

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G06F 17/00

G06F 9/06 H04N 7/32



[12] 发明专利申请公开说明书

[21] 申请号 03110399.5

[43] 公开日 2003 年 11 月 5 日

[11] 公开号 CN 1453725A

[22] 申请日 2003.4.10 [21] 申请号 03110399.5

[30] 优先权

[32] 2002. 4. 10 [33] US [31] 60/371, 860

[32] 2003. 3. 4 [33] US [31] 10/382, 294

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 S·斯里尼瓦桑

[74] 专利代理机构 上海专利商标事务所

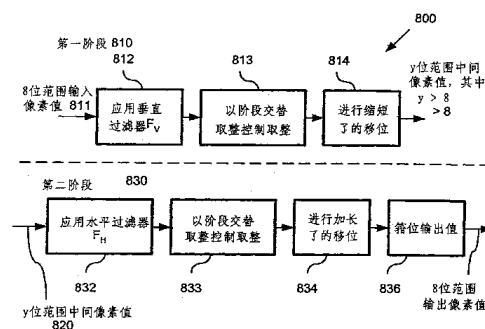
代理人 张政权

权利要求书 3 页 说明书 31 页 附图 13 页

[54] 发明名称 用于多阶段插值的取整控制

[57] 摘要

描述了用于多阶段插值中的取整控制的多种技术和工具。 举例而言，在运动估计和补偿期间，视频编码器在通过多阶段插值计算参照视频帧中的像素值时使用阶段交替取整控制。 或者，在运动补偿期间，视频解码器在通过多阶段插值计算参照视频帧中的像素值时使用阶段交替取整控制。



ISSN 1008-4274

8、一种计算机可读取媒介，存储用于使一计算机系统在视频编码期间执行权利要求 1 所述方法的计算机可执行指令。

9、一种计算机可读取媒介，存储用于使一计算机系统在视频解码期间执行权利要求 1 所述方法的计算机可执行指令。

10、一种存储计算机可执行指令的计算机可读取媒介，这些指令用于使一被编程的计算机系统能执行一方法，该方法包括：

在参照视频帧中的多个像素位置中的每一个处，用多阶段插值计算一个像素值，其中多阶段插值包括阶段交替取整控制，以在多阶段插值过程中逐阶段地改变边缘像素值的取整方向。

11、如权利要求 10 所述的计算机可读取媒介，其特征不在于计算包括：

在多个阶段插值的第一个阶段中，根据阶段交替取整控制在第一方向上将第一边缘像素值取整到最近的整数；以及

在多个阶段插值的第二个阶段中，根据阶段交替取整控制在与第一方向相反的第二方向上将第二边缘像素值取整到最近的整数。

12、如权利要求 11 所述的计算机可读取媒介，其特征不在于一个取整控制参数控制第一和第二方向。

13、如权利要求 12 所述的计算机可读取媒介，其特征不在于取整控制参数在一帧一帧的基础上在 0 和 1 间交替。

14、如权利要求 10 所述的计算机可读取媒介，其特征不在于所述参照视频帧是第一参照视频帧，该方法进一步包括为第二参照视频帧重复所述计算，并根据一取整控制参数为第二参照视频帧倒转第一取整方向。

15、如权利要求 14 所述的计算机可读取媒介，其特征不在于第一参照视频

- 1、在一个计算机系统中的一个计算机实现的方法，该方法包含：
在多个采样位置中的每一个处，用具有阶段交替取整控制的多阶段插值计算一个值。
- 2、如权利要求 1 所述的方法，其特征在于计算包括：
在多个阶段的第一个阶段中，根据阶段交替取整控制在第一方向上将第一边际值取整到最近的整数；以及
在多个阶段的第二个阶段中，根据阶段交替取整控制在与第一方向相反的第二方向上将第二边际值取整到最近的整数。
- 3、如权利要求 2 所述的方法，其特征在于一个取整控制参数控制第一和第二方向。
- 4、如权利要求 3 所述的方法，其特征在于取整控制参数在一帧一帧的基础上交替。
- 5、如权利要求 1 所述的方法，其特征在于所述多个采样位置在参照视频帧中，且其中所述值是像素值。
- 6、如权利要求 5 所述的方法，其特征在于所述参照视频帧是第一参照视频帧，所述阶段交替取整控制为第一参照视频帧指示第一取整方向，该方法进一步包括为第二参照视频帧重复所述计算，并根据一取整控制参数为第二参照视频帧倒转第一取整方向。
- 7、如权利要求 6 所述的方法，其特征在于第一参照视频帧是一关键帧，对于该关键帧，第一取整方向是预定的。

帧是一关键帧，对于该关键帧，第一取整方向是预定的。

16、一种系统，包括：

用于用具有阶段交替取整控制的多阶段插值计算参照视频帧中的像素值的装置；

用于相对于该参照视频帧进行运动补偿的装置。

17、如权利要求 16 所述的系统，其特征在于在多个阶段的第一个阶段中，用于计算的装置在第一方向上将第一边际像素值取整到最近的整数；在多个阶段的第二个阶段中，用于计算的装置在与第一方向相反的第二方向上将第二边际像素值取整到最近的整数。

18、如权利要求 16 所述的系统，其特征在于一取整控制参数控制第一和第二方向，该取整控制参数在逐帧的基础上在 0 和 1 之间交替。

用于多阶段插值的取整控制

相关申请参照

本申请要求美国临时专利申请（序列编号 60/371860，2002 年 4 月 10 日提交，在此附录以作参考）的权益。本申请与题为“在运动估计与补偿中的次像素插值”的美国专利申请（序列编号 aa/bbbccc，与本发明同时提交）、题为“近似双立方过滤”的美国专利申请（序列编号 aa/bbbccc，与本发明同时提交）、题为“色度运动矢量取整”的美国专利申请（序列编号 aa/bbbccc，与本发明同时提交）有关，这些都在此附录以作参考。

技术领域

描述了用于多阶段插值中的取整控制的技术和工具。举例而言，某种视频编码器或解码器在通过多阶段插值计算参照视频帧中的像素值时，使用了阶段交替取整控制。

背景技术

数字视频消耗大量的存储和传输容量。一个典型的未加工数字视频序列包括了每秒 15 或 30 帧。每一帧可以包含几万或几十万的像素（也称象素）。每个像素都代表了图像的一个微小元素。在未加工的形式下，计算机一般以 24 位来表示一个像素。举例来说，一个像素可能由定义该像素灰度成分的一个 8 位亮度值（也称亮值）和定义该像素色彩成分的两个 8 位色度值（也称色值）组成。因此，一个典型的未加工数字视频序列的每秒的比特数（或比特率）可能是每秒 5 百万比特或更多。

许多计算机与计算机网络缺乏资源来处理未加工的数字视频。由于这一原因，工程师们使用压缩（也称编码或组码）来降低数字视频的比特率。压缩可

以是无损的（其中视频的质量不受损失，但由于视频的复杂性，比特率的降低有限），也可以是有损的（其中视频的质量会受损失，但比特率的降低较为显著）。解压则是压缩的反向。

一般而言，视频压缩技术包括帧内压缩和帧间压缩。帧内压缩技术压缩单独的帧（一般称为 I 帧或关键帧）。帧间压缩技术参考前面或后面的帧（一般称为预测帧、P 帧或 B 帧）来压缩多个帧。

微软公司的视窗媒体视频版本 8（WMV8）包括一个视频编码器和一个视频解码器。WMV8 编码器使用帧内和帧间压缩，而 WMV8 解码器使用帧内和帧间解压。WMV8 编码器中的帧间压缩使用基于块的运动补偿预测编码，并继之以残差的转换编码。

在 WMV8 中，一个帧以三个像素平面来表示：一个亮度像素值的亮度（Y）平面和两个色度像素值的色度（U，V）平面。Y 平面的解析度在水平和垂直都是 U 和 V 平面解析度的两倍。这样，一个 320 像素 x240 像素的帧就具有一个 320 像素 x240 像素的 Y 平面，和 160 像素 x120 像素的 U 与 V 平面。

WMV8 编码器将一个预测帧分割为 8x8 的像素块。四个 8x8 的亮度块和两个同位的 8x8 的色度块（一个用于 U 色度平面，一个用于 V 色度平面）的组合形成 16x16 的宏块。这样，每个 16x16 的宏块就包括四个 8x8 的亮度块和两个 8x8 的色度块。

对于预测帧的宏块，WMV8 编码器进行运动估计。运动估计通过在预测帧的宏块中搜索与匹配来自某个参照帧的宏块，来近似该预测帧宏块的运动。举例来说，在图 1 中，WMV8 编码器计算在预测帧（110）中的一个宏块（115）的一个运动矢量。为了计算该运动矢量，编码器在一个参照帧（130）的一个搜索区域（135）中进行搜索。在该搜索区域（135）中，编码器将来自预测帧（110）的宏块（115）的亮度值与来自参照帧（130）的多个候选块的亮度值相比较，以找出一个好的匹配。WMV8 编码器可以转换运动矢量的精确度，也可以使用具有整个、半个或四分之一像素的水平解析度以及整个或半个像素的垂直解析度的搜索范围和运动矢量。以次像素精确度的运动矢量，WMV8 编码器可以近似视频序列中的次像素运动。

在运动补偿期间，WMV8 编码器使用预测帧的宏块的运动矢量来确定来自参

照帧的宏块的预测。对于每个预测运动的宏块，WMV8 编码器计算原始宏块与其预测之间的差（也称残留或误差）。WMV8 编码器将这些残留分割成块，并有损压缩这些残块。为重建预测帧的预测运动的宏块，WMV8 编码器解压这些残留，并将它们叠加到相应宏块的预测上。

WMV8 编码器还使用预测帧的宏块的运动矢量来确定来自参照帧的宏块的预测。为重建预测帧的预测运动的宏块，WMV8 编码器解压这些残留，并将它们叠加到宏块的预测上。

在运动估计或补偿期间，当一个运动矢量具有次像素的精确度（即半个像素或四分之一像素）时，WMV8 编码器或解码器必须确定在参照帧中次像素位置上的像素值。WMV8 编码器或解码器用插值过滤器来为次像素位置产生数值。图 2 显示了次像素采样位置 H_0 、 H_1 、 H_2 ，其值由整数像素值 a 、 b 、 c 、...、 p 的插值来计算。

当以半像素的运动矢量精确度来运作时，用于三个不同半像素位置 H_0 、 H_1 、 H_2 的亮度像素值的插值过滤器是：

$$H_0 = (f + g + R_2) \gg 1 \quad (1)$$

$$H_1 = (f + j + R_2) \gg 1 \quad (2), \text{ 和}$$

$$H_2 = (f + g + j + k + R_1) \gg 2 \quad (3)$$

其中 R_1 和 R_2 是取整控制值，由一个指示对特定帧的取整模式的 1 位取整控制标志来控制。如果该取整控制标志置为 0，则 $R_1=2$ 而 $R_2=1$ 。如果该取整控制标志置为 1，则 $R_1=R_2=0$ 。对每个 P 帧该取整控制标志的值在 1 与 0 之间交替。在每个 I 帧，该取整控制标志置为 0。这样，取整控制按一帧一帧来进行。

公式 1、2 和 3 是双线性插值的示例。双线性插值很快，趋向于平滑的像素值。这种平滑可能具有所希望的效果（例如降低量化噪声的可察觉性），但也可以导致有效像素信息的损失。

对于四分之一像素运动矢量的解析度，WMV8 编码器或解码器首先使用双立方过滤器来插值在半像素位置的亮度像素值。双立方插值比双线性插值慢，但趋向于保持边际值，并能使有效像素信息损失较少。用于三个不同半像素位置 H_0 、 H_1 、 H_2 的双立方过滤器是：

$$H_0 = (-e + 9f + 9g - h + 8) \gg 4 \quad (4)$$

$$H_1 = (-b + 9f + 9j - n + 8) \gg 4 \quad (5), \text{ 和}$$

$$H_2 = (-t_0 + 9t_1 + 9t_2 - t_3 + 8) \gg 4 \quad (6)$$

其中 t_0 、 t_1 、 t_2 和 t_3 按以下来:

$$t_0 = (-a + 9b + 9c - d + 8) \gg 4 \quad (7)$$

$$t_1 = (-e + 9f + 9g - h + 8) \gg 4 \quad (8)$$

$$t_2 = (-i + 9j + 9k - l + 8) \gg 4 \quad (9), \text{ 和}$$

$$t_3 = (-m + 9n + 9o - p + 8) \gg 4 \quad (10)$$

公式 (4) - (10) 可以导致输出超出输入值的范围。举例来说, 对于 8 位的输入 (范围在 0 至 255), 值 0 255 255 0 的序列在公式 (4) - (10) 中任何一个都会产生一个输出值 287。所以, WMV8 编码器或解码器要箝位 (或“裁剪”) 公式 (4) - (10) 中任何一个的输出值, 使其在有效范围之内。举例来说, 对于 8 位的输出值, 小于 0 的值将变到 0, 而大于 255 的值将变到 255。箝位解决了范围问题, 但减慢了计算。而且, 箝位还会导致精确度的损失。

而后 WMV8 编码器或解码器在一个后续插值阶段计算在某个四分之一像素位置的像素值。这些四分之一像素位置在水平上位于两个半像素位置或一个整像素位置与一个半像素位置之间。对于这些四分之一像素位置, WMV8 编码器或解码器使用采用两个水平相邻的半像素/整像素位置的双线性插值 (即 $(x + y + 1) \gg 1$), 而不进行取整控制。

当计算出亮度运动矢量后, WMV8 编码器或解码器导出同位的色度运动矢量。由于在 WMV8 中的一个色度平面在水平和垂直上都是一个亮度平面的一半, 亮度运动矢量值必须缩放成适当的色度运动矢量值。在 WMV8 中, 这一转换过程包括了将亮度运动矢量减半, 并将得到的色度运动矢量取整到半像素的精确度。这样, 具有半像素精确度的亮度运动矢量就不转换到具四分之一精确度的色度运动矢量。而且, 在 WMV8 中的色度取整以一种单独的模式运作, 不能由用户改变或选择。

在 WMV8 中, 在参照帧中次像素位置的像素值在某些情况下可能表现出下溢或上溢。举例来说, 如果相邻整像素位置的值是 255, 而相邻半像素位置的值是 287 ($0 + 9*255 + 9*255 - 0 + 8 \gg 4 = 287$), 在四分之一像素位置的亮度像素值就可能是 271 (在 0 至 255 的范围之外) ($255 + 287 + 1 \gg 1 =$

271)。为了解决这一问题，在将残留块叠加到宏块的预测上后，如果必要，WMV8 编码器或解码器将箝位对该宏块的重建值，使之在 0 至 255 的范围内。

除 WMV8 以外，还有几个与视频压缩和解压有关的国际标准。这些标准包括运动图像专家组（“MPEG”）1、2 和 4 标准，和来自国际电信联盟（“ITU”）的 H. 261、H. 262 和 H. 263 标准。与 WMV8 类似，这些标准都使用帧内和帧间压缩的组合，尽管这些标准在所用压缩技术的细节上一般与 WMV8 不同。

有几个标准（如 MPEG4 和 H. 263）使用双线性过滤器和基本取整控制为半像素运动提供估计和补偿。而且，在 H. 263 中，理论上具有四分之一像素解析度（即半像素亮度运动矢量的一半解析度）的色度运动矢量被取整到半像素或全像素精确度，这样在色度空间就不允许有四分之一像素的值。关于在这些标准中运动估计/补偿的其它细节，请参见这些标准自身的说明。

运动估计和补偿是有效的压缩技术，但（如在 WMV8 和上面所讨论的标准中的）多种以前的运动估计/补偿技术具有几项缺点，包括：

（1）当计算在参照帧中次像素位置的像素值时，编码器和解码器会不必要地在中间数值上损失精度。举例而言，当在 WMV8 中计算一个四分之一像素位置的像素值时，在半像素位置的中间值向右移位了 4 位，而无视可以有更大的位深度的事实。而且，WMV8 编码器和解码器还在四分之一像素位置的两阶段插值期间箝位中间值，这减慢了计算，并导致不必要的精度损失。

（2）在四分之一像素运动估计和补偿中的像素值插值在许多情况下是没有效率的。举例来说，在 WMV8 中，一维四分之一像素位置的计算要求对半像素位置使用某个过滤器后，继续使用某个双线性过滤器。

（3）编码器和解码器不能计算在多阶段插值中可能产生的取整误差的累积。举例来说，在一个视频序列中一帧一帧地将像素值反复向下取整时，会发生误差。这一取整误差可以在低质量、低比特率的视频序列中造成可察觉的伪像。举例而言，当 WMV8 编码器和解码器以多阶段对在四分之一像素位置的像素值插值时，不使用取整控制。相反，每个阶段的结果都以与每个阶段插值（且没有取整控制）相同的方式来取整；而且

（4）色度取整不是以四分之一像素精确度来进行的，而且对于色度运动矢量的取整选项也没有控制。举例来说，WMV8 编码器和解码器将所有色度运动

矢量取整到某个半像素值，并只以一个单独的模式来运作。

考虑到对数字视频而言运动估计和补偿的极端重要性，运动估计和补偿成为丰富发展的领域就不奇怪了。但是，无论以前的运动估计和补偿技术具有何种益处，它们都没有以下技术与工具的优点。

发明内容

总体而言，详细描述关注于用于多阶段插值中的取整控制的多种技术和工具。举例而言，在视频编码和解码应用中，视频编码器和解码器在通过多阶段插值计算参照视频帧中的像素值时，使用了阶段交替取整控制。使用阶段交替取整控制减小了来自于多阶段插值的取整控制的逐帧累积。可以通过减小在某类视频序列（如，因特网上的低质量、低比特率视频序列流）中的可察觉的模仿来提高总的性能。

一个部件（如视频编码器或解码器）用具有阶段交替取整控制的多阶段插值计算值。例如，在第一阶段中，该部件在第一方向上将第一边际值取整到最近的整数，在第二阶段，部件在与第一方向相反的第二方向上将第二边际值取整到最近的整数。在某些情况中，取整控制参数控制第一和第二方向。该取整控制参数自身可以在逐帧或其他基础上交替（在 0 和 1 之间）。

这多种技术和工具可以组合或独立使用。通过以下的详细描述，同时参照附图，其它的特性和优点将得以彰显。

附图说明

图 1 是一幅显示遵循已有技术的视频编码器中运动估计的图。

图 2 是一幅显示遵循已有技术的用于次像素运动估计和补偿中插值的次像素位置的图。

图 3 是一幅可在其中实现几种所描述的实施例的合适的计算环境的方框图。

图 4 是一幅在几种所描述的实施例中使用的普遍化的视频编码器系统的方

框图。

图 5 是一幅在几种所描述的实施例中使用的普遍化的视频解码器系统的方框图。

图 6 是一幅显示在次像素运动估计和补偿期间用于像素值插值的位置的图。

图 7 是一幅显示其像素值用于为次像素位置计算插值后像素值的整像素位置的图。

图 8 是一幅显示用于对次像素位置的值进行插值的两阶段插值技术的图。

图 9 是一幅显示一个水平半像素而垂直半像素的采样位置和用以计算在该采样位置的值的次像素位置中间值的图。

图 10 是一幅显示一个水平四分之一像素而垂直半像素的采样位置、一个水平半像素而垂直四分之一像素的采样位置和用以计算在该采样位置的值的次像素位置中间值的图。

图 11 是一幅显示一个水平四分之一像素而垂直四分之一像素的采样位置和用以计算在该采样位置的值的次像素位置中间值的图。

图 12 是一幅显示一种具有增强动态范围（以比特为单位）的中间值的多阶段插值技术的图。

图 13 是一幅显示一种跳过了箝位的多阶段插值技术的图。

图 14 是一幅显示一种推迟移位的多阶段插值技术的图。

图 15 是一幅显示一种使用阶段交替取整控制的多阶段插值技术的图。

图 16 是一幅显示一种用于在多种色度取整和插值模式中进行选择的技术的流程图。

图 17 是一个显示第一种色度取整模式的表格。

图 18 是一个显示第二种色度取整模式的表格。

具体实施方式

所描述的实施例与用于运动估计和补偿中的次像素插值的技术和工具有关。多种实施例涉及通过将箝位和/或移位（这些操作可能导致精度的损失）

推迟到插值中的较后阶段而在多阶段插值中保持精度的技术和工具。其它的实施例则与用于多阶段插值的高效过滤或取整操作有关。

一个编码器或解码器在一个参照帧或一个帧的一部分如一个或多个块或宏块上进行次像素插值。该编码器/解码器在参照帧内的次像素位置计算像素值。而后该编码器/解码器可以用次像素精确的运动矢量进行运动补偿。

在某些实施例中，一个视频编码器或解码器在视频编码或解码应用中进行次像素插值。可替换地，另一个编码器或解码器，或另一种部件，进行次像素插值或以下在另一种应用中描述的其它技术。

作为在参照帧上进行次像素插值的一种替代，在某些实施例中，一个编码器/解码器在域、对象层或其它图像上进行次像素插值。

在某些实施例中，通过计算在 YUV 色彩空间的参照帧的亮度和色度平面中的像素值来进行次像素插值。可替换的是，色彩空间可以不同（如 YIQ 或 RGB）。

这多种技术和工具可以组合或独立使用。不同的实施例实现了一种或多种所描述的技术和工具。尽管这些技术的运作为演示目的而一般按特定的顺序来描述，但应当明白这种描述方式包括了对操作顺序的较小的重新排列，除非要求某种特定的排序。举例来说，顺序描述的操作在有些情况下可能重新排列，或同时进行。而且，为简单起见，流程图一般不显示特定技术可能与其它技术共同使用的多种途径。

在某些实施例中，一个视频编码器和解码器在比特流中使用多种标志和记号。虽然已描述了特定的标志和记号，应当明白这种描述方式包括了对这些标志和记号的不同的转换（如是 0 而不是 1）。

1. 计算环境

图 3 说明了一个可以在其中实现几种所描述的实施例的合适的计算环境（300）的普遍示例。该计算环境（300）并不提出对使用或功能的任何限制，而这一技术和工具可以以不同的通用或专用计算环境来实现。

参照图 3，该计算环境（300）包括至少一个处理单元（310）和存储器（320）。在图 3 中，这一最基本的配置（330）包括在一根虚线内。处理单元（310）执行计算机可执行指令，可以是一个真实的或虚拟的处理器。在多处理系统中，

有多个处理单元执行计算机可执行指令来提高处理能力。存储器（320）可以是易失存储器（如寄存器、高速缓存、RAM）、非易失存储器（如ROM、EEPROM、闪存等）或这两者的某种组合。存储器（320）储存在编码器和/或解码器（如视频编码器和/或解码器）中实现次像素插值技术的软件（380）。

一个计算环境可以具有其它的特性。举例来说，计算环境（300）包括了存储设备（340）、一个或多个输入设备（350）、一个或多个输出设备（360），以及一个或多个通信连接（370）。某种互连机制（未示出）如总线、控制器或网络将计算环境（300）的部件相互连接。一般来说，操作系统软件（未示出）为在计算环境（300）中运行的其它软件提供一个操作环境，并协调计算环境（300）的部件的活动。

存储设备（340）可以是可拆卸或不可拆卸的，包括磁盘、磁带或卡带、CD-ROM、DVD，或可用以储存信息并可在计算环境（300）内读取的任何其它媒介。存储设备（340）还储存用于实现次像素插值技术的软件（380）的指令。

输入设备（350）可以是某种触点输入设备如键盘、鼠标、触笔或轨迹球、某种语音输入设备、某种扫描设备，或向计算环境（300）提供输入的其它某种设备。对于音频或视频编码，输入设备（350）可以是声卡、视频卡、电视调谐卡，或接受模拟或数字形式的音频或视频输入的类似设备，或将音频或视频样本读入计算环境（300）的CD-ROM或CD-RW。输出设备（360）可以是显示器、打印机、扬声器、CD刻录机，或提供来自计算环境（300）的输出的其它设备。

通信连接（370）实现了通过某种通信媒介向另一个计算实体的通信。通信媒介传达信息，如计算机可执行指令、音频或视频输入或输出，或调制数据信号形式的其它数据。调制数据信号是这样一种信号，其一种或多种特征以能够在编码其中信息的方式来设定或变化。举例来说（但非限制），通信媒介包括以电、光、RF、红外、声或其它载体实现的有线或无线技术。

这些技术和工具可以在计算机可读取媒介的总体环境中来描述。计算机可读取媒介是可以在计算环境内访问的任何可用媒介。举例来说（但非限制），就计算环境（300）而言，计算机可读取媒介包括存储器（320）、存储设备（340）、通信媒介，和以上这些的任何组合。

这些技术和工具可以在计算机可执行指令（如那些包含在程序模块中，在计算环境中某个真实或虚拟目标处理器上执行的）的总体环境中来描述。一般而言，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、库、对象、类、组件、数据结构等。这些程序模块的功能可以按在多个实施例中所期望的那样在程序模块间组合或分割。用于程序模块的计算机可执行指令可以在某个局部或分布式计算环境内执行。

为演示目的，本详细描述使用如“确定”和“选择”这样的术语来描述在计算环境中的计算机操作。这些术语是对由计算机进行的操作的高层次抽象，不应与由人进行的行动相混淆。对应于这些术语的实际计算机操作根据实现而有变化。

2. 普遍化的视频编码器和解码器

图 4 是一幅某种普遍化的视频编码器（400）的方框图，而图 5 是一幅某种普遍化的视频解码器（500）的方框图。

在该编码器和解码器内的模块间所显示的关系指示了在该编码器和解码器中的主要信息流，其它关系为简单起见没有示出。具体来说，图 4 和 5 通常并不示出指示用于视频序列、帧、宏块、块等的编码器设定、模式、表格等的伴随信息。这样的伴随信息在输出比特流中发送，一般是在这些伴随信息的熵编码之后。输出比特流的格式可以是视窗媒体视频格式或某种其它格式。

编码器（400）和解码器（500）是基于块的，并使用 4:2:0 宏块格式，每个宏块包括 4 个 8x8 的亮度块（有时当作一个 16x16 宏块来对待）和两个 8x8 的色度块（如一个 U 块，一个 V 块）。可替代的是，编码器（400）和解码器（500）可以是基于对象的，使用某种不同的宏块或块模式，或在与 8x8 的块和 16x16 的宏块不同的尺寸或配置的像素集上进行操作。

根据所期望的实现和压缩类型，编码器或解码器的模块可以添加、省略、分割成多个模块、与其它模块组合，以及/或以类似的模块来代替。在可替代的实施例中，具有不同模块和/或其它模块配置的编码器或解码器实现一种或多种所描述的技术。

A. 视频编码器

图 4 是一幅某种普遍化的视频编码器系统 (400) 的方框图。该视频编码器系统 (400) 接收一个包括当前帧 (405) 的视频帧序列, 并产生压缩了的视频信息 (495) 作为输出。视频编码器的特定实施例一般使用普遍化的编码器 (400) 的某个变异或补充版本。

编码器系统 (400) 压缩预测帧和关键帧。为演示目的, 图 4 显示了一条关键帧通过编码器系统 (400) 的途径, 和一条前向预测帧的途径。编码器系统 (400) 的许多部件都用于压缩关键帧和预测帧。由这些部件进行的具体操作可以根据所压缩信息的类型而变化。

预测帧 (也称 p 帧、指双向预测的 b 帧, 或交互编码帧) 是以来自其它一个或多个帧的预测 (或差异) 来表示的。预测残留就是所预测的与原始帧之间的差异。与之相反, 关键帧 (也称 I 帧、内编码帧) 则是不参照其它帧而压缩的。

如果当前帧 (405) 是一个前向预测帧, 则一个运动估计器 (410) 参考一个参照帧来估计宏块或该当前帧 (405) 的其它像素集的运动, 该参照帧是重建的前一帧, 缓冲在帧存储设备 (420) 中。在可替换的实施例中, 该参照帧是后一帧, 或者该当前帧是双向预测的。该运动估计器 (410) 将运动信息 (415), 如运动矢量, 作为伴随信息输出。一个运动补偿器 (430) 将该运动信息 (415) 应用于重建的前一帧 (425), 以形成一个运动补偿了的当前帧 (435)。然而, 预测很少是完美的, 而运动补偿了的当前帧 (435) 与原始的当前帧 (405) 之间的差异就是预测残留 (445)。可替换的是, 运动估计器和运动补偿器应用另一种类型的运动估计/补偿。

一个频率变换器 (460) 将空间域的视频信息转换成频域 (即频谱) 数据。对于基于块的视频帧, 该频率变换器 (460) 对运动预测残留数据的块应用一个离散余弦变换 (“DCT”) 或 DCT 的变异, 产生 DCT 系数块。可替换的是, 该频率变换器 (460) 应用另一种传统的频率变换如 Fourier 变换, 或使用小波或次波段分析。在某些实施例中, 该频率变换器 (460) 对用于关键帧的空间预测残留块应用频率变换。该频率变换器 (460) 可以应用 8x8、8x4、4x8 或其它尺寸的频率变换。

而后一个量化器（470）将频谱数据系数块量化。该量化器以一种在逐帧或其它基础上变化的步长来对频谱数据应用统一的标量量化。可替换的是，该量化器对频谱数据系数应用另一种量化，例如不统一的、矢量的或非自适应性的量化，或者直接在一个不使用频率变换的编码器系统中量化空间域数据。在自适应量化以外，编码器（400）还可以为速率控制使用帧丢弃、自适应过滤或其它技术。

当后续运动估计/补偿需要重建的当前帧时，一个反向量化器（476）在量化了的频谱数据系数上进行反向量化。随后一个反向频率变换器（466）进行频率变换器（460）的反向操作，产生一个重建的（对预测帧的）预测残留或一个重建的关键帧。如果当前帧（405）是一个关键帧，则采用重建的关键帧作为重建的当前帧（未示出）。如果当前帧（405）是一个预测帧，则将重建的预测残留叠加到运动补偿了的当前帧（435）上，以形成重建的当前帧。帧存储设备（420）缓冲重建的当前帧，用于预测下一帧。在某些实施例中，编码器对重建的帧应用一个解块过滤器，以适应性地平滑在该帧的块中的不连续。

熵编码器（480）压缩量化器（470）的输出以及某些伴随信息（如运动信息（415）、量化步长）。典型的熵编码技术包括算术编码、微分编码、Huffman编码、运行长度编码、LZ编码、字典编码，以及以上这些的组合。熵编码器（480）一般对于不同的信息种类（如DC系数、AC系数、不同种类的伴随信息）使用不同的编码技术，并且可以从多种编码表格中选择某种特定的编码技术。

熵编码器（480）将压缩了的视频信息（495）放到缓存（490）中。将一个缓存水平指示器反馈给比特率自适应模块。从缓存（490）中按某种不变的或相对不变的比特率消耗压缩了的视频信息（495），并储存用于按该比特率的后续流注。可替换的是，熵编码器系统（480）在压缩后立即流注压缩了的视频信息。

在缓存（490）之前或之后，可以将压缩的视频信息（495）按信道编码，用于在网络上的传输。信道编码可以对压缩的视频信息（495）应用差错检测和纠正数据。

B. 视频解码器

图 5 是一幅某种通用视频解码器系统(500)的方框图。该解码器系统(500)接收用于一个压缩的视频帧序列的信息(595)，并产生包括一个重建帧(505)的输出。视频解码器的特定实施例一般使用普遍化的解码器(500)的某个变化或补充版本。

解码器系统(500)解压预测帧和关键帧。为演示目的，图 5 显示了一条关键帧通过解码器系统(500)的途径，和一条前向预测帧的途径。解码器系统(500)的许多部件都用于解压关键帧和预测帧。由这些部件进行的具体操作可以根据所压缩信息的类型而变化。

一个缓存(590)接收用于一个压缩的视频序列的信息(595)，并使所接收的信息对熵解码器(580)可用。该缓存(590)一般按某个随时间基本不变的比特率来接收信息，并且包含一个抖动缓存，以平滑在带宽或传输中的短期变化。该缓存(590)可以包括一个回放缓存以及其它缓存。可替换的是，该缓存(590)按某种变化的速率接收信息。在缓存(590)之前或之后，可以将压缩的视频信息(495)按信道编码和处理，用于差错检测和纠正。

熵解码器(580)对熵编码的量化数据以及熵编码的伴随信息(如运动信息(515)、量化步长)熵解码，一般应用在熵编码器中进行的熵编码的反向操作。熵解码技术包括算术解码、微分解码、Huffman 解码、运行长度解码、LZ 解码、字典解码，以及以上这些的组合。熵解码器(580)经常对于不同的信息种类(如 DC 系数、AC 系数、不同种类的伴随信息)使用不同的解码技术，并且可以从多种解码表格中选择某种特定的解码技术。

如果要重建的帧(505)是一个前向预测帧，则一个运动补偿器(530)对一个参照帧(525)应用运动信息(515)，以形成对所重建的帧(505)的一个预测。举例来说，该运动补偿器(530)使用一个宏块运动矢量来找出一个在参照帧(525)中宏块。一个帧缓存(520)储存前面的重建帧，用作参照帧。可替换的是，一个运动补偿器应用另一种类型的运动补偿。由运动补偿器作的预测很少是完美的，因此解码器(500)还重建预测残留。

当解码器需要一个重建的帧用于后续运动补偿时，帧存储设备(520)缓冲该重建的帧，用于预测下一帧。在某些实施例中，解码器对重建的帧应用一

个解块过滤器，以适应性地平滑在该帧的块中的不连续。

一个反向量化器（570）将熵解码的数据反向量化。总的来说，反向量化器以一种在逐帧或其它基础上变化的步长来对熵解码的数据应用统一的标量反向量化。可替换的是，该反向量化器对这些数据应用另一种反向量化，例如不统一的、矢量的或非自适应性的量化，或者直接在一个不使用反向频率变换的解码器系统中反向量化空间域数据。

一个反向频率变换器（560）将量化的频域数据转换成空间域的视频信息。对于基于块的视频帧，该反向频率变换器（560）对 DCT 系数块应用一个反向 DCT（“IDCT”）或 IDCT 的变异，产生运动预测残留数据。可替换的是，该频率变换器（560）应用另一种传统的反向频率变换如 Fourier 变换，或使用小波或次波段综合。在某些实施例中，该反向频率变换器（560）对用于关键帧的空间预测残留块应用反向频率变换。该反向频率变换器（560）可以应用 8x8、8x4、4x8 或其它尺寸的反向频率变换。

3. 运动估计和补偿

帧间编码利用帧之间的时间冗余来达成压缩。时间冗余缩减在编码当前帧时使用以前编码的帧作为预测者。在下面描述的实施例中，视频编码器利用在典型视频序列中的时间冗余，以用较少数量的位来编码信息。视频编码器使用运动估计来参数化块、宏块或预测帧的其它像素集相对于参照帧（如某个以前编码的前面的帧）的运动。视频编码器（以及相应的解码器）使用运动补偿，用运动信息和参照帧来重建预测帧。

运动补偿是通过位移参照帧来为一个视频帧产生预测（即预测帧）的过程。如上面所记述的，预测是对来自参照帧的块、宏块或其它数据集形成的。而且，一般这一位移是在所预测的整个片上直线性和不变的。这一位移由一个运动矢量来定义，该运动矢量具有两个分量，对应于沿 X 和 Y 方向的位移或平移。X（水平）和 Y（垂直）运动矢量分量表示了当前所预测的片和在参照帧中相应的位置之间的位移。正值表示在当前位置的下方和右方的位置。负值表示在当前位置的上方和左方的位置。

在某一实现中，一个块是 8x8 的像素片，一个宏块是 16x16 的像素片，而

运动矢量定义至四分之一像素精确度。在其它实现中，编码器和解码器对于不同尺寸的片或任意变化尺寸的片，按不同的解析度或任意变化的运动矢量，并且/或者使用运动信息而不是运动矢量，来应用一种或多种所描述的技术。

运动矢量分量通常以像素位移，经常是以次像素精确度来说明。次像素位移是通过用恰当定义的运动补偿过滤器过滤参照帧来实现的。对于直线性次像素运动补偿的情况，X 和 Y 分量以固点数来表示。这些数的整数部分称为全像素位移，其小数部分称为次像素位移。当次像素位移为零时，运动是某整数个的像素。更经常的是，这实现为来自参照帧的一个块复制，以产生预测者（尽管在理论上，可以潜在地应用某种形式的过滤）。另一方面，当次像素位移不为零时，通过对在参照帧中整像素位置应用对应于次像素位移的一个或多个过滤器来产生预测者。因此，运动补偿过滤器是由次像素位移来决定的。

为了将次像素位移实现成过滤操作，运动补偿过滤器根据在整像素位置上的参考值来在小数的像素位置插值数据点。总的来说，插值的质量因过滤器的支持而提高。某些实施例使用对应于双线性和双立方插值器的可分离的（在各方向上）2 分支和 4 分支过滤器。

在某些实施例中，运动补偿过滤器使用整数算法和实现为一个移位的除法。取整控制参数 R 表现为值 0 或 1，并决定对这些除法的取整方向。这一取整控制参数可以设定成一个常数、从外部通知，或隐含地从以往的编码信息中取得。

图 6 显示了在某些实施例中的次像素运动估计和补偿期间使用的参照帧（600）中的整像素和次像素位置。在每个方向上以四分之一像素间隔，该参照帧（600）包含次像素位置，对该位置某个编码器或解码器可以为特定的位移而插值像素值。在图 6 中该参照帧（600）的整数位置 a-p 显示成阴影的圆圈，而插入这些整数位置之间的四分之一像素和半像素位置则显示成无阴影的圆圈。如在表格 1 中所描述的，位置 P_0 到 P_8 显示了九个代表性的次像素位置。

位置	描述
P_0	水平四分之一像素，垂直全像素。
P_1	水平半像素，垂直全像素。
P_2	水平全像素，垂直四分之一像素。

P_3	水平四分之一像素，垂直四分之一像素。
P_4	水平半像素，垂直四分之一像素。
P_5	水平全像素，垂直半像素。
P_6	水平四分之一像素，垂直半像素。
P_7	水平半像素，垂直半像素。
P_8	水平全像素，垂直四分之三像素。

表格 1: 代表性的次像素位置

在 P_3 例示的四分之三像素位置可以认为是四分之一像素位置的一个特例，它是从一个全像素位置移开四分之一像素。其它的四分之三像素也是可能的，但未示出。在稍后的插值过滤器的描述中将参照到这些次像素位置 P_0 至 P_8 。在可替换的实施例中，编码器和解码器在另外的或不同的次像素位置插值，例如在那些各个方向上四分之一像素以外的间隔。

A. 近似双立方插值过滤器

在某些实施例中对于次像素插值，视频编码器和解码器使用如以下所定义的线性/双线性和/或立方/双立方过滤器。

一个线性插值器是在一个维度上，利用在离所插值点最近的两个格点处的已知值的一个线性（或一阶）的多项式。线性函数在所插值点的值就是该线性插值。该线性多项式的乘数是通过求解一个线性系统方程，确定该线性过滤器的系数而确定的。一个线性插值过滤器由两个过滤器分支来定义的。一个双线性插值器就是一个分离在两个维度上的线性插值器。

一个立方插值器是在一个维度上，利用在离所插值点最近的四个格点处的已知值的一个立方（或三阶）的多项式。立方函数在所插值点的值就是该立方插值。该立方多项式的乘数是通过求解一个系统方程，确定该立方过滤器的系数而计算来的。一个立方插值过滤器由四个过滤器分支来定义的。一个双立方插值器就是一个分离在两个维度上的立方插值器。

术语线性和双线性在视频压缩和解压领域一般可以互换使用。在一个常规的两维插值中，在一个维度上进行的一个插值操作可以在另一个维度重复，因

此每个过滤阶段都称为双线性过滤。术语立方和双立方也可以类似地互换使用。

在本文中，术语线性和双线性互换使用来描述在一个、两个或多个维度上的过滤。类似地，术语立方和双立方也互换使用来描述在一个、两个或多个维度上的过滤。举例来说，公式(11)至(13)定义了立方过滤器的类型，但被称为双立方过滤器，因为在对于参照视频帧的两阶段插值的一般应用中，这些过滤器是在对于该两阶段插值的两个维度的重复操作中使用的。更为一般的是，过滤的维度从环境中得知。

在某些实施例中，编码器和解码器使用近似双立方过滤器来在次像素位置插值。举例来说，该编码器和解码器在如图6中所示的一个参照帧中可能的移动位置使用以下过滤器（其中 F_1 是一个双立方过滤器，而 F_2 和 F_3 是近似双立方过滤器）。

$$\text{半像素移位 } F_1: [-1 \ 9 \ 9 \ -1] \quad (11)$$

$$\text{四分之一像素移位 } F_2: [-4 \ 53 \ 18 \ -3] \quad (12), \text{ 和}$$

$$\text{四分之三像素移位 } F_3: [-3 \ 18 \ 53 \ -4] \quad (13)$$

在实用中，该过滤器还包括一个右移位（如对 F_1 是 4 位，对 F_2 和 F_3 是 6 位），以补偿由过滤器系数所潜在引入的展开。操作符 \gg 是一个右移位操作符。一个右移位操作将一个二进制数的位向右移动，将最低位丢弃并加上 0 作为最高位。这一操作获得一个简单的除法，即除以 2 的所移位个数次幂（如右移位 3 得到除 $2^3 = 8$ ），并将余数丢弃。

用于 F_2 和 F_3 的过滤器系数松散地基于真正的四分之一像素双立方插值器，该插值器也是一个四分支过滤器。以下的公式显示了对位置 P_0 应用该真正四分之一像素双立方过滤器的结果：

$$(-7e + 105f + 35g - 5h) \gg 7 \quad (14)$$

这些系数值总和为 128，而该过滤的结果右移了 7 位。如在以下的公式中所示的那样，近似双立方过滤器 F_2 和 F_3 在性能方面达到了纯的双立方过滤器，但具较低的解析度。

$$\begin{aligned} & (-7e + 105f + 35g - 5h) \gg 7 \\ & = (-3.5e + 52.5f + 17.5g - 2.5h) \gg 6 \end{aligned}$$

$$(-4e + 53f + 18g - 3h) \gg 6 \quad (15)$$

在许多情况下，使用纯的双立方过滤器会导致在多阶段插值中位精度的损失，因此用于近似双立方过滤器的正规化因子要减小至少 $1/2$ （即右移减少 1 位或更多）。对公式（15）的近似双立方过滤器的过滤器系数的选择基于对真正双立方过滤器取整，并考虑了频域特性（如保留高频信息）和经验特性（如在某比特率下达到最低失真）。特别的是，过滤器 F_2 和 F_3 仍然包括四个过滤器系数。（总的来说，在过滤器中使用较少的过滤器系数能加快实现的速度，但必须用足够的过滤器系数来解决相邻像素的噪声。）将过滤器系数的值调整到总和为 64，这能帮助用 16 位算术来实现并近似较高解析度的双立方过滤器。也可以使用总和为 64 的其它过滤器系数，仍能近似双立方过滤器。一个充分地象纯双立方过滤器一样进行，但具有较少的支持和/或较低的解析度的过滤器称为一个“近似”的双立方过滤器。一种客观地评价一个过滤器是否充分地象纯双立方过滤器一样进行的方法是检查该近似双立方过滤器是否与该纯双立方过滤器较好地相关（即在某个定义的阈值内）。在某个实施例中，相关性是通过这些过滤器的矢量间角度的余弦来测度的（希望尽可能接近 1），而阈值为 .95。也可以使用其它客观或主观的测度、其它相关性测度和/或阈值。举例来说，可以选择近似双立方过滤器的过滤器系数，使它们的总和为某个其它的能实现高效的 Fourier 变换或其他数学操作的值。

如下面所更充分描述的那样，图 7 从总体上显示了具有用以对在公式（11）-（13）中所列出的各种双立方过滤器情况计算插值像素的像素值的整像素位置。P 代表对其计算像素值的次像素位置。 I_1 、 I_2 、 I_3 和 I_4 表示沿着插值维度的整像素位置。图 7 显示了水平的插值，但同样的操作和位置排列也应用于垂直的插值。

在可替换的实施例中，编码器和解码器使用其它的和/或另外的插值过滤器。例如，编码器和解码器使用一个双线性（即 2 分支）过滤器来插值。举例来说，参照图 6 中的次像素位置，用以确定 P_1 、 P_5 和 P_7 的值的插值过滤器可以在公式（1）-（3）中所说明的那些过滤器。

B. 一维的插值

对于多种次像素位置，某些实施例的编码器和解码器只在一个维度上计算一个插值。如在图7中所说明的那样，以下的公式显示了在整像素间插值时过滤器 F_1 （半像素移位）、 F_2 （四分之一像素移位）和 F_3 （四分之三像素移位）的操作：

$$F_1: (-1I_1 + 9I_2 + 9I_3 - 1I_4 + 8 - r) \gg 4 \quad (16)$$

$$F_2: (-4I_1 + 53I_2 + 18I_3 - 3I_4 + 32 - r) \gg 6 \quad (17), \text{ 和}$$

$$F_3: (-3I_1 + 18I_2 + 53I_3 - 4I_4 + 32 - r) \gg 6 \quad (18)$$

其中值 r 控制取整。值 r 按以下这样取决于二进制帧级取整控制参数 R 和插值方向：

$$r = \begin{cases} 1-R & \text{(垂直方向)} \\ R & \text{(水平方向)} \end{cases} \quad (19)$$

为进一步说明一维的插值，图6的 P_1 和 P_5 显示了只在一个维度（即对 P_1 的水平方向和对 P_5 的垂直方向）上要求插值的在参照帧（600）中的半像素位置。以下的公式显示了在对 P_1 和 P_5 的整像素间插值时过滤器 F_1 （半像素移位）的操作：

$$P_1: (-1e + 9f + 9g - 1h + 8 - r) \gg 4 \quad (20), \text{ 和}$$

$$P_5: (-1b + 9f + 9j - 1n + 8 - r) \gg 4$$

(21)

类似地，图6的 P_0 和 P_2 显示了只在一个维度上要求插值的在参照帧（600）中的四分之一像素位置。以下的公式显示了在对 P_0 和 P_2 的整像素间插值时过滤器 F_2 （四分之一像素移位）的操作：

$$P_0 = (-4e + 53f + 18g - 3h + 32 - r) \gg 6 \quad (22), \text{ 和}$$

$$P_2 = (-4b + 53f + 18j - 3n + 32 - r) \gg 6 \quad (23)$$

该近似四分之一像素双立方过滤器 F_2 还可以只作一个小改动而用来计算四分之三像素位置。举例来说，以下的公式显示了在对 P_8 的整像素间插值时过滤器 F_3 （四分之三像素移位）的操作：

$$P_8 = (-3b + 18f + 53j - 4n + 32 - r) \gg 6 \quad (24)$$

可替换的是，编码器和解码器在一个维度上对半像素、四分之一像素或四分之三像素移位位置使用其它的和/或另外的插值过滤器。例如，编码器和解码器以更多或更少的过滤器系数、不同过滤器系数、不同的取整或不取整来使

用过滤器。

C. 多维的插值

在某些实施例中，对在两个维度上偏离的次像素位置进行插值。举例来说，在图 6 中， P_3 、 P_4 、 P_6 和 P_7 就是在水平和垂直两个维度上都发生插值的位置。

在某个与在图 8 中所示的插值方法 (800) 相应的实施例中，首先沿垂直方向，而后沿水平方向插值两维次像素位置。如在以下所更充分地描述的那样，插值是使用上面在公式 (16) 至 (18) 中所说明的过滤器 F_1 、 F_2 或 F_3 中的一个或多个来进行的。在图 8 所说明的实施例中，取整在垂直过滤和水平过滤后应用。在取整规则中的移位能够在中间结果中保持 16 位算术所允许的精度。

在图 8 中，首先进行垂直过滤，而后是水平过滤。从垂直过滤开始能在某些架构中改善性能。在其它的实施例中，过滤的顺序是不同的。例如，在垂直方向之前，在水平方向上进行插值。或者，使用插值过滤器的多种其它组合（如多个水平和/或多个垂直过滤器等）。

输入和输出像素的值 (811, 838) 具有 8 位的位深，并具有 256 个值的动态范围。中间值 (820) 具有 16 位的位深，用于有 65536 个值的动态范围。在可替换的实施例中，输入、输出和中间值具有不同（如更高）的位深。

在第一阶段 (810)，对 8 位输入像素值 (811) 应用一个适当的垂直过滤器 (F_v) (812)。所应用的垂直过滤器取决于所选择的次像素位置是移位四分之一像素、半个像素，还是四分之三像素，并可能采用上面所描述的双立方过滤器中某一个的形式。

在垂直过滤后的取整规则定义为：

$$(S + R_v) \gg \text{shiftV} \quad (25)$$

其中 S 是垂直过滤了的结果，而 $R_v = 2^{\text{shiftV}-1} - 1 + R$ 。R 是一个在帧到帧基础上于 0 和 1 之间交替的取整控制值。因此，该取整规则包含了以阶段交替取整控制 (813) 和一个移位 (814) 来取整。

向右移位会潜在地引起在解析度上的损失，因此将至少某些向右移位推迟到插值的较后阶段。向右移位值 shiftV 取决于所插值的次像素位置。具体来说，对于 P_3 、 P_4 、 P_6 和 P_7 $\text{shiftV} = \{5, 3, 3, 1\}$ 。移位的量小于为补偿由第

一阶段过滤器系数值的展开所要求的（如该移位小于用于近似双立方过滤器的6位），但足够保证后续过滤的中间结果保持在中间值的动态范围（如对于16位字的65536个可能的值）内。与完全移位相比，这一缩短了的移位保持了在第一阶段插值（810）后的中间像素值（820）的精度。中间像素值（820）具有 y 位的动态范围，其中 y 大于8位。在第一阶段所进行的移位的量取决于可用的位深和插值过滤器的系数。举例来说，在这里所描述的示例性实现中，中间值受限于16位的字长限制。

请考虑图6的点 P_3 和在0...255范围（8位）中的输入值。由于来自过滤器系数的展开因子，对8位输入值应用近似双立方过滤器系数 $[-4\ 53\ 18\ -3]$ 所得到的中间值范围是从-1785至18105（大约14.3位，为具体实施而向上取整到15位）。对该中间值应用（具额外展开的）近似双立方过滤器系数的后续水平过滤可能产生16位动态范围以外的值，引起上溢。因此，要将该中间值足够地移位，以保证后续水平过滤产生在16位动态范围内的某个值。对于 P_3 ，第一次移位的数量是5位，则移位了的中间值的动态范围是从-55至565（约为9.3位，为具体实施而向上取整到10位）。则对该移位了的中间值应用近似双立方过滤器系数的输出的范围是从-7860至40500，具有小于16位的动态范围。如此，缩短的移位就按这样计算，能保证在插值的第二阶段（830）充分利用但不超出16位字的限制。

在第二阶段（830），应用一个适当的水平过滤器（ F_H ）（832），以从该由垂直过滤器确定的值（820）在该二维次像素位置插值。在水平过滤后的取整规则为：

$$(S + 64 - R) \gg 7 \quad (26)$$

其中 S 是水平过滤了的结果，而 R 是一个在帧到帧基础上交替的取整控制。象第一阶段取整规则一样，第二阶段取整规则包含了以阶段交替取整控制（833）和一个移位（834）来取整。由于在第一阶段的推迟移位，在第二阶段中的移位数量一般会超出对所选定水平过滤器正常所期望的，而计算来输出具有所希望的动态范围的某个值。

所有的双立方过滤情况都可能潜在地产生一个其值是负的或大于范围最大值（如对8位输出为255）的插值后的像素。在这些（8位输出值）情况下，

编码器和解码器将该输出值（836）箝位在所接受的范围内。具体来说，下溢设定为 0，而上溢设定为 255。在箝位后，输出一个插值后的 8 位值（838）。

在图 8 中，第二阶段移位 7 位。这样，就保持了一个 9 位的过滤了的输出值。举例来说，继续前面 P3 的示例，过滤了的输出值的范围从 -61 至 316，具有约为 8.6 位（为具体实施而向上取整到 9 位）的动态范围。尽管该插值后数据的有效范围只有 8 位，但头上额外的一位提供了上溢和下溢的信息。换句话说，如果设置了最高位（即“符号”位），就存在下溢或上溢。这两种中具体发生哪一种，可通过观察余下的 8 个“尾数”位来推出。

图 9-11 进一步说明了上面所描述并在图 8 中说明的两维插值。图 9 显示了图 6 的参照帧（600）的次像素位置 P_7 （水平半像素，垂直半像素）。用两个半像素双立方插值过滤器来插值 P_7 的值。在第一阶段，用具有以下一般形式的一个半像素双立方过滤器从最近的整像素位置计算中间值 V_1-V_4 ：

$$V_{\text{inter.}} = (-1x_1 + 9x_2 + 9x_3 - 1x_4) \quad (27)$$

如此，

$$V_1 = (-1a + 9e + 9i - 1m) \quad (28)$$

$$V_2 = (-1b + 9f + 9j - 1n) \quad (29)$$

$$V_3 = (-1c + 9g + 9k - 1o) \quad (30), \text{ 和}$$

$$V_4 = (-1d + 9h + 9l - 1p) \quad (31)$$

在加上适当的 R_v 值后，将结果向右移位一位。在第二阶段中，由一个半像素过滤器用中间结果 V_1-V_4 来计算在 P_7 的像素值。具体来说，使用一个具有以下形式的半像素过滤器：

$$P_7 = (-1V_1 + 9V_2 + 9V_3 - 1V_4) \quad (32)$$

如上面所讨论的，第二阶段的结果向右移位七位，以获得一个 9 位的值。该 9 位的值包括八个尾数位和一个符号位。在进行某些必要的箝位以对溢或下溢作出补偿后，就输出了一个最终的 8 位插值。

图 10 显示了图 6 的参照帧（600）的次像素位置 P_4 （水平半像素，垂直四分之一像素）。用一个四分之一像素和一个半像素双立方插值过滤器来插值 P_4 的值。在第一阶段，用具有以下一般形式的一个四分之一像素双立方过滤器从最近的整像素位置计算中间值 V_1-V_4 ：

$$V_{inter.} = (-4x_1 + 53x_2 + 18x_3 - 3x_4) \quad (33)$$

这一过滤器以与上面关于计算 P_7 所描述的同样方式应用于参照帧 (600) 的整像素值。在加上适当的 R_v 值后, 将结果向右移位三位。在第二阶段中, 由一个半像素过滤器用这些中间结果 V_1-V_4 来计算在 P_4 的像素值。具体来说, 使用一个具有以下形式的半像素过滤器:

$$P_4 = (-1V_1 + 9V_2 + 9V_3 - 1V_4) \quad (34)$$

第二阶段的结果向右移位七位以获得一个 9 位的值, 进行某些必要的箝位, 并输出一个最终的 8 位插值。

图 10 还显示了一个次像素位置 P_6 (水平四分之一像素, 垂直半像素)。为了插入 P_6 的值, 以微小的修改来使用用于插值 P_4 的技术。在修改后的技术中, 在第一阶段用一个半像素双立方过滤器来确定中间值。在图 10 中 V_5-V_8 处显示了中间像素位置。在第二阶段中, 一个四分之一像素双立方过滤器用这些中间值来计算 P_6 的值。具体来说, 使用一个具有以下形式的四分之一像素双立方过滤器:

$$P_6 = (-4V_5 + 53V_6 + 18V_7 - 3V_8) \quad (35)$$

在第一和第二阶段中移位的数量与在用于计算 P_4 的技术中的相同(即第一阶段移位为三, 而第二阶段移位为七)。

图 11 显示了图 6 的参照帧 (600) 的一个次像素位置 P_3 (水平四分之一像素, 垂直四分之一像素)。用两个四分之一像素双立方插值过滤器来插值 P_3 的值。在第一阶段, 用具有以下一般形式的一个四分之一像素双立方过滤器从最近的整像素位置计算中间值 V_1-V_4 :

$$V_{inter.} = (-4x_1 + 53x_2 + 18x_3 - 3x_4) \quad (36)$$

这一过滤器以与上面关于计算 P_4 所描述的同样方式应用于参照帧 (600) 的整像素值。在加上适当的 R_v 值后, 将结果向右移位五位。在第二阶段中, 由另一个四分之一像素过滤器用这些中间结果 V_1-V_4 来计算在 P_3 的像素值。具体来说, 使用一个具有以下形式的四分之一像素过滤器:

$$P_3 = (-4V_1 + 53V_2 + 18V_3 - 3V_4) \quad (37)$$

第二阶段的结果向右移位七位以获得一个 9 位的值, 进行某些必要的箝位, 并输出一个最终的 8 位插值。

尽管在图 9-11 中没有示出，也可以计算在一个或两个维度上具有四分之三像素移位的次像素位置的值。为了计算这样的次像素位置，可以通过用适当的四分之三像素双立方过滤器代替四分之一像素双立方过滤器，来修改上面所列出的方法。

在其它实施例中，用双线性过滤器或双线性和双立方过滤器的组合来在次像素采样位置插值。双线性过滤器的使用一般会减少（在第一阶段后和总体）要进行的移位的数量，因为其系数引入的展开小于双立方过滤器的。举例来说，在某个使用双线性过滤器和 16 位中间值的实现中，在第一阶段中不进行移位，因而最大化了对 16 位字限制的利用，而在最后的阶段后进行 4 位的右移位。类似地，箝位也可以推迟到最后阶段。

潜藏于上面所描述的方法下的原则之一是在保持在某个所希望的“字尺寸”限制 W 内的同时，在过滤的每个阶段利用最高的可能精确度。如果输出值具有 D 位的动态范围，而在最后阶段丢弃了 L 位，则最后阶段过滤的输出可以采用总共 $D + L + 1$ 位，其中将一个额外的位用于标志下溢和上溢。向后推，如果最后阶段过滤导致 k 位的展开，则向该最后阶段的输出应当在 $D + L - k$ 内。因此，为了以一个 W 位表示来保持最大的精确度，存在以下的关系：

$$D + L + 1 = W \quad (38)$$

而且，向最后阶段的输出应当为 $D + L - k = W - k - 1$ 位。

上面的逻辑可以递归地应用于过滤的倒数第二阶段，以及诸如此类。实际上，可以通过使用小数位来加紧限制，以表示非 2^k 的范围和展开因子。

图 12-15 是说明多种技术的图，这些技术以上面组合描述，但也能单独应用于多阶段插值。图 12-15 没有显示各多阶段插值（1200、1300、1400、1500）可与其它多阶段插值技术一起使用的多种方式。

而且，尽管图 12-15 中的每一幅都显示了两个阶段，但在图 12-15 中所说明的多阶段插值（1200、1300、1400、1500）技术可以包括更多个阶段。更为普遍的是，这些多阶段插值（1200、1300、1400、1500）技术可以以在多个维度上任何种类的可分离过滤器，以及在某种层叠、格架或栅格结构中实现的任意过滤器来实现。

图 12-15 显示了在多阶段插值中使用的普遍化了的输入值、输出值和过滤

器。第一阶段输入值、最后阶段输出值和中间值的位深的特定选择可以按照目标架构或应用的技术规范而任意地扩展。举例来说，输入值可以是在参照帧中整像素位置的 8 位像素值，输出值可以是在参照帧中次像素位置的 8 位像素值，而过滤器可以是（如上面参照图 6-8 所描述的）规范的或近似的双立方过滤器。可替换的是，输入值和/或输出值可具有不同位深的动态范围，或者可使用不同的过滤器。

如参照图 4 和 5 所分别描述的编码器或解码器那样的一个部件可以进行该多阶段插值（1200、1300、1400、1500）。可替换的是，另一个编码器或解码器（或另一种部件）也可以进行该多阶段插值（1200、1300、1400、1500）。

图 12 显示了一幅中间插值具有增强的动态范围（以位为单位）的多阶段插值（1200）的图。在第一阶段（1210），该部件对一个或多个 x 位范围的输入值（1211）应用（1212）第一个过滤器 F_1 ，产生一个或多个 y 位范围的中间值（1220），其中 y 大于 x 。举例来说， y 位的中间值是具有大于 8 位的动态范围的像素，而 x 位的输入值具有 8 位的动态范围。

在零个或多个中间阶段（1222）（未详细示出）中的每一个阶段，该部件对 y 位范围的中间值（1220）应用一个过滤器。从该中间阶段的输出是一个或多个 z 位范围的中间值（1229），其中 z 大于 x 。（在图 12-15 中，如果最后阶段是第二阶段，则从第一阶段的中间值输出就是对最后阶段的输入中间值。）

在最后阶段（1230），该部件对该一个或多个 z 位范围的中间值（1229）应用（1232）一个最后的过滤器 F_L 。最终的输出是一个 x 位范围的输出值（1234）。如果必要，对于每种多阶段插值（1200、1300、1400、1500）技术，该部件将为另外的输出值而重复该多阶段插值（1200、1300、1400、1500）。在所重复的插值中，该部件可以重用某些在前面的插值中计算的中间值。

图 13 显示了一幅跳过了箝位的多阶段插值（1300）的图。举例来说，由于该部件不再将每个中间值与范围的上或下边界进行检查，推迟箝位就加快了计算的速度。推迟箝位还在中间值中保持了精度。

在第一阶段（1310），该部件对一个或多个 x 位范围的输入值（1311）应用（1312）第一个过滤器 F_1 。在应用第一个过滤器 F_1 后，不再进行箝位。这样，从第一个过滤器 F_1 的一个或多个中间值（1320）输出可以具有大于 x 位的动态

范围。举例来说，输入值为 8 位值，而从第一个过滤器 F_1 的输出由于由第一个过滤器 F_1 系数引入的展开因子而具有 9 位或更多位的动态范围。

在零个或多个中间阶段（1322）（未详细示出）中的每一个阶段，该部件对该一个或多个未箝位的中间值（1320）应用一个过滤器。在该零个或多个中间阶段（1322）中箝位也可以跳过。将从该零个或多个中间阶段的中间值（1329）输出输入到某个最后阶段（1330），在其中该部件对这些值（1329）应用（1332）最后的过滤器 F_L 。箝位（1334）从该最后的过滤器 F_L 的最终输出，并输出一个 x 位范围的值（1336）。

图 14 显示了一幅推迟移位的多阶段插值（1400）的图。在第一阶段（1410），该部件对一个或多个 x 位范围的输入值（1411）应用（1412）第一个过滤器 F_1 。在应用第一个过滤器 F_1 同时或之后，进行一个缩短的移位（1414）。该缩短的移位（1414）小于（考虑到第一过滤器 F_1 系数的展开因子）为保证一个 x 位范围输出值所要求的，因而也小于一般与该第一个过滤器 F_1 所关联的。因此，该缩短的移位（1414）产生具有大于 x 位的（ y 位）动态范围的一个或多个中间值。举例来说，输入值具有 8 位的动态范围，而中间值具有大于 8 位的动态范围。

在零个或多个中间阶段（1422）（未详细示出）中的每一个阶段，该部件对该一个或多个中间值（1420）应用一个过滤器。从该零个或多个中间阶段（1422）输出具有 z 位（大于 x 位）动态范围的一个或多个中间值（1429），而在最后阶段（1430），该部件对这些值（1429）应用（1432）最后的过滤器 F_L 。从该最后的过滤器 F_L 的最终输出移位某个大于一般与该最后过滤器 F_L 相关联的数量，从而使输出值（1434）的动态范围符合某个特定的位深。举例来说，该输出值（1434）的（位）动态范围等于 x 或 $x + 1$ 。在某个实现中，第一阶段和任何中间阶段的移位尽量推迟到最后阶段。推迟移位的数量可以取决于中间计算的可用位深和相应过滤器的展开因子。

图 15 显示了一种使用阶段交替取整控制的多阶段插值技术（1500）。在该插值技术（1500）中的多个阶段交替它们应用取整控制的方法，以对取整进行调整。这帮助防止了在某些视频序列中取整误差从帧到帧累积。举例来说，如果某个低质量的视频序列包含在一个维度（盘面）或两个维度（升缩）上的

逐渐运动，则取整误差的累积可能导致从帧到帧的逐渐的色彩减弱，这可以造成可察觉的伪像。阶段交替的取整控制帮助防止了这样的色彩减弱。

许多示例可以帮助说明在向右移位前应用阶段交替取整控制的取整。向右移位本质上会导致右移位的值的除法和切断。在移位前加上一个取整值使移位的值能被向上或向下取整到最近的整数，而不是总被向下取整（切断）。使用取整控制改变了对边际值取整的方向（向上或向下）。举例来说，假设在多阶段的每一个中，过滤的输出通过在向右移位前加上右移位的“约数”的 $1/2$ 来调整（如在向右移位 5 位前加上 $2^4 = 16$ ，在向右移位 7 位前加上 $2^6 = 64$ ）。这样的增加的效果是将值向上取整（到下一个较大的整数），该值本来（在对应于该移位的除法后）会有一个.5 或更大的小数部分。在其他情况下这样的值会被该右移位切断（到下一个较小的整数）。如果没有这一增加，本来（在对应于该移位的除法后）会具有小于.5 的小数部分的值也仍然会被该右移位切断（到下一个较小的整数）。而后取整控制改变对某些边际值取整的方向。举例来说，在多阶段的每一个中，过滤的输出在向右移位前以减去 0 或 1（交替的取整控制值）来进一步调整（如 $2^{\text{shiftV}-1}$ 或 $2^{\text{shiftV}-1} - 1$ ）。取整控制调整的效果是改变对（在对应于该移位的除法后）会具有.5 的小数部分的值取整的方向。在减去 1 时，这样的边际值被向下取整。否则，这样的边际值被向上取整。

多个阶段中的每一个在多阶段插值前使用一个在 0 与 1 之间交替的取整控制值，这样这些不同的阶段就交替应用取整控制值的方法。可替换的是，多阶段插值技术（1500）使用一个其本身从阶段到阶段交替的取整控制值。

在图 15 的第一阶段（1510）中，该部件对一个或多个 x 位范围的输入值（1511）应用（1512）第一个过滤器 F_1 。在应用第一个过滤器 F_1 同时或之后，对来自第一个过滤器 F_1 的输出进行取整（1514）。该取整（1514）由阶段交替取整控制来调整。举例来说，在第一阶段（1510），如果输出值是一个边际值，该阶段交替取整控制使该输出值被向上取整到最近的整数（在其他情况下该输出值会被向下取整）。将一个或多个取整了的中间值（1520）从第一阶段输出进第二阶段（1530）。

在第二阶段（1530）中，该部件对一个或多个中间值（1520）应用（1532）第二个过滤器 F_2 。对来自第二个过滤器 F_2 的输出进行取整（1534）。在应用第

二个过滤器 F_2 同时或之后，以阶段交替取整控制来进行取整（1534），该取整控制实现在与第一阶段对边际值相反方向上的取整。举例来说，在第二阶段（1530），如果输出值是边际值，该阶段交替取整控制使该输出值被向下取整到最近的整数。从第二阶段输出一个或多个中间值（1536），并可能在零个或多个另外的阶段（1540）使用。该零个或多个另外的阶段（1540）可以进一步包括阶段交替取整控制。

交替的取整控制不限于在连续的阶段中应用，但可以在这些阶段的多个其它组合中应用。而且，第一个方向可以取决于许多参数。举例来说，在一个视频编码器和解码器中，第一个方向可以取决于在前一帧中或在所插值的帧的类型（如 I 帧、P 帧或 B 帧）上使用过的取整控制。在其它实施例中，第一个方向可以设定为某个从原因信息中（如从以往的编码/解码信息中）隐含地获得的、用某个伪随机数生成器获得的、或被标记成比特流的一部分的常数。阶段交替取整控制可以应用于使用多种插值过滤器（包括双线性、双立方和近似双立方过滤器）中任何一种的多阶段插值中。

D. 色度运动矢量

由于色度运动矢量是从同位置的亮度运动矢量中隐含地获得的，因此其精确度有限，只提供简化的范围。这样的简化可能降低在编码器和解码器中对色度值次像素插值的计算复杂度，而不会显著降低所编码视频的可察觉质量。而且，编码器和解码器可以在用于色度运动矢量取整和插值的不同模式间切换。举例来说，一种模式以更大计算复杂度的代价来强调所编码视频的质量。另一种模式则以某些质量代价来强调计算的简化。

在某个实现中，一种视频编码器和解码器使用一个顺序层次的 1 位域“FASTUVMC”来控制对色度值的次像素插值和色度运动矢量的取整。这样，该视频编码器和解码器就有选择性地以两种不同色度取整模式（一种快速模式和一种基本模式）中的一种来运作。

图 16 显示了一种用于在多种色度取整和插值模式中进行选择的技术（1600）。举例来说，一种如上面参照图 4 和 5 所分别描述的视频编码器和解码器实现该技术。

该视频编码器和解码器确定一个一位标志 FASTUVMC 是指示使用一种快速色度运动补偿模式（标志 = 1）还是基本色度运动补偿模式（标志 = 0）。举例来说，该标志是一个顺序层次的域，对应于某个用户设定，该域由一个编码器写入所编码视频的比特流中，并由一个解码器从该比特流中读取。可替换的是，该编码器和解码器使用定长或变长编码，用多个位来标志一个色度取整和/或一个插值模式，举例来说，以在多于两个的可用模式中进行选择。或者，代替对应于某个用户设定的一个顺序层次域，切换信息在比特流中的某处标记，并/或根据不同的条件来设定。

而后该视频编码器和解码器以基本模式（1620）或快速模式（1630）来进行色度运动补偿。基本模式（1620）和快速模式（1630）运动矢量取整和插值的细节将在下面对某个实现来给出。可替换的是，这些模式都具有不同的实现。举例来说，在下面所描述的快速模式（1630）实现中使用的检索表可改变为某个不同的映射，以对某种特定的硬件架构得到某种所希望的性能水平，或者可改变为对不同精确度的运动矢量都可用。代替或除了基本模式（1620）和快速模式（1630）以外，编码器和解码器还可以使用其它的用于色度运动矢量取整和插值的模式。

在某个实现中，在快速模式中，（如若色度取整标志 = 1）在四分之一像素偏移处（即四分之一像素和四分之三像素偏移）的色度运动矢量取整到最近的全像素位置，在半像素偏移处的色度运动矢量向左反取整，并对所有的色度插值使用双线性过滤。在这一模式中，提高了编码器和解码器的速度。这一优化的动机是在（a）整像素位置；b）半像素位置；c）对于（x 和 v 的）至少一个坐标的四分之一像素位置及 d）对于两个坐标的四分之一像素位置处插值像素偏移的复杂度之间的显著差异。a:b:c:d 的比例约略为 1:4:4.7:6.6。通过应用这一快速模式，可以偏好（a）和（b），从而减少解码时间。因为这只对色度插值进行，因此编码和质量损失（特别是可视质量）是都可以忽略的。

在这一快速模式中，按以下对色度运动矢量实现最后层次的取整：

$$\begin{aligned}
 //\text{RndTbl}[-3] &= -1, \text{RndTbl}[-2] = 0, \text{RndTbl}[-1] = +1, \text{RndTbl}[0] = 0 \\
 //\text{RndTbl}[1] &= -1, \text{RndTbl}[2] = 0, \text{RndTbl}[3] = +1 \\
 \text{cmv}_x &= \text{cmv}_x + \text{RndTbl}[\text{cmv}_x \% 4];
 \end{aligned}
 \tag{39}$$

$$\text{cmv_y} = \text{cmv_y} + \text{RndTbl}[\text{cmv_y} \% 4];$$

其中 cmv_x 和 cmv_y 以四分之一像素单位表示色度运动矢量的 x 和 y 坐标，而 $\%$ 表示取模（或取余）操作，其以如此来定义： $(x \% a) = -(-x \% a)$ 。（一个负数的模等于其相应正数的模的负数。）因此，当 cmv_x （或 cmv_y ）被 4 可除时，则色度运动矢量具一个整数位移。当 $\text{cmv_x} \% 4 = +/-2$ 时，则色度运动矢量具有一个半像素位移。当 $\text{cmv_x} \% 4 = +/-1$ 或 $+/-3$ 时，则色度运动矢量具有一个四分之一像素位移。如通过上面的重新映射操作所可以看到的那样，四分之一像素位置是将色度运动矢量取整到最近的整数位置所不接受的（半像素位置是不能向左变更的）。这样，这一模式就将色度坐标重新映射到整数或半像素位置。双线性过滤在这一模式中可以用于所有的色度插值，以便进一步提高速度。尽管这一快速模式实现是结合多种取整模式中的选择一起描述的，但该快速模式实现也可以独立使用（即作为唯一可能的模式）。

图 17 是一个说明色度取整的这一快速模式的表格（1700）。第一行（1710）以四分之一像素的精确度显示了亮度运动矢量值。尽管这些亮度运动矢量值是以其从整像素值的小数位移来显示的，但它们也可以整数来表示，其中每个整数表示一个四分之一像素的增量（即 0、1、2、3、4 替代 0、1/4、1/2、3/4、1）。第二行（1720）显示了相应的色度运动矢量值如何在快速模式中取整，以使它们具有整数和半像素的精确度。

在这一实现的第二个基本模式中（如若色度取整标志 = 0），在四分之一像素偏移处的没有取整误差的色度运动矢量仍然在四分之一像素偏移处。在其它次像素位移的色度运动矢量则取整到最近的全像素或半像素位置。在这一模式中，解码器的速度可能会比其它模式慢，但计算色度像素值的精确度较高。这样，这一基本模式就将色度坐标重新映射到整数、半像素和四分之一像素位置。如上面所描述的双立方或双线性过滤可以用于色度插值。

图 18 是一个说明色度取整的这一基本模式的表格。第一行（1810）以四分之一像素的精确度显示了亮度运动矢量。第三行（1830）显示了相应的色度运动矢量值如何在上面所描述的基本模式中取整，以使它们具有整数、半像素和四分之一像素的精确度。在其它实现中，色度空间取整到色度空间中的其它解析度。

在参照多个实施例描述和说明了我们的发明的原理后，可认识到这多种实施例可以在设置和细节上改动，而不背离这些原理。举例而言，上面所描述的原理和技术不限于在视频编码器和/或视频解码器中使用。可替代的是，它们可以在其中部分地根据一个或多个中间值来计算某个值或其中在多个维度上使用可分离的过滤器的任何计算环境中应用。

应当理解在此所描述的程序、过程或方法并不关于或限于任何特定种类的计算环境，除非指明。多种通用或专用的计算环境都可以按照在此所描述的教义来使用或进行操作。以软件显示的实施例元素可以以硬件来实现，反之亦可。

考虑到我们的发明所可能应用的许多可能的实施例，我们将所有可能在以下的权利要求及其等价物范围和精神内的这些实施例主张为我们的发明。

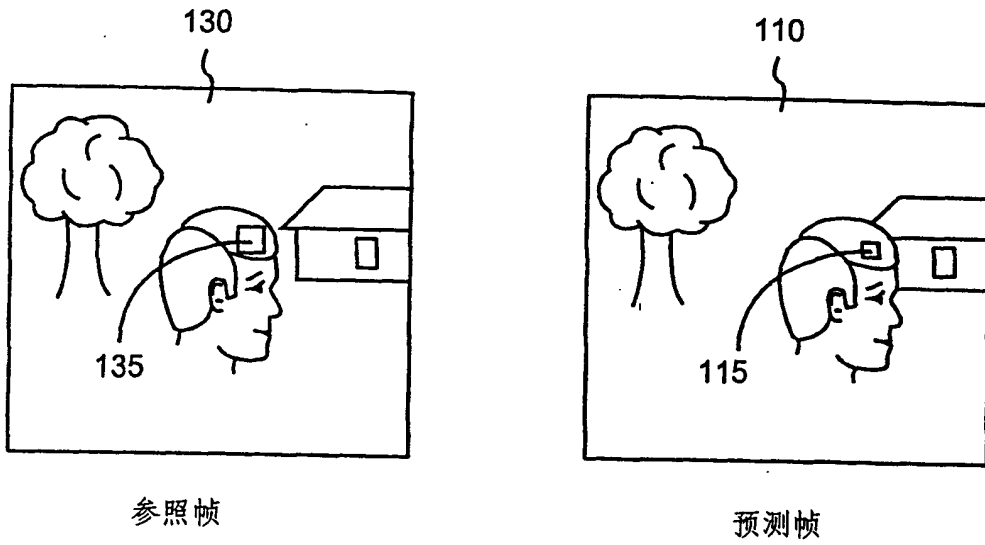
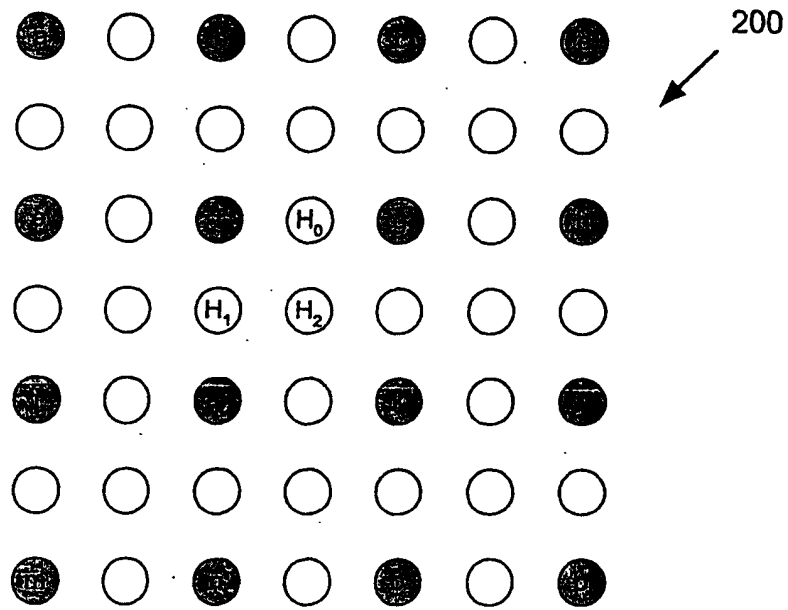


图 1
现有技术



- a, b, c, ..., p 代表整像素位置
- H_0, H_1, H_2 代表半像素位置

图 2
现有技术

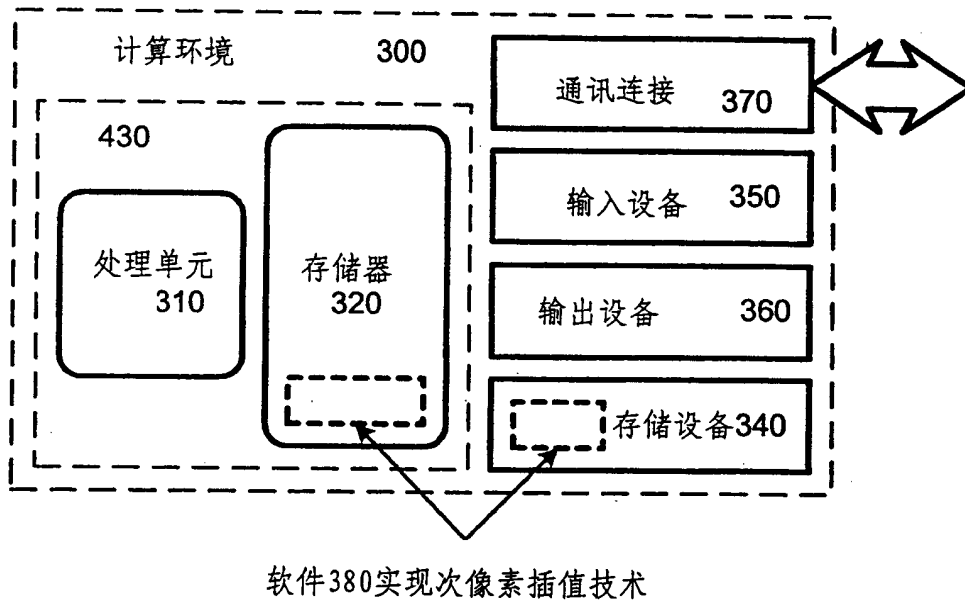


图 3

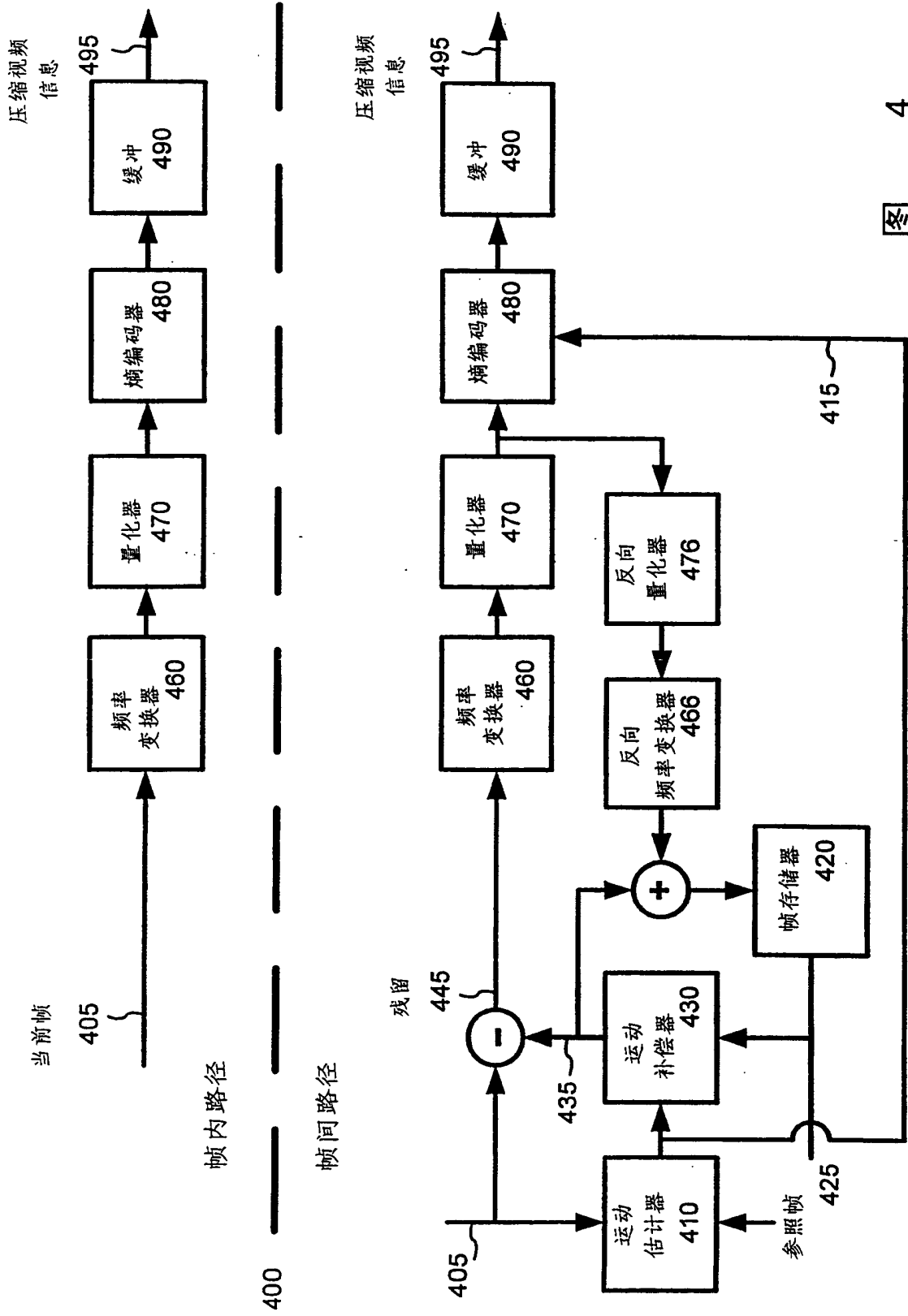
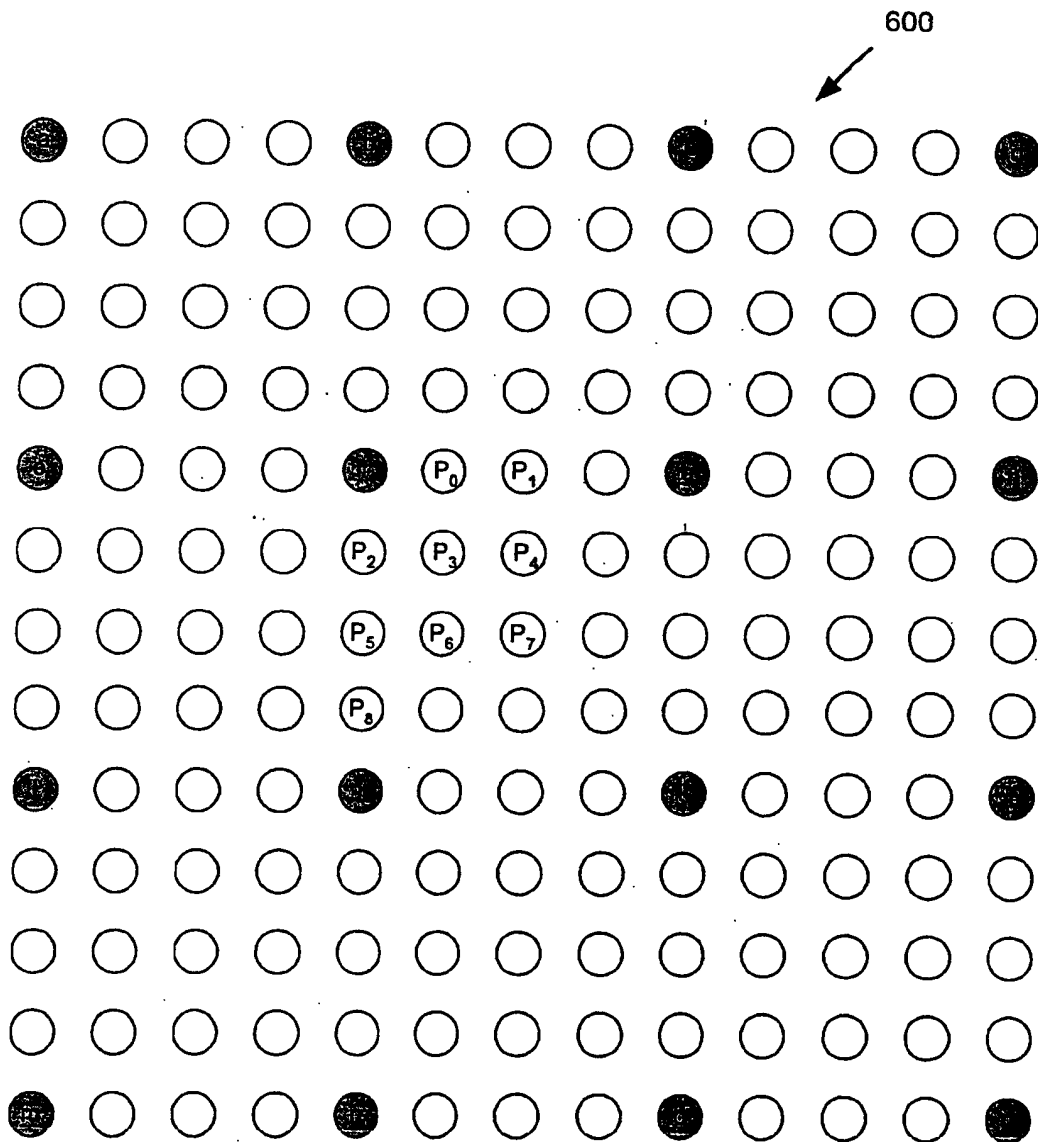


图 4



- 每个圆圈代表一个四分之一像素间隔的像素位置
- a ... p代表整像素位置
- P₀...P₈ 代表示例性次像素位置

图 6

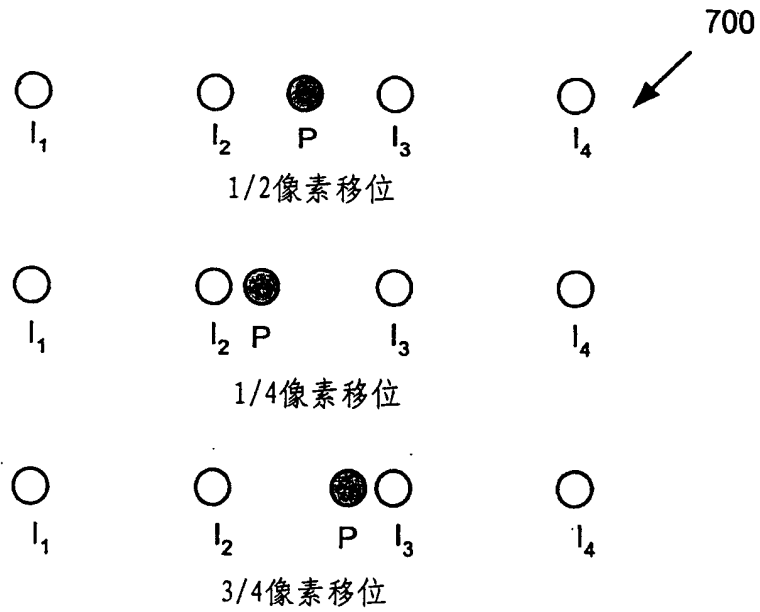


图 7

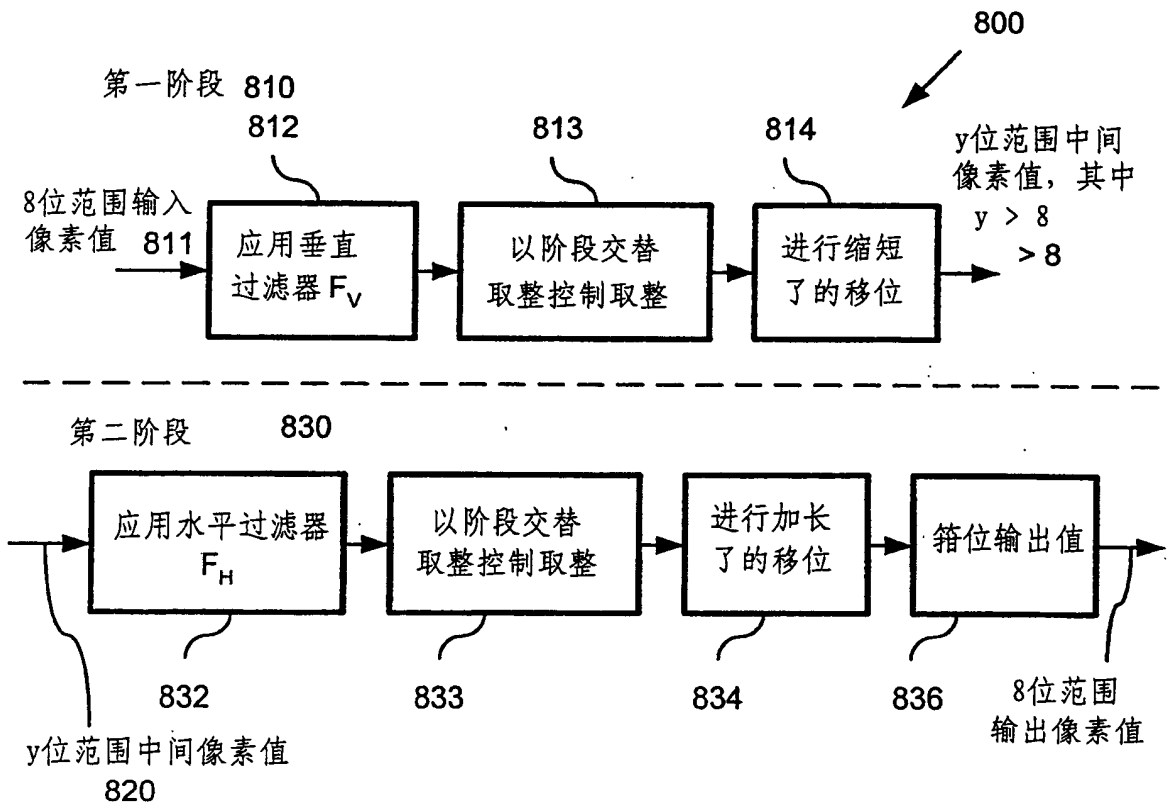


图 8

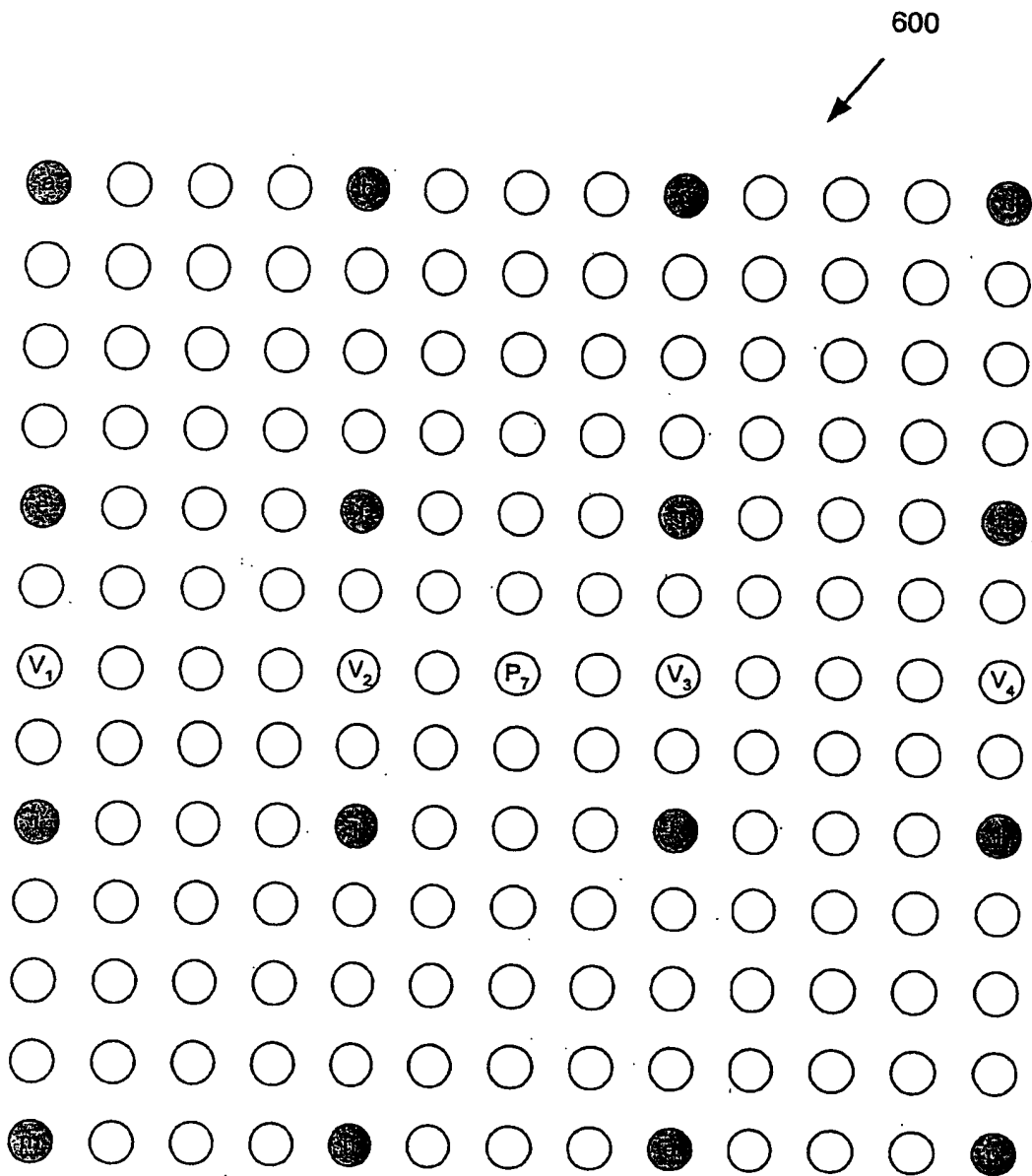


图 9

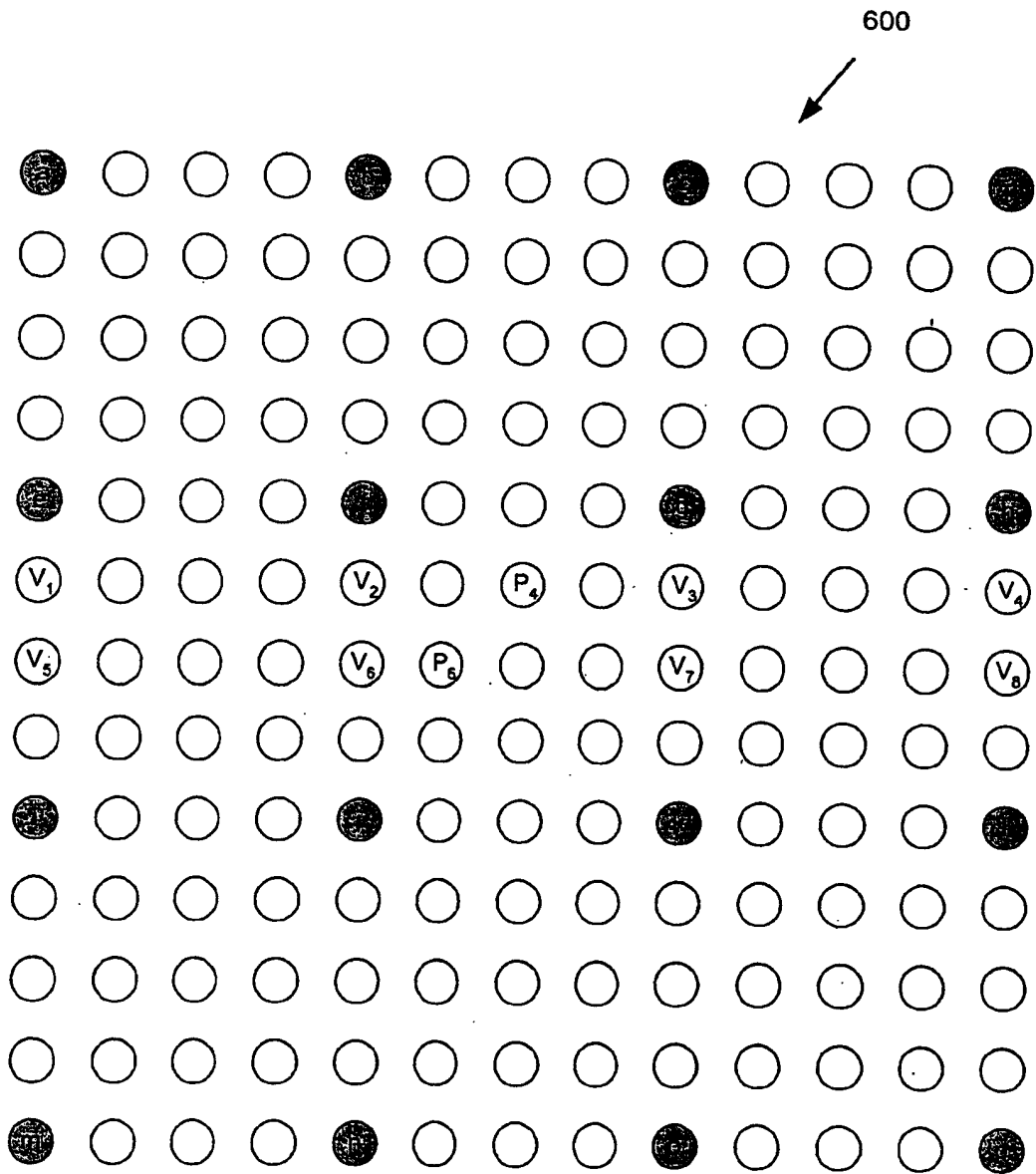


图 10

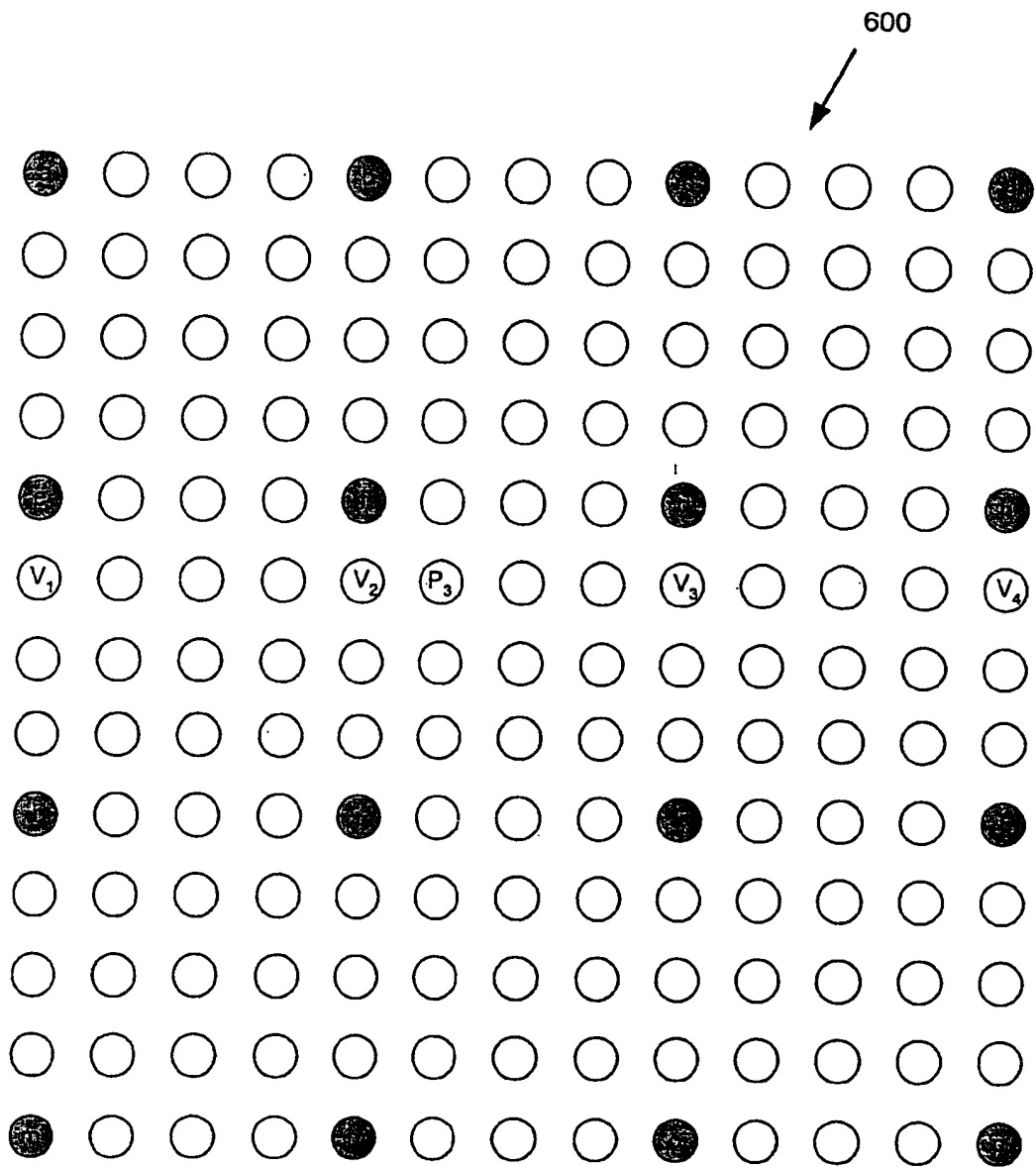


图 11

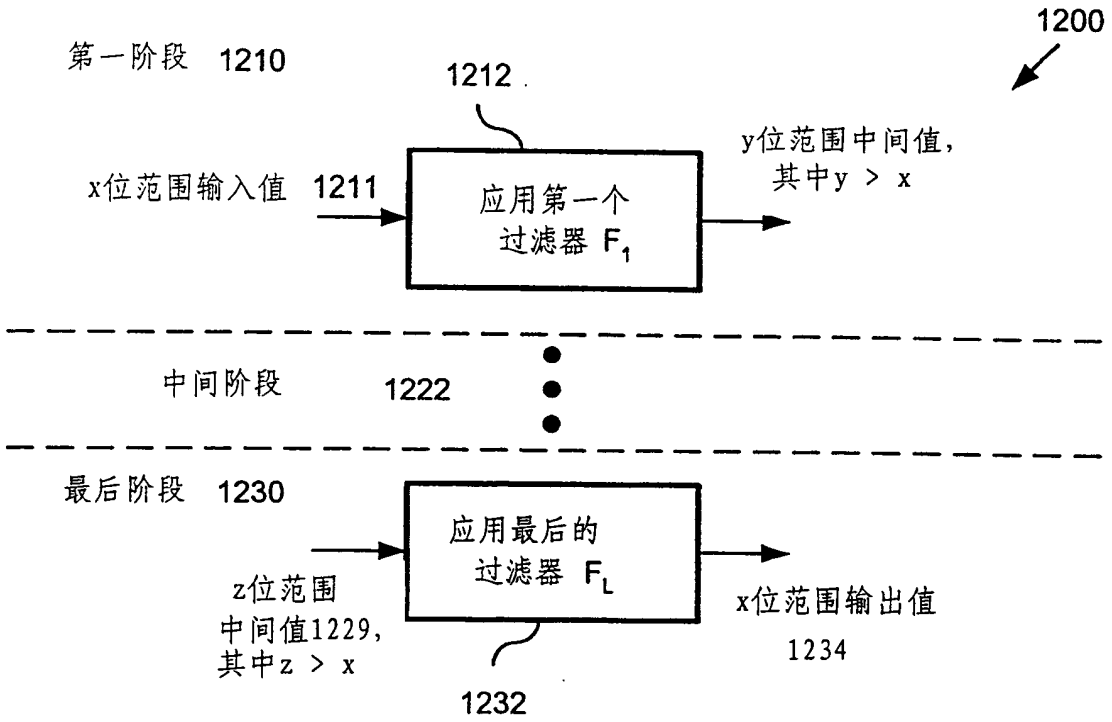


图 12

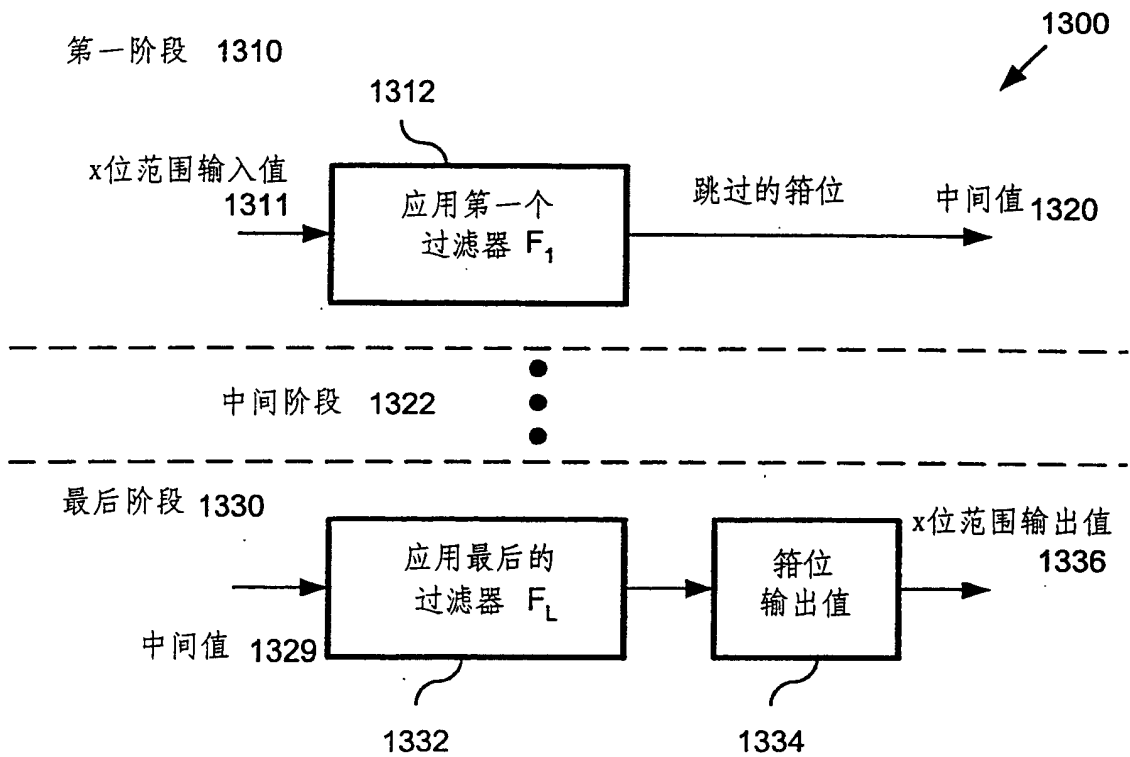


图 13

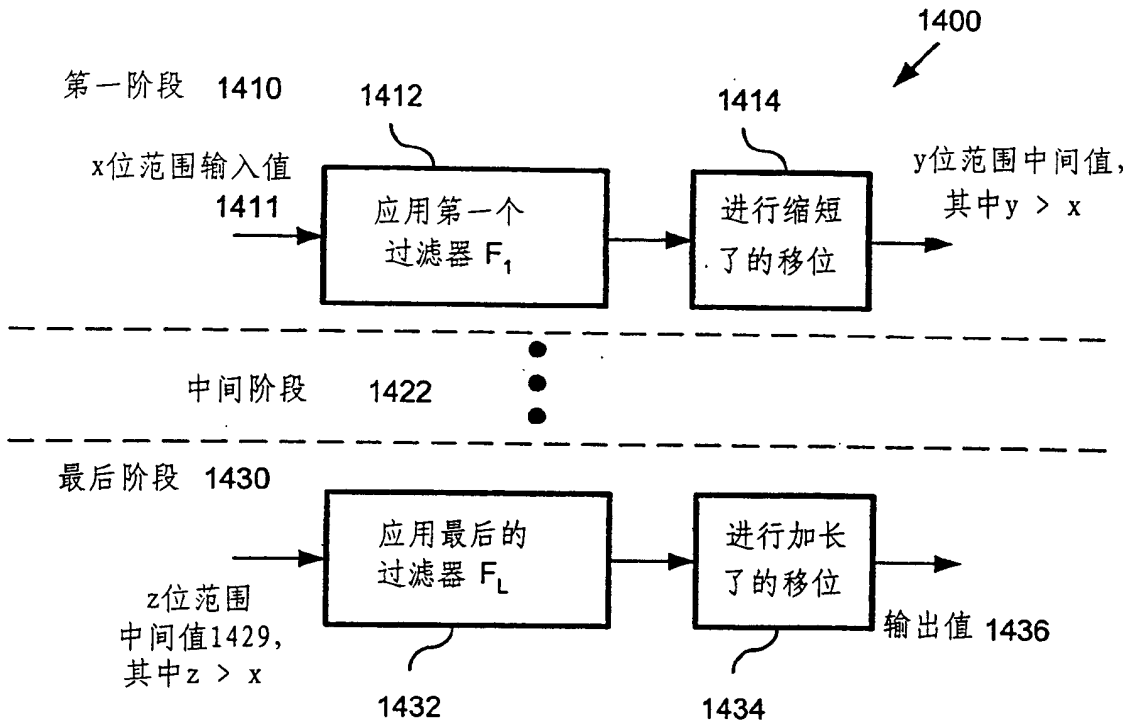


图 14

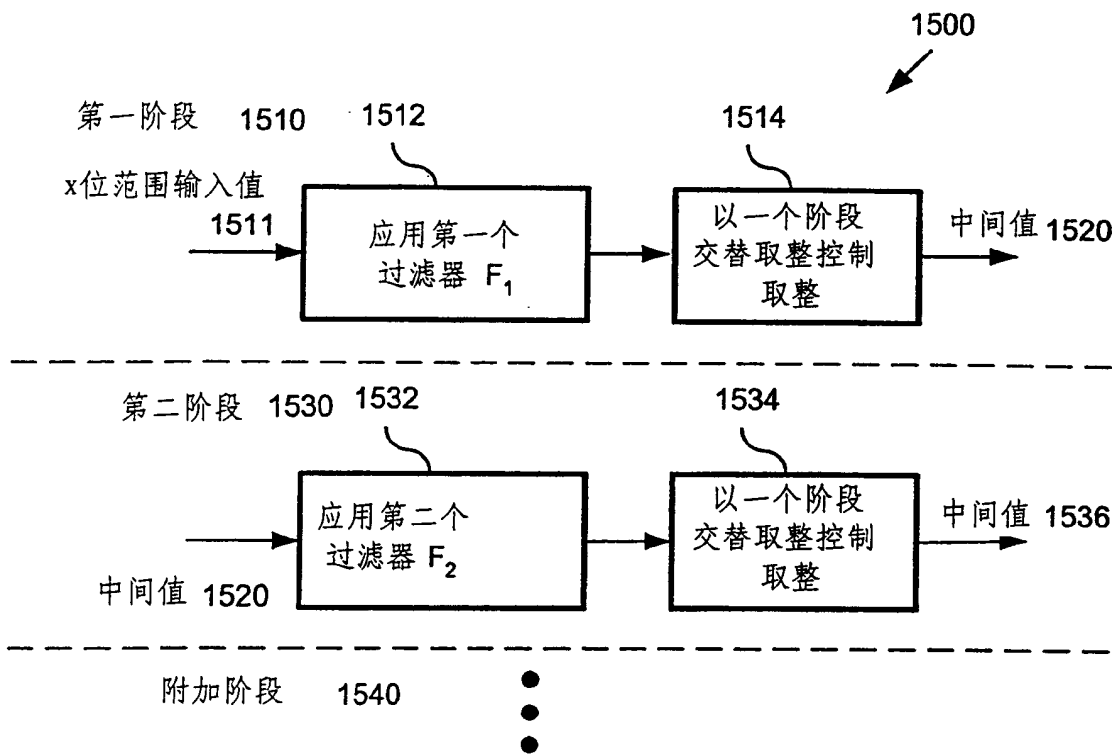


图 15

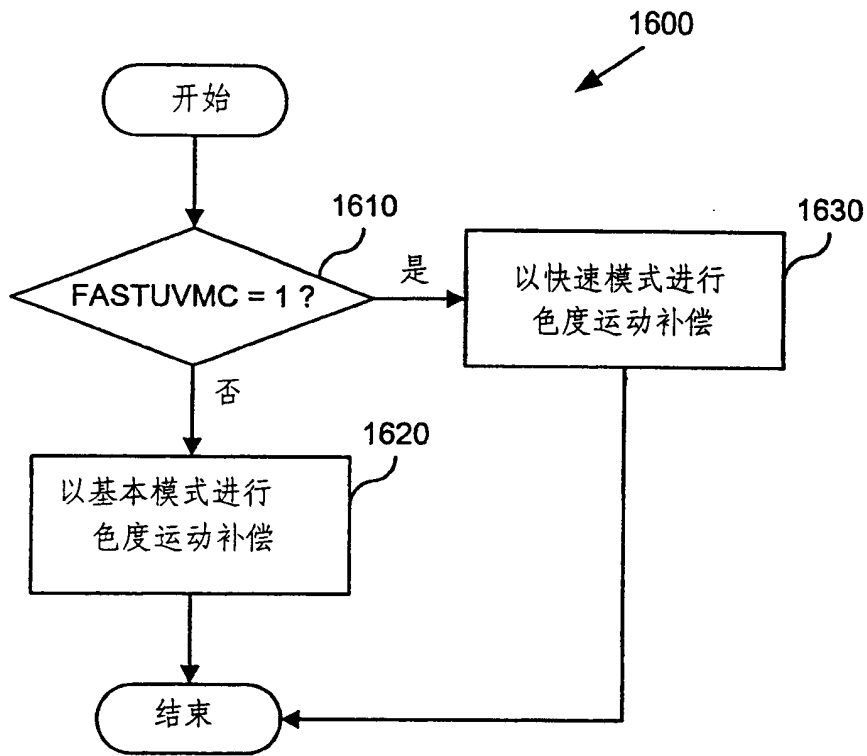


图 16

1700

亮度	-1	$-\frac{3}{4}$	$-\frac{1}{2}$	$-\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1
色度	$-\frac{1}{2}$	$-\frac{1}{2}$	0	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$

1710

1720

图 17

1800

亮度	-1	$-\frac{3}{4}$	$-\frac{1}{2}$	$-\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{4}$	1
色度	$-\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{4}$	0	0	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$

1810

1820

图 18