X, 高度 = Y, 并且块大小 =  $N \times N$  的帧而言, 在搜索范

围  $\pm W$  中需要对 SAD 进行评估以查找最优运动矢量的搜索点总数等于：

$$\left(\frac{X}{N}\right)\left(\frac{Y}{N}\right)(2W+1)^2,$$

对于  $X = 352$ ,  $Y = 288$ ,  $N = 16$  并且  $W = 32$  的情况下, 其等于 1673100。这是在视频编码器中消耗巨大计算能力的巨大数值。已经提出了许多快速算法[2]-[9]来减少在 ME 中搜索点的数目, 例如三步搜索 (TSS) [11], 2D 对数搜索[12], 新三步搜索 (NTSS) [3], MVFAST [7], 以及 PMVFAST [2]。MVFAST 和 PMVFAST 显著地优于前三个算法, 因为它们使用中值运动矢量预测器作为搜索中心执行中心偏离 ME, 由此通过平滑运动矢量场降低了 MV 编码的位的数目。

PMVFAST 算法 (其是对 MVFAST 及其它快速算法的重要改进, 并且因而被 MPEG 标准[10]所接受) 最初考虑一组 MV 预测器, 包括中值、零点、左边、顶部、右上方的和先前的共位 (co-located) MV 预测器。图 1 举例说明了当前块、左块、顶部块、右上方块、右上方块、以及右块 (其是"未来块", 即, 在当前块之后被处理的块) 的位置。它计算每个预测的 SAD 代价。在后来的发展中, 对 PMVFAST 进行修改以计算 RD (速率失真) 代价[13]来替代使用以下代价函数的 SAD 代价:

$$J(m, \lambda_{motion}) = SAD(s, c(m)) + \lambda_{motion}(R(m-p)) \quad (1)$$

其中  $s$  是原始视频信号,  $c$  是参考视频信号,  $m$  是当前 MV,  $p$  是当前块的中值 MV 预测器,  $\lambda_{motion}$  是 Lagrange 乘法器,  $R(m-p)$  表示用于编码运动信息的位。在 PMVFAST 中的下一步骤是选择具有最小代价的 MV 预测器, 并且根据从 MV 预测器获得的最小代价的值来执行数量大菱形搜索 (diamond search) 或者小菱形搜索。

在定义当前视频编码标准中独立但是重要问题是使用亚像素预定矢量, 包括半像素、1/4 像素或者可能甚至 1/8 像素运动矢量, 其提供对运动的更加精确的描述, 并且可以提供整像素运动估计的大约 1dB 的 PSNR 增益。采用半像素精度, 运动矢量可以采用等间距位置

值, 诸如 0.0, 0.5, 1.0, 1.5, 2.0 等等。采用 1/4 像素精度, 运动矢量可以采用诸如 0.00, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00 等等之类的位置值。采用 1/8 像素精度, 运动矢量可以采用诸如 0.000, 0.125, 0.250, 0.375, 0.500, 0.625, 0.750, 0.875, 1.000, 1.125, 1.250, 1.375, 1.500, 1.625, 1.750, 1.875, 2.000 等等之类的位置值。

众所周知, 运动矢量分布趋向于中心偏移, 这意味着运动矢量趋向于非常地接近于(0,0)。在图 6(a)中示出了这种情况, 其示出为(0,0) MV 使用完全搜索 (FS) 算法的在 Foreman 序列中的运动矢量分布。此外, 如图 6(b)所示, 运动矢量分布还向中值预测器 (中值 MV) 偏移, 它是在图 1 中示出的运动矢量左块、顶部块和右上方块的中值。此外, 如图 6(c)所示, 运动矢量还向在当前帧中的相邻运动矢量 (leftMV, topMV, topRightMV) 和先前帧中所设置的运动矢量 (preMV) 偏移, 如图 6(d)所示。这些可以都是用于当前矢量的运动矢量的可考虑的预测器, 并且它们可以被用于 PMVFAST。

## 发明内容

本发明目的是提供用于运动估计的新的并且有益的技术, 其适用于数字信号压缩、编码和表达的方法和系统。

特别是, 本发明设法提供新的并且有益的有效运动估计技术, 其可以例如应用在 MPEG-1、MPEG-2、MPEG-4、H.261、H.263、H.264 或 AVS 或者其它相关视频编码标准中。

本发明第一方面基于这样的实现: PMVFAST 算法的运动估计尽管与在前技术相比确实具有优点, 但是其并不是最佳的。在原理上, 对于视频中的每个帧, 存在整体上使得整个帧的 RD 代价最小化的运动矢量场  $\{m_{i,j}, i = 0..M-1, j = 0..N-1\}$  :

$$total\_RD\_Cost = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [SAD(s_{i,j}, c(m_{i,j})) + \lambda_{i,j}(R(m_{i,j} - p_{i,j})))] \quad (2)$$

其中(i, j)表示在包含 M×N 个块的帧中的第(i, j)个块。对于固定  $Q_p$  (其是量化参数),  $\lambda_{i,j} = \lambda = \text{恒量}$ , 而且

$$p_{i,j} = median(m_{i,j-1}, m_{i-1,j}, m_{i-1,j+1}) \quad (3)$$

然而，考虑到整个帧的全部 RD 代价同时需要指数级的计算复杂性，这是不实际的。因而，PMVFAST 和其它已知算法每次仅仅考虑仅一个块的 RD 代价，而不是一帧中全部块。

特别是，MVFAST 或 PMVFAST 都没有考虑当导出相对于当前块的运动矢量时，这导致下一个块的在中值 MV 预测器中的变化。这能够影响整个运动矢量场的平滑。

一般地说，本发明的第一方面通过改善 PMVFAST 的代价定义和运动预测器候选的选择而提出了一种新的 ME 算法。特别是，对于第一图像的每个当前块（当前块可以是  $16 \times 16$ ， $16 \times 8$ ， $8 \times 16$ ， $8 \times 8$ ， $4 \times 8$ ， $8 \times 4$ ， $4 \times 4$  或其它矩形长度，甚至非矩形），根据一个代价函数来选择第二图像（参考图像）的类似块，所述代价函数包括两项：

- (i) 当前块与相似块的不相似测量（例如 SAD, SAE）的项，以及
- (ii) 作为至少是对第一图像的未来块的运动矢量的预测的函数的项。

特别是，所提出的算法通过包含当前中值 MV 预测器并且还包含未来（即至今未处理）的编码块的估计中值 MV 预测器，使得改善运动场平滑性成为可能。

本发明的许多变化是可能的。特别是，所述块可以具有任何大小和任何形状。

可以有多个第二图像（即多个参考值）并且所述搜索可以包括在所有第二图像中的候选位置。

此外，可以为一起构成所述第一图像中的较大区域的、并且采用由编码数字所定义的编码顺序进行编码的多个子块执行所述新的代价函数。这些子块无须具有相同大小或形状。

本发明具有另一方面，其可以与本发明第一方面组合或独立使用。一般地说，本发明第二方面提出当对第一图像（当前块可以是  $16 \times 16$ ， $16 \times 8$ ， $8 \times 16$ ， $8 \times 8$ ， $4 \times 8$ ， $8 \times 4$ ， $4 \times 4$ ，或其它矩形大小，甚至非矩形）的当前块进行编码时，使用所选出的、具有从与已知技术中所使用的值不同的一组值中选出的位置值（即两个轴线方向中的各自组件）的运动矢量进行编码。

考虑一个可能的运动矢量预测器： $(0, 0)$ 。而整像素的常规技术

允许运动矢量采用诸如-2.0, -1.0, 0, 1.0, 2.0 等的位置值, 本发明第二方面提出修改接近于所述预测器的一组可能位置值。对于最接近 0 的位置值 1.0, 我们可以使用另一位置值诸如 0.85, 从而使得可允许的位置值将会包括-2.0, -0.85, 0, 0.85, 2.0 等等。其优点在于, 在统计上, 运动矢量趋向接近于 0。并且, 因而通过选择更加接近于 0 的位置, 我们将更加接近真实运动矢量, 并且因而可以给出可以导致较高压缩效率的更好的运动补偿。

因而, 在本发明一个特定表达中, 可以选择为至少一个轴方向选择一组可能位置值, 从而使得它们无法全部写作  $Lm$ , 其中  $m = \dots, 2, -1, 0, 1, 2, \dots$ , 并且  $L$  是常量 (例如 1 个像素间隔,  $1/2$  像素间隔, 或  $1/4$  像素间隔); 即, 位置值是不均匀的。特别是, 可以选择为至少一个轴方向选择一组可能位置值, 从而使得它们无法全部写作  $m/n$ , 其中  $m = \dots, 2, -1, 0, 1, 2, \dots$  并且  $n$  是 1 或 2 的幂。

注意, 本发明的第二方面并不局限于从一组非均匀的空间位置值中选择位置值; 与传统的位置值组相比, 也不局限于仅仅选择最接近于零的两个位置值。作为示例, 在本发明的第二方面的另一个示例里, 位置值 2.0 可变成 1.9, 从而可允许的位置值将包括 -1.9, -0.85, 0, 0.85, 1.9 等。

因而, 在本发明第二方面的可替换的特定表达中, (为至少一个所述方向) 选择一组可能位置值, 以包括可以写作为  $LA_m m / n$  的一个或多个位置值, 其中  $m = \dots, 2, -1, 0, 1, 2, \dots$ ,  $n$  是 1 或 2 的幂,  $L$  是常量 (例如 1 像素间隔,  $1/2$  像素间隔, 或  $1/4$  像素间隔), 并且  $A_m$  是小于 1 但是至少是 0.75 的值 (对于不同的  $m$  值选择性地不同), 更加优选地至少是 0.80, 并且最优选地至少是 0.85。

我们已经发现  $A_m$  的最佳值取决于视频。

本发明第二方面的特定实施例的一个优点在于, 它们所产生的运动矢量可以采用与常规算法相同的格式码进行编码, 除了该位置值的常规码应当分别解释为该实施例所使用的可能位置值之外。例如, 如果由特定实施例所使用的位置值是 -1.9, -0.85, 0, 0.85, 1.9 等, 则位置值 1.0 的常规码应该解释为 0.85, 并且位置值 2.0 的常规码应该

解释为 1.9, 等。

根据本发明的第二方面的方法可以包括以下步骤: 定义搜索区域, 在所述搜索区域内定义多个候选位置, 所述多个候选位置包括由本发明第二方面的新的位置值所定义的一组多个位置。这些位置值是来自关键位置(例如, (0,0) 运动矢量位置, 预测运动矢量, 等)处的候选位置的相应位移值。对于每个候选运动矢量而言, 我们计算代价函数, 所述代价函数是第一图像中的当前块与第二图像中的所述候选运动矢量处的块之间的相似度测量(例如 SAD, SAE)的函数。可选地, 其还可以是以下运动矢量的函数: 所述候选运动矢量、当前预测运动矢量, 以及可选地, 如本发明第一方面, 一个或多个未来预测运动矢量。例如, 可选地, 如本发明第一方面, 可以给定该代价函数。

## 附图说明

现在将仅仅参考以下附图描述本发明实施例作为示例, 其中:

图 1 示出了当前块、左块、顶部块、右上方块、右上方块、和右块;

图 2 示出了 Diff 的典型分布, 其是  $|m_{i,j+1} - p_{i,j+1}|$  与  $|m_{i,j} - p_{i,j+1}|$  之间的差异;

图 3a 示出了如本发明第一实施例所使用的巨大菱形搜索的搜索方式;

图 3b 示出了如本发明第一实施例中所使用的修改的巨大菱形搜索的搜索方式;

图 4 示出了如本发明第一实施例中所使用的小菱形搜索的搜索方式;

图 5a 和 5b 比较 PMVFAST 和本发明第一实施例的 MV 域的平滑性;

图 6 示出了对当前帧中的 (a) (0,0) MV、(b) 中值 MV、(c) 相邻 MV 和先前帧中所配置的 MV、(d) 在先前帧中的右下方的 MV (PreBottomRightMV) 使用完全搜索 (FS) 算法得到的在 Foreman 序列中的运动矢量分布;

图 7 是本发明第一实施例的流程图。

### 具体实施方式

本发明第一实施例采用了 PMVFAST 算法的许多特征，但是是通过考虑几个邻近块而不是仅仅一个块来对 PMVFAST（及其它现有算法）加以改进的。等式（2）和（3）示出，当前块 MV 的选择直接影响邻近块的 RD 代价，所述邻近块包括右块（或第  $(i, j+1)$  块），左下方块（或第  $(i+1, j-1)$  块），以及下方块（或第  $(i+1, j)$  块）。这是因为，当前 MV 将会影响这些邻近块的所预测 MV，并且因此进而影响那些块的最佳运动矢量。这些是“未来”块，因为当处理当前块时，还没有对它们执行运动估计。我们无法与当前块同时地计算这些未来块的最佳运动矢量，因为我们将需要按照等式（2）同时计算整个帧中所有块的最佳运动矢量，这将会非常复杂。

作为替代，为了评定在右块或第  $(i, j+1)$  块上的当前块的当前 MV 的选择的蕴含式，我们可以为等式（1）的当前块的 RD 代价函数增加一项：

$$R(|m_{i,j+1}-p_{i,j+1}|) \quad (4)$$

其中， $m_{i,j+1}$  是基于当前块的当前 MV 的右块的最佳运动矢量，并且  $p_{i,j+1}$  是基于当前块的当前 MV 的右块的中值 MV 预测器，即

$$p_{i,j+1} = \text{median}(m_{i,j}, m_{i-1,j+1}, m_{i-1,j+2}) \quad (5)$$

然而，等式（4）中的  $m_{i,j+1}$  是未知的，因为右块（未来块）的运动估计还没有执行。然而，我们注意到， $|m_{i,j+1}-p_{i,j+1}|$  可以通过  $|m_{i,j}-p_{i,j+1}|$  很好地近似。让 Diff 为  $|m_{i,j+1}-p_{i,j+1}|-|m_{i,j}-p_{i,j+1}|$ 。我们对许多视频测试顺序进行实验并且研究 Diff 的分布。图 2 显示了 Foreman 序列的概率密度函数（概率分布函数），如图 2 所示，典型的结果严重地偏向零。这意味着这两个量在大部分情况中是基本相同的（在大约 70% 情况下相同，而在大约 23% 的情况下仅仅相差 1）。这意味着， $|m_{i,j}-p_{i,j+1}|$  是  $|m_{i,j+1}-p_{i,j+1}|$  的良好近似。因此， $R(|m_{i,j+1}-p_{i,j+1}|)$  可以由  $W \cdot R(|m_{i,j}-p_{i,j+1}|)$  来近似，其中  $W > 0$ 。同样，我们可以对左下方块和下方块添加附加项到代价中。



对于第 (i, j) 块, 让 *medianMV* 表示由等式 3 给出中值 MV 预测器。让 *FmedianMV* 表示由等式 (5) 给出的未来中值 MV 预测器 (用于右块的中值 MV 预测器)。从而, *FmedianMV* 是 MV 候选的函数。在此, 第一实施例被称为对于第 (i, j) 块的“增强预测运动矢量场自适应搜索技术”(E-PMVFAST)。该实施例的步骤如下。

实施例的各步骤如下, 示于图 7 中。

对任何候选 MV, 如下定义代价。

$$\text{cost}(MV) = SAD + \lambda * [w * R(MV - \text{medianMV}) + (1-w) * R(MV - F\text{medianMV})] \quad (6)$$

1. 计算三个运动矢量预测器的代价: (i) 中值 MV 预测器 (“*medianMV*”), (ii) 右块的估计运动矢量 (“*futureMV*”), 其定义为如下:

$$\text{futureMV} \equiv \text{median}(\text{TopMV}, \text{TopRightMV}, \text{TopRightRightMV})$$

和 (iii) 来自过去块 (“*pastMV*”) 的 MV 预测器, 其是先前的共位 MV (“*PreMV*”) 和远离于 *medianMV* 的先前右下方的 MV (“*PreBottomRightMV*”) 中的一个, 即

$$\text{pastMV} \equiv \arg \max_{MV \in \{\text{PreMV}, \text{PreBottomRightMV}\}} \{abs(MV - \text{medianMV})\}$$

注意, 项 (ii) 可以由用于另一邻近未来块 (诸如左下方的、底部, 和/或右下方的块) 的估计运动矢量来补充或代替。

还应当注意, 在项 (iii) 中, 先前右下方的 MV 可以由用于另一邻近块的先前 MV 预测器补充或替换。

注意, 项 (ii) 和 (iii) 形代价发明的独立方面。

如果以上任意一个 MV 预测器不可用 (例如在帧的边界), 则跳过该预测器。

2. 如果运动矢量预测器的最小代价小于阈值 T1, 则停止搜索并且转到步骤 7。否则, 选择具有最小代价的运动矢量作为 *currentMV* (当前 MV) 并且转到下一步骤。注意, 3 个运动矢量预测器的代价可以采用预定顺序来计算 (例如 *medianMV*, 接着是 *futureMV*, 接着

是  $\text{pastMV}$ )，并且在任何时刻，如果任何运动矢量预测器的代价小于特定阈值，则搜索可能停止并且转到步骤 7。

3. 围绕  $\text{currentMV}$  执行定向小菱形搜索的一次迭代。下面解释定向小菱形搜索的概念。

4. 如果最小代价小于阈值  $T_2$ ，则停止搜索并且转到步骤 7。否则，选择具有最小代价的运动矢量作为  $\text{currentMV}$  并且转到下一步骤。

5. 如果 ( $\text{currentMV} = \text{medianMV}$ ) 并且当前最小代价小于阈值  $T_3$ ，则执行小菱形搜索并且转到步骤 7。

6. 如果视频不是隔行扫描，则执行大菱形搜索，如图 3 (a) 所示；否则，执行如图 3 (b) 所示的修改的大菱形搜索。在这些步骤中的每个步骤中，对菱形的每一标志点评估代价函数。

7. 选择具有最小代价的  $\text{MV}$ 。

在我们的实验中，发现  $w$  的值大约 0.8 是有效。

现在解释定向小菱形搜索的步骤。假定， $\text{centerMV}$  是当前搜索中心，并且  $\text{MV}_1$ ， $\text{MV}_2$ ， $\text{MV}_3$  和  $\text{MV}_4$  是四个围绕搜索点，如图 4 所示。为每个  $\text{MV}_i$  计算  $R(\text{MV}_i - \text{medianMV})$ 。如果  $R(\text{MV}_i - \text{medianMV}) < R(\text{centerMV} - \text{medianMV})$ ，则计算  $\text{MV}_i$  的 SAD 和代价。否则，忽略该  $\text{MV}_i$ 。选择具有最低代价的  $\text{MV}$  作为  $\text{currentMV}$ 。注意，定向方块搜索的概念被认为是新的，并且构代价发明一个独立方面，其无须与使用  $\text{futureMV}$  的概念相结合来执行。

大菱形搜索和修改的大菱形搜索的步骤是相同的，但是所述搜索是对于图 3 (a) 和 3 (b) 中分别示出的点的所有集合而完成的。

我们现在考虑在本发明范围内的实施例的多个可能的变体。

首先，注意，对于不同的块，在代价函数中的加权系数  $w$  可以是不同的。此外，可选择地，对于不同的  $\text{MV}$  候选所述  $w$  可以是不同的。特别地是， $w$  的定义可以取决于诸如  $\text{MV}$  候选是否接近于  $\text{medianMV}$  和/或  $\text{futureMV}$ 、或  $\text{MV}$  候选的 X 轴分量或 Y 轴分量是否与所述  $\text{FmedianMV}$  的 X 轴分量或 Y 轴分量相同之类的情况。

此外，代价函数可以不限于等式 (6) 的形式。其可以是包括

失真测量项（例如 SAD、失真平方和（SSD）、平均偏差失真（MAD）、MSD 等）和考虑了对当前块和某些邻近块（例如右块、下方块、左下块等）的运动矢量进行编码所必需的位的项的任何函数。

此外，在步骤 1 中，futureMV 的定义不局限于以上步骤 1 给出的形式。对于 futureMV 的两个可能的替换定义是：

$$\text{futureMV} \equiv \text{median}(\text{leftMV}, \text{TopRightMV}, \text{TopRightRightMV})$$

$$\text{futureMV} \equiv \text{median}(\text{medianMV}, \text{TopRightMV}, \text{TopRightRightMV})$$

此外，在以上表达的步骤 1 中，pastMV 被选为在先前帧中的可能 MV（preMV 和 preBottomRightMV）的列表集合中距离 medianMV 最远的一个。然而，要考虑的 MV 列表可以包含两个以上可能的 MV（例如 preMV, preLeftMV, preRightMV, preTopMV, preTopLeftMV, preTopRightMV, preBottomMB, preBottomLeftMV, preBottomRightMV, 等）。此外，来自一个以上先前编码帧的 MV 可以包括在所述列表中（例如如果当前帧是帧 N，则所述列表可以包含帧 N - 1, N - 2, N-3, ...）。如果当前帧是 B 帧，则先前编码帧的列表可以包括未来 P 帧。

此外，在步骤 1 中，选择 pastMV 为距离参考 MV（在步骤 1 中的 medianMV）最远的可能的 MV。其它参考 MV 也是可以的，包括 leftMV、或 TopMV、或 TopRightMV、或某组合。从可能的 MV 的列表中进行选择的其它方法也是可能的。

在步骤 2 中，所述 3 个运动矢量预测器的代价是在按照某种预定义顺序中得到的。可能的预定义顺序包括：

- a) medianMV, 随后是 futureMV, 随后是 pastMV
- b) medianMV, 随后是 pastMV, 随后是 futureMV
- c) futureMV, 随后是 medianMV, 随后是 pastMV
- d) futureMV, 随后是 pastMV, 随后是 medianMV
- e) pastMV, 随后是 medianMV, 随后是 futureMV
- f) pastMV, 随后是 futureMV, 随后是 medianMV

此外，虽然如以上表达，在步骤 3 中执行了定向小菱形搜索的一

次迭代，可以应用一个以上的迭代。

### 模拟结果

我们现在给出实施例 E-PMVFAST 的模拟结果。将所述实施例嵌入到 H.264 参考软件 JM9.3[13]中，并且使用各种 QP、视频序列、分辨率和搜索范围对其进行模拟。表 1 (a-c) 和 2 (a-c) 示出一些典型的模拟结果。PSNR（峰值信号与噪声的比例）变化和 BR（比特率）变化是相对于完全搜索（FS）的 PSNR 和比特率的变化。模拟结果示出，所提出的 E-PMVFAST 的比特率和 PSNR 趋向于与完全搜索和 PMVFAST 类似，但是在大范围的视频序列和比特率上，E-PMVFAST 趋向于比 PMVFAST 快大约 40%。E-PMVFAST 的一个重要特征在于，其运动矢量场趋向于非常地平滑，从而使得运动矢量可以比其它快速移动估计算法更加准确地表示对象的移动。

在图 5(a) 以及 5(b) 中，左侧的图像示出（如短线）由 PMVFAST 算法获得的运动矢量场，而右侧的成像示出通过所述实施例获得的相同成像的运动矢量。E-PMVFAST 的运动矢量场显著地比 PMVFAST 的运动矢量场更加平滑，特别是在所圈出的区域中。对于将感知转换编码（perceptual trans-coding）、速率控制、多种块大小运动估计、多种标准帧运动估计等中的视频运动内容进行分类而言，平滑的运动场非常有用。

Foreman CIF								
FS			PMVFAST			E-PMVFAST		
QP	PSNR	BR	PSNR 改变	BR 改变	速度 提高	PSNR 改变	BR 改变	速度 提高
24	38.61	10457168	-0.01	1.12%	269.90	-0.01	0.73%	345.62
26	37.25	7498832	-0.02	1.13%	259.61	-0.02	1.17%	337.49
28	35.99	5397584	-0.02	1.53%	250.27	-0.03	1.21%	329.79
30	34.64	3804232	-0.02	1.27%	241.91	-0.04	1.20%	326.41
32	33.34	2730064	-0.02	1.19%	235.26	-0.03	1.22%	321.96
34	32.25	2044176	-0.05	1.57%	230.48	-0.05	1.38%	325.01

36	31.70	1529848	-0.07	1.20%	226.76	-0.09	1.28%	325.47
38	29.9	1184744	-0.05	1.75%	223.29	-0.06	0.99%	328.10
40	28.89	960056	-0.09	1.85%	220.98	-0.11	1.09%	334.25
42	27.79	794312	-0.14	1.71%	218.94	-0.16	1.74%	341.62
44	26.7	672224	-0.17	1.58%	217.42	-0.22	0.89%	351.22
平均			-0.06	1.45%	235.89	-0.07	1.17%	333.36

表格 1a - foreman CIF 序列的模拟结果

Coastguard CIF								
FS			PMVFAST			E-PMVFAST		
QP	PSNR	BR	PSNR 改变	BR 改变	速度提高	PSNR 改变	BR 改变	速度提高
24	37.32	19901336	0	0.02%	245.39	0	0.02%	328.81
26	35.77	15169632	-0.01	-0.03%	236.12	0	0.01%	319.07
28	34.35	11403728	-0.01	-0.07%	232.24	-0.01	-0.03%	308.17
30	32.8	8097840	0	0.07%	229.89	0	0.04%	301.18
32	31.34	5616880	0.01	0.23%	228.38	0	-0.04%	296.55
34	30.08	3913296	-0.01	-0.08%	227.20	-0.01	-0.23%	294.13
36	28.75	2546328	-0.02	0.03%	225.88	-0.01	-0.04%	292.18
38	27.55	1684056	-0.03	0.48%	225.13	-0.03	0.35%	293.95
40	26.55	1172448	-0.04	-0.01%	224.78	-0.04	0.15%	299.45
42	25.52	800328	-0.06	0.65%	225.22	-0.06	1.00%	308.85
44	24.54	572224	-0.07	-0.10%	225.47	-0.06	0.39%	321.38
平均			-0.02	0.11%	229.61	-0.02	0.15%	305.79

表格 1b - Coastguard CIF 序列的模拟结果

Hall CIF								
FS			PMVFAST			E-PMVFAST		
QP	PSNR	BR	PSNR 改变	BR 改变	速度提高	PSNR 改变	BR 改变	速度提高
24	39.63	6954760	-0.01	0.26%	424.55	-0.01	0.18%	700.53
26	38.6	4539912	-0.02	0.17%	397.68	-0.02	0.43%	704.18
28	37.58	2993752	-0.04	0.07%	370.17	-0.04	0.06%	686.71
30	36.36	2021712	-0.04	0.01%	340.96	-0.04	0.18%	674.18
32	35.17	1411344	-0.03	-0.86%	321.03	-0.05	-0.40%	655.08

34	33.94	957600	0.02	0.40%	301.76	-0.02	0.44%	659.53
36	32.68	677768	-0.04	0.12%	288.55	-0.06	-0.12%	642.68
38	31.4	493168	-0.05	-0.04%	276.97	-0.05	-0.38%	641.42
40	30.05	379240	0.05	-0.45%	263.07	0	-0.48%	625.73
42	28.67	279008	-0.03	-0.99%	251.93	-0.07	-0.84%	624.02
44	27.38	214544	-0.02	-1.32%	241.90	-0.04	-1.03%	635.41
平均			-0.02	-0.24%	316.23	-0.04	-0.18%	659.04

表格 1c—Hall CIF 序列的模拟结果

Foreman QCIF								
FS			PMVFAST			E-PMVFAST		
QP	PSNR	BR	PSNR 改变	BR 改 变	速度 提高	PSNR 改变	BR 改变	速度 提高
24	38.32	2816360	0	0.68%	260.10	-0.01	-0.05%	356.91
26	36.85	2122432	-0.02	0.28%	249.94	-0.02	-0.42%	343.24
28	35.47	1615528	-0.01	0.41%	242.29	-0.01	0.09%	329.89
30	34.01	1213792	-0.01	0.44%	235.40	-0.02	0.35%	322.04
32	32.61	918272	-0.04	0.57%	231.38	-0.01	-0.13%	316.56
34	31.39	710976	-0.04	0.65%	228.06	-0.05	0.05%	314.51
36	30.07	536152	-0.1	0.85%	225.33	-0.07	0.60%	311.05
38	28.81	416112	-0.04	0.31%	223.95	-0.1	-0.10%	310.02
40	27.65	328288	-0.08	0.74%	222.89	-0.08	-0.30%	311.37
42	26.5	259936	-0.12	-0.39%	221.19	-0.1	0.47%	315.72
44	25.36	207344	-0.18	1.43%	218.38	-0.25	-0.08%	321.03
平均			-0.06	0.54%	232.63	-0.07	0.04%	322.94

表格 2a - foreman QCIF 序列的模拟结果

Akiyo QCIF								
FS			PMVFAST			E-PMVFAST		
QP	PSNR	BR	PSNR 改变	BR 改 变	速度 提高	PSNR 改变	BR 改变	速度 提高
24	40.81	530832	0.01	0.01%	483.45	0.01	0.01%	1143.12
26	39.42	391936	0	-0.38%	428.54	0	-0.38%	1017.49
28	38	289096	0	-0.09%	389.95	-0.01	-0.25%	879.72

30	36.57	208488	-0.01	0.40%	352.74	-0.03	0.33%	779.84
32	35.03	153776	-0.04	0.83%	312.72	-0.02	1.01%	699.16
34	33.66	118440	0.03	-0.05%	287.94	0.02	0.58%	644.63
36	32.27	89056	0.01	1.11%	270.48	0.03	0.88%	621.79
38	31.09	70280	-0.01	-0.01%	254.10	-0.03	-0.31%	598.06
40	30.13	57200	0	0.92%	245.76	-0.06	1.58%	612.01
42	28.77	47648	-0.02	0.22%	241.09	-0.02	0.15%	629.43
44	27.68	40608	0.02	0.55%	240.05	0.01	-0.91%	669.52
平均			0.00	0.32%	318.80	-0.01	0.25%	754.07

表格 2b - Akiyo QCIF 序列的模拟结果

Coastguard QCIF								
FS			PMVFAST			E-PMVFAST		
QP	PSNR	BR	PSNR 改变	BR 改 变	速度 提高	PSNR 改变	BR 改变	速度 提高
24	37	4456208	0	0.15%	251.59	0	0.09%	346.68
26	35.4	3298352	0	0.12%	238.53	0	0.08%	330.67
28	33.92	2386152	-0.01	0.15%	234.60	-0.01	0.24%	316.97
30	32.33	1633736	-0.02	0.06%	232.80	0	0.01%	310.88
32	30.86	1110088	0	0.22%	231.60	0	-0.17%	309.50
34	29.62	770656	-0.01	-0.38%	230.98	-0.01	-0.32%	310.31
36	28.3	511000	-0.02	0.28%	230.30	-0.01	0.50%	312.49
38	27.06	354208	0.02	-0.47%	229.61	0.01	-0.57%	314.41
40	26.08	256480	-0.08	-0.72%	229.39	-0.08	-1.08%	317.67
42	25.02	179528	-0.06	2.16%	228.04	-0.07	0.45%	325.54
44	24	131856	-0.09	-0.05%	227.30	-0.09	0.57%	335.34
平均			-0.02	0.14%	233.16	-0.02	-0.02%	320.95

表格 2c - Coastguard QCIF 序列的模拟结果

我们现在转向本发明的第二实施例，其示出了本发明的第二方面。

如上所述，常规完全整数像素允许运动矢量在每个方向上采用 -2.0, -1.0, 0, 1.0, 2.0 等等的位置值。在本发明第二实施例中，选择接

近于预测器的可能位置值。对于最接近 0 的位置值，我们可以使用(代替 1.0)另一位置值，诸如 0.85，从而使得允许的位置值可以包括-2.0，-0.85，0，0.85，2.0 等等。此优点在于，在统计上运动矢量趋向接近于 0。因此，通过选择更加接近 0 的位置，我们会更加接近于真实的运动矢量，并且因而可以给出可以导致较高压缩效率的更好的运动补偿。类似地，可以改变其它位置值。例如，可以将位置值 2.0 改变为 1.9，从而使得允许的位置值会包括-1.9，-0.85，0，0.85，1.9 等等。除了 1.0 的编码运动矢量位置应当解释为 0.85，并且 2.0 的编码运动矢量位置应当解释为 1.9 以外，所提出的改变的优点在于可以使用相同运动矢量代码。

半像素精度允许运动矢量采用诸如 0.0, 0.5, 1.0, 1.5, 2.0 等等之类的位置值。我们建议修改这些位置值，特别是那些接近于预测器的位置值。对于非常接近于 0 的位置值 0.5，我们建议使用一个不同的值。例如，一种可能性是使用 0.4 代替 0.5。换句话说，位置值将包括 0.0, 0.4, 1.0, 1.5, 2.0。类似地，其它位置值可以被修改。例如，位置值 1.0 可以改变为 0.9，从而使得该组新位置值将包括 0.0, 0.4, 0.95, 1.5, 2.0 等等。同样，这可以有助于提高压缩效率。类似地，可以修改其它位置值以提高压缩效率。然而，改变这种位置能够显著地导致编码器和解码器两者处的较高的计算效率。通常，大部分压缩效率增益来自于将位置值改变为接近于预测器。

1/4 像素精度允许运动矢量采用诸如 0.00, 0.25, 0.50, 0.75, 1.00 等等之类的位置值。我们可以修改位置值，特别那些接近于预测器的位置值。例如，我们可以将它们修改为 0.00, 0.20, 0.47, 0.73, 0.99 等。

注意，所提出的方法允许我们在每个整数位置值之间选择任意数量的位置值。例如，在位置值 0 和 1 之间，半像素精度使用 1 个位置值{0.5}，1/4 像素精度使用 3 个位置值{0.25, 0.50, 0.75}，以及 1/8 像素精度使用 7 个位置值{0.125, 0.250, 0.375, 0.500, 0.625, 0.750, 0.875}。提出的方法允许我们在 0 和 1 之间选择任何 N 个位置值。例如，我们可以选择 N=2 值诸如 0.3 和 0.6。

所提出的不均匀的亚像素运动估计和补偿不是必须要应用于每



个帧的每个区域。相反，可以将某些位引入到头部中，以指示对于所述视频帧的每个区域（例如片）其是否被开启或关闭。除此之外，其可以在没有任何语法改变的情况下直接应用于现有标准，因为可以应用相同的运动矢量代码。

采用 H.264 JM82 软件对所提出的不均匀的亚像素运动估计和补偿进行模拟，并且其结果在以上表格中示出，其中 QP 代表量化参数。该模拟在 x 和 y 方向中使用位置值 (...-1, -0.75, -0.5, -0.15, 0, 0.15, 0.5, 0.75, 1..)。即，与使用 1/4 像素空间位置值的标准方法相比较，仅仅修改了在 -0.25 和 +0.25 处的位置值。除了使用新的位置值之外，所述算法在其他方面与已知的 H.264 标准算法相同。如在表中所示出的，第二实施例在实现相似的 PSNR 的同时，明显地降低了比特率。不需要修改 H.264 的语法。

序列		QP=20			QP=24		
		JM82	建议值	差异	JMS2	建议值	差异
Akiyo	PSNR	44.11	44.13	0.02	41.73	41.74	0.01
	比特	3332792	3140080	-5.78%	1766112	1674296	-5.20%
News	PSNR	42.79	42.83	0.04	40.19	40.22	0.03
	比特	6908656	6665160	-3.52%	4156840	4017968	-3.34%
weather	PSNR	43	43.01	0.01	39.75	39.58	0.01
	比特	6929368	6803496	-1.82%	4425912	4351312	-1.69%
Sean	PSNR	42.65	42.67	0.02	39.8	39.83	0.03
	比特	5009336	4755328	-5.07%	2734592	2585480	-5.45%

序列		QP = 28			QP = 32		
		JM82	建议值	差异	JM82	建议值	差异
Akiyo	PSNR	39.33	39.34	0.01	36.66	36.69	0.03
	比特	927791	904744	-2.48%	491032	497600	1.34%
News	PSNR	37.6	37.62	0.02	34.68	34.68	0
	比特	2497384	2441216	-2.25%	1476680	1469640	-0.48%

weather	PSNR	36.29	36.29	0	32.88	32.89	0.01
	比特	274184	272312	-0.80%	1660648	1665368	0.28%
Sean	PSNR	36.94	36.95	0.01	33.92	33.91	-0.01
	比特	1497232	1433544	-4.25%	793552	778816	-1.86%

尽管以上描述的仅仅是本发明的几个实施例，在本发明范围内许多变化都是可能的。

例如，以上给出的本发明的描述是用于具有一个参考帧的 P 帧中的固定大小的块。然而，本发明可以应用于具有多种子块大小的块，并且所述块不是必须要非重叠的。可以有一个以上的参考帧，并且参考帧可以是视频序列的相对于当前帧的过去或将来的任何块。

对于视频，一个图像元素（像素）可以具有一个或多个分量，诸如亮度分量、红色，绿色，蓝色（RGB）分量、YUV 分量、YCrCb 分量、红外分量、X 光或其它分量。图像元素的每个分量是可以表示为数字的符号，所述数字可以是自然数、整数、实数甚至是复数。在自然数的情况下，它们可以是 12 位，8 位，或任何其它位分辨率。虽然在视频中的像素是具有的矩形采样网格和均匀采样周期的 2 维样本，但是所述采样网格不是必须为矩形并且所述采样周期不是必须为均匀的。

#### 工业实用性

本发明的每个实施例适合于由 MPEG-1、MPEG-2、MPEG-4、H.261、H.263、H.264、AVS 或其他可以被修改以包括以上它的相关视频编码标准或方法的快速、低延迟和低代价软件和硬件实现来实现。可能的应用包括数字视频广播（地面、卫星、有线）、数字照相机、数字可携式摄像机（camcorder）、数字录像机、机顶盒、个人数字助理（PDA）、可使用多媒体的蜂窝式电话（2.5G、3G 及以上）、视频会议系统、视频点播系统、无线局域网设备、蓝牙应用、网络服务器、低或高带宽应用中的视频流服务器、视频代码转换机（从一个格式转换到另一格式）、及其它电视通信系统等。

## 参考文献

以下参考文献的公开内容于此全面引入：

[1] Joint Video Team of ITU-T and ISO / IEC JTC 1, “Draft ITU-T Recommendation and Final Draft international Standard of Joint Video Specification (ITU-T Rec.H.264 | ISO/IEC 14496-10AVC)”, document JVT-G050r1, may 2003

[2] A. M. Tourapis, O.C.Au and M. L. Liou, “Predictive Motion Vector Field Adaptive Search Technique (PMVFAST)”, ISO/IEC JTC1/SC29/WG11 MPEG2000, Noordwijkerhout, ML, March’2000

[3] R. LI, B. Zeng and M.L. Liou, “A new three - step search algorithm for block motion estimation”, On Circuits and Systems for Video Technology, vol4, no4, pp438-42, Aug’94

[4] Z.L.He 和 M.L. Liou, “A high performance fast search algorithm for block matching motion estimation”, IEEE Trans. on Circuits and Systems for Vld&o 2Technology, vol.7, no5, pp826-8, Oct’97

[5] A.M. Tourapis, O.C. Au, and M.L. Liou, “Fast Motion Estimation using Circular Zonal Search”, Proc. of SPIE Sym. Of Visual Comm. & Imagg Processin, vol2, pp.1496-1504, Jan.25-27, ‘99

[6] A, M. Tourapis, O. C. Au, M.L Liou, G. Shen, and I. Ahmad, “Optimizing the Mpeg-4 Encoder - Advanced Diamond Zonal Search”, in Pros. of 2000 IEEE Inter. Sym. on Circuits and Systems, Geneva, Switzerland, May, 2000

[7] K.K.Ma and P.I. Hosur, “Performance Report of Motion Vector Field Adaptive Search Technique (MVFAST) ”, in ISO/IEC JTC1/SC29/WG11 MPEG99/m81, Noordwijkerhout, NL Mar’00

[8] A.M. Tourapis, O.C. Au, and M.L. Liou, “Fast Block-Matching Motion Estimation using Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) ”, in ISO/IEC/ JTC1/SC29/WG11 MPEG2000 / M5866, Noordwijkerhout, NL, Mar’00

[9] Implementation Study Group, “Experimental conditions for

---

evaluating encoder motion estimation algorithms”, in ISO/IEC JTC1 / SC29 / WG11 MPEG99/n3141, Hawaii, USA, Dec'99

[10] “MPEG-4 Optimization Model Version 1.0”, in ISO/IEC JTC1/SC29/WG11 MPEG2000/n3324, Noordwijkerhout, NL, Mar'00

[11] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, “Motion compensated interframe coding for video conferencing” Proc. Nat. Telecommun. Conf., New Orleans, LA, pp. G.5.3.1-G.5.3.5, Dec'81 .

[12] J.R. Jain and A.K. Jain, “Displacement measurement and its application in interframe image coding”, IEEE Trans. On Communications, vol. COM-29, pp.1799-808, Dec'81

[13] JVT reference software JM9.2 for JVT/H.264 FRext

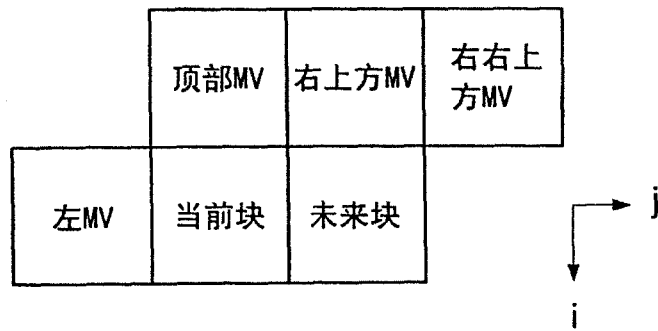


图1

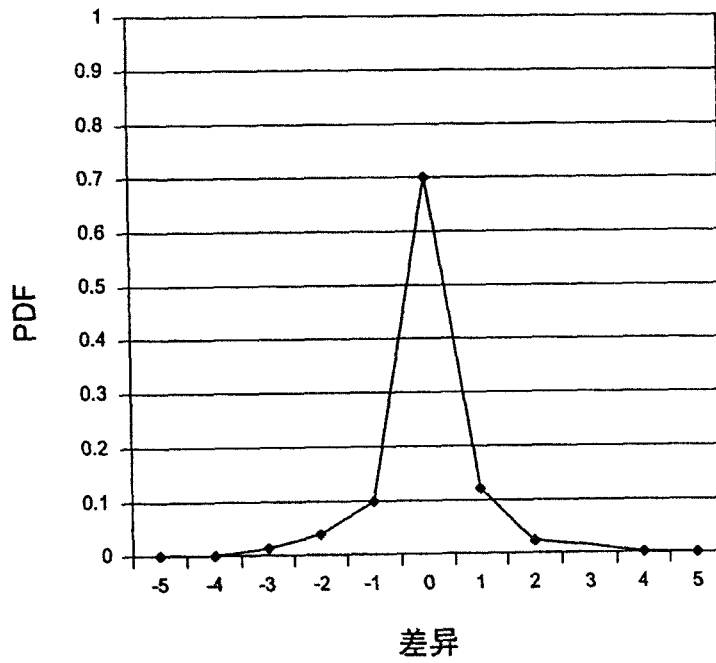


图2

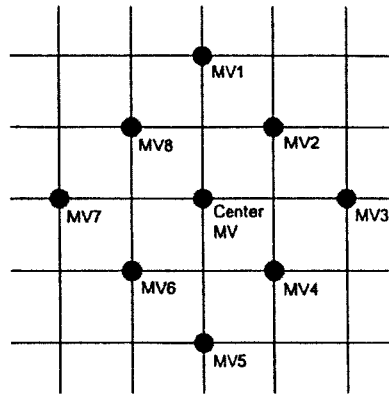


图3a

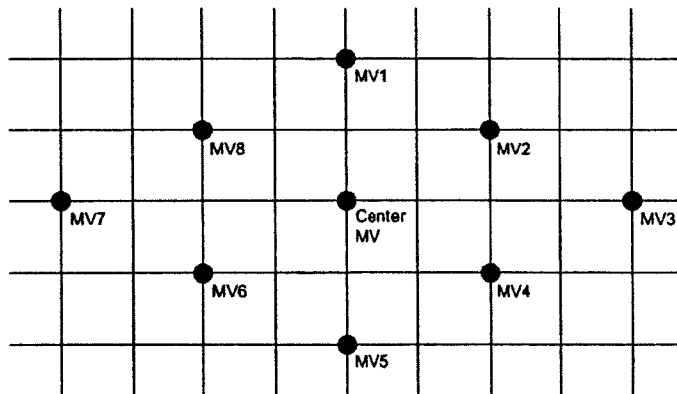


图3b

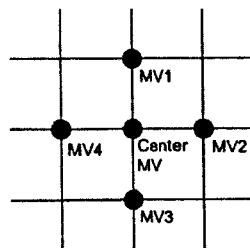


图4

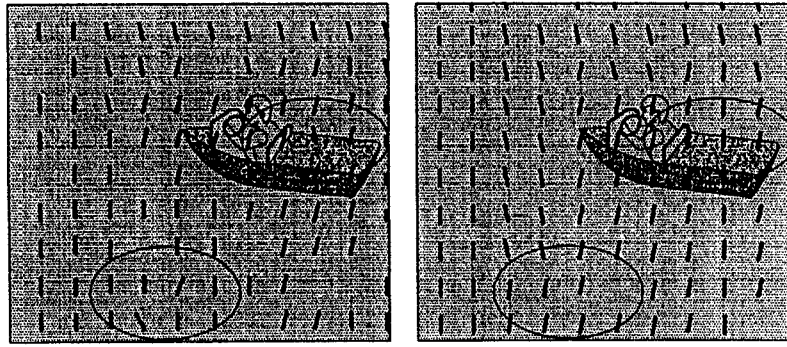


图5a

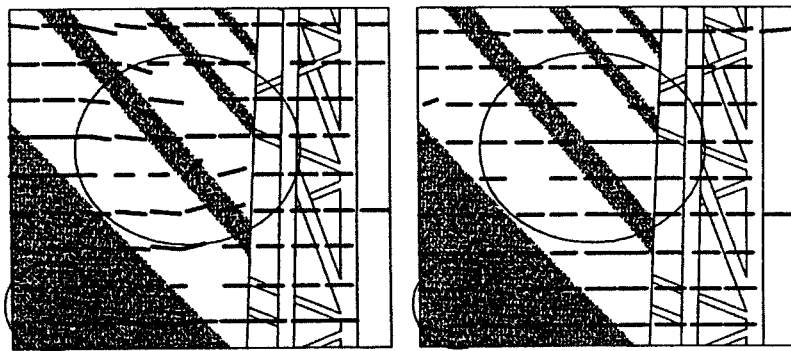


图5b

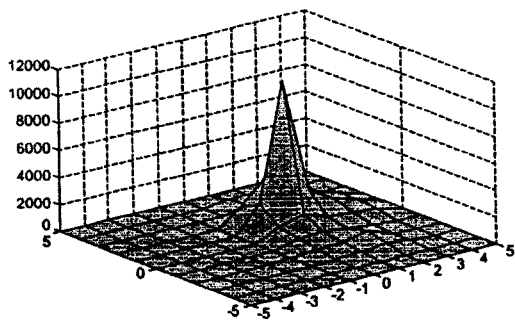


图6a

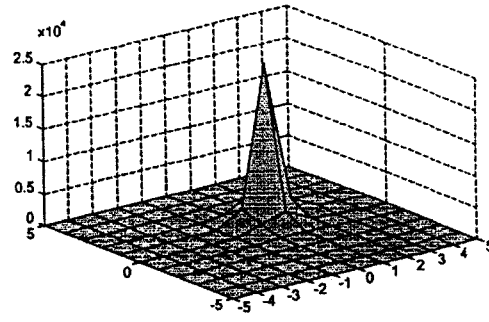


图6b

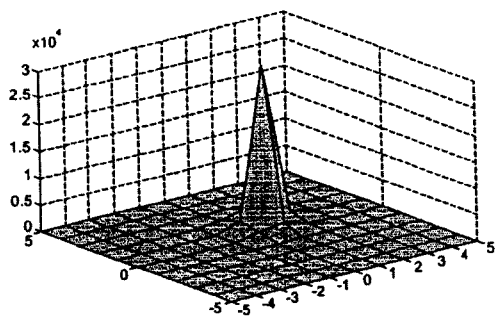


图6c

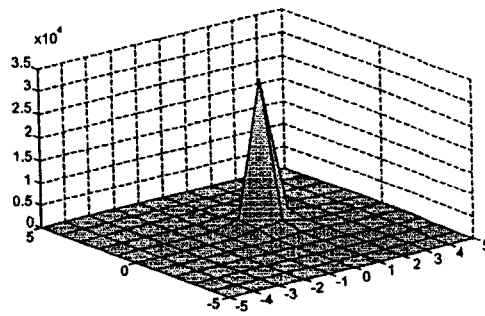


图6d

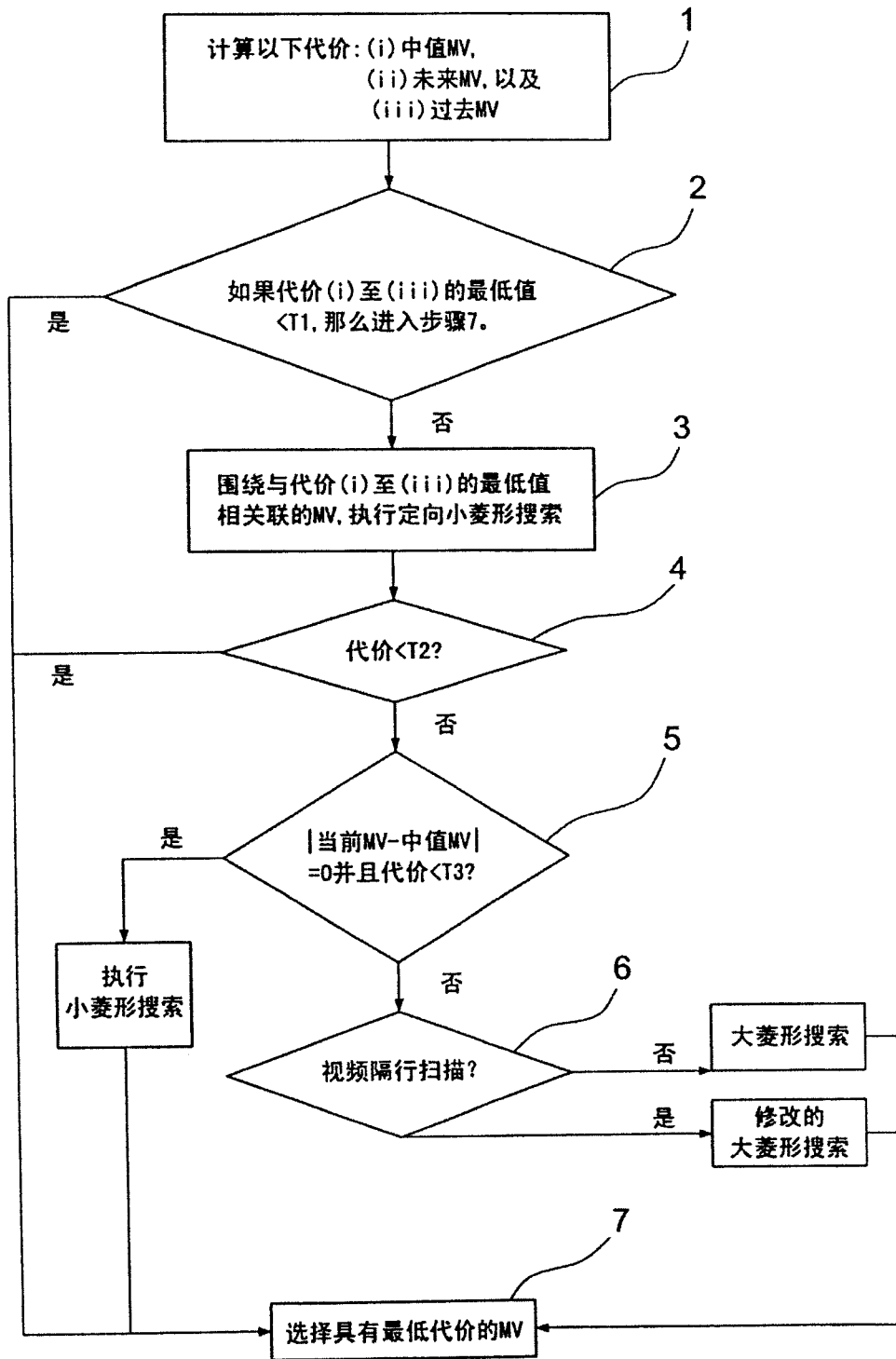


图7