



[12] 发明专利申请公开说明书

[21] 申请号 200410077192.2

[43] 公开日 2005年6月22日

[11] 公开号 CN 1630374A

[22] 申请日 2004.9.7

[21] 申请号 200410077192.2

[30] 优先权

[32] 2003. 9. 7 [33] US [31] 60/501,081

[32] 2004. 5. 27 [33] US [31] 10/857,473

[71] 申请人 微软公司

地址 美国华盛顿州

[72] 发明人 T·W·豪尔克姆

[74] 专利代理机构 上海专利商标事务所有限公司

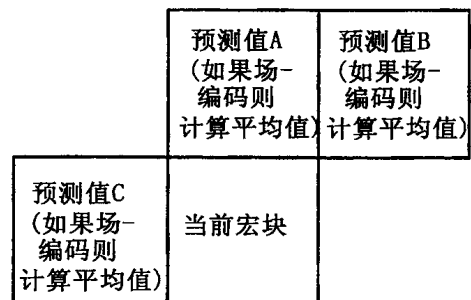
代理人 李家麟

权利要求书5页 说明书38页 附图37页

[54] 发明名称 用于前向预测的隔行扫描视频帧的场的预测运动矢量

[57] 摘要

描述用于编码和解码隔行扫描的视频中预测图像的技术和工具。例如，视频编码器或解码器为隔行扫描的P场的一部分(例如，一个块或宏块)的运动矢量计算运动矢量预测值，包括在为该部分使用相同极性或使用相反极性的运动矢量预测值之间选择。编码器/解码器至少部分地基于为该运动矢量计算的预测值来处理运动矢量。该处理可以包括在编码过程中计算运动矢量与运动矢量预测值之间的差分，并在解码过程中由运动矢量差分与运动矢量预测值来重新构造运动矢量。该选择可以至少部分地基于在该部分周围的近邻的相反极性运动矢量的计数和/或相同极性运动矢量的计数。



1. 一种方法，其特征在于，所述方法包括：
为隔行扫描的 P 场的一部分的运动矢量计算运动矢量预测值，包括在为所述部分使用相同极性运动矢量预测值与使用相反极性运动矢量预测值之间选择；以及
至少部分地基于为运动矢量计算的运动矢量预测值来处理运动矢量。
2. 如权利要求 1 所述的方法，其特征在于，所述处理的步骤包括在编码过程中计算运动矢量与运动矢量预测值之间的运动矢量差分。
3. 如权利要求 1 所述的方法，其特征在于，所述处理的步骤包括在解码过程中由运动矢量差分和运动矢量预测值来重新构造运动矢量。
4. 如权利要求 1 所述的方法，其特征在于，所述部分是一宏块。
5. 如权利要求 1 所述的方法，其特征在于，所述部分是一宏块的一块。
6. 如权利要求 1 所述的方法，其特征在于，所述选择的步骤至少部分地基于在所述部分周围的一个近邻的相反极性运动矢量的计数和/或基于所述近邻的相同极性运动矢量的计数。
7. 如权利要求 1 所述的方法，其特征在于，所述选择的步骤至少部分地基于比特流中的单个比特信号。
8. 如权利要求 1 所述的方法，其特征在于，所述选择的步骤包括：
至少部分地基于在所述部分周围的一个近邻的相反极性运动矢量的计数和/或所述近邻的相同极性运动矢量的计数，确定主导预测值；以及
基于主导预测值与比特流中单个比特信号来选择相同极性运动矢量预测值或相反极性运动矢量预测值。
9. 如权利要求 1 所述的方法，其特征在于，所述选择的步骤包括：
如果用于所述部分的运动矢量参考相同极性的场，则选择相同极性的运动矢量预测值，或者如果用于所述部分的运动矢量参考相反极性的场，则选择相反极性的运动矢量预测值。
10. 如权利要求 1 所述的方法，其特征在于，还包括可以作为在所述部分周围的一个近邻的多个相同极性的运动矢量预测值候选的中值来计算相同极性的运动矢量预测值。
11. 如权利要求 10 所述的方法，其特征在于，通过定标一个相反极性的运

动矢量预测值候选来导出多个相同极性的运动矢量预测值候选的至少一个。

12. 如权利要求 1 所述的方法，其特征在于，还包括作为在所述部分周围的一个近邻的多个相反极性的运动矢量预测值的候选的中值来计算相反极性的运动矢量预测值。

13. 如权利要求 12 所述的方法，其特征在于，通过定标一个相同极性的运动矢量预测值候选来导出多个相反极性的运动矢量预测值候选的至少一个。

14. 如权利要求 1 所述的方法，其特征在于，还包括为隔行扫描的 P 场的多个其它部分的每一个重复所述计算和处理步骤。

15. 一种方法，其特征在于，包括：

至少部分地基于多个相邻的运动矢量来为隔行扫描的 P 场的一部分的运动矢量计算运动矢量预测值，其中，所述多个相邻的运动矢量包括一个或多个相反极性运动矢量和一个或多个相同极性运动矢量；以及

至少部分地基于为运动矢量计算的运动矢量预测值来处理运动矢量。

16. 如权利要求 15 所述的方法，其特征在于，所述处理的步骤包括在编码过程中计算运动矢量与运动矢量预测值之间的运动矢量差分。

17. 如权利要求 15 所述的方法，其特征在于，所述处理的步骤包括在解码过程中由运动矢量差分和运动矢量预测值来重新构造运动矢量。

18. 如权利要求 15 所述的方法，其特征在于，所述部分是一宏块。

19. 如权利要求 15 所述的方法，其特征在于，所述部分是一宏块的一块。

20. 如权利要求 15 所述的方法，其特征在于，还包括作为多个相同极性的运动矢量预测值候选的中值来计算相同极性的运动矢量预测值，其中，至少基于定标来自一个或多个相反极性的运动矢量之一的值来导出多个候选的至少一个。

21. 如权利要求 15 所述的方法，其特征在于，还包括作为多个相反极性的运动矢量预测值候选的中值来计算相反极性的运动矢量预测值，其中，至少基于定标来自一个或多个相同极性的运动矢量之一的值来导出多个候选的至少一个。

22. 一种处理隔行扫描的预测视频帧的方法，其特征在于，所述方法包括：
为视频帧中当前场中的宏块确定运动矢量预测值候选的第一集合，运动矢量预测值候选的第一集合引用相对于当前场具有相同极性的第一参考场；

至少部分地基于运动矢量预测值候选的第一集合中的一个或多个，为宏块

计算第一运动矢量预测值；

为宏块确定运动矢量预测值候选的第二集合，运动矢量预测值候选的第二集合引用具有相对于当前场的相反极性的第二参考场；

至少部分地基于运动矢量预测值候选的第二集合中的一个或多个，为宏块计算第二运动矢量预测值；以及

在所述处理步骤中使用第一和第二运动矢量预测值。

23. 如权利要求 22 所述的方法，其特征在于，所述使用第一和第二运动矢量预测值的步骤包括选择第一和第二运动矢量预测值之一并基于选择的运动矢量预测值来为宏块计算运动矢量差分，所述方法还包括熵编码运动矢量差分。

24. 如权利要求 22 所述的方法，其特征在于，还包括为宏块熵解码运动矢量差分，其中，所述使用第一和第二运动矢量预测值的步骤包括选择第一和第二运动矢量预测值之一并将选择的预测值与运动矢量差分结合起来为宏块重新构造运动矢量。

25. 一种具有在其上存储用于在视频解码器中执行如权利要求 22 所述的方法的计算机可执行指令的计算机可读介质。

26. 一种具有在其上存储用于在视频解码器中执行如权利要求 22 所述的方法的计算机可执行指令的计算机可读介质。

27. 一种方法，其特征在于，包括：

通过定标具有不同于第一极性的第二极性的实际运动矢量值，来计算第一极性运动矢量预测值候选；以及

当为隔行扫描的 P 场的一部分的运动矢量计算运动矢量预测值时，使用第一极性运动矢量预测值候选。

28. 如权利要求 27 所述的方法，其特征在于，还包括在编码过程中计算运动矢量与运动矢量预测值之间的运动矢量差分。

29. 如权利要求 27 所述的方法，其特征在于，还包括在解码期间由运动矢量差分与运动矢量预测值来重新构造运动矢量。

30. 如权利要求 27 所述的方法，其特征在于，所述部分是一宏块。

31. 如权利要求 27 所述的方法，其特征在于，所述部分是一宏块的一块。

32. 如权利要求 27 所述的方法，其特征在于，第一极性运动矢量预测值候选具有与所述部分相反的场极性，其中，实际运动矢量值具有与所述部分相同

的场极性，以及其中，调整所述定标以从相同极性定标成相反极性。

33. 如权利要求 27 所述的方法，其特征在于，第一极性运动矢量预测值候选具有与所述部分相同的场极性，其中，实际运动矢量值具有与所述部分相反的场极性，以及其中，调整所述定标以从相反极性定标成相同极性。

34. 如权利要求 27 所述的方法，其特征在于，所述定标是至少部分地基于参考帧距离。

35. 一种处理隔行扫描的预测视频帧的方法，其特征在于，所述方法包括：
为视频帧中当前场中的宏块确定第一运动矢量预测值候选，第一运动矢量预测值候选具有第一极性；

为宏块确定第二运动矢量预测值候选，其中，第二运动矢量预测值候选是从第一运动矢量预测值候选使用定标操作来导出的；以及

其中，第二运动矢量预测值候选具有不同于第一极性的极性。

36. 一种具有在其上存储用于在视频解码器中执行如权利要求 35 所述的方法的计算机可执行指令的计算机可读介质。

37. 一种具有在其上存储用于在视频解码器中执行如权利要求 35 所述的方法的计算机可执行指令的计算机可读介质。

38. 一种处理隔行扫描的预测视频帧的方法，其特征在于，所述方法包括：
为视频帧中当前场中的宏块确定第一运动矢量预测值候选，第一运动矢量预测值候选引用具有第一极性和参考距离的场；以及

为宏块确定第二运动矢量预测值候选，其中，第二运动矢量预测值候选是从第一运动矢量预测值候选使用至少部分地基于参考距离变化的定标操作来导出的。

39. 如权利要求 38 所述的方法，其特征在于，第二运动矢量预测值候选具有不同于第一极性的极性。

40. 一种具有在其上存储用于在视频解码器中执行如权利要求 38 所述的方法的计算机可执行指令的计算机可读介质。

41. 一种具有在其上存储用于在视频解码器中执行如权利要求 38 所述的方法的计算机可执行指令的计算机可读介质。

42. 一种为编码的视频数据比特流中的当前帧解码数据的方法，其特征在于，所述方法包括：

解码比特流中一个场级处的第一元素，第一元素包括用于视频帧中的当前场的参考场数量信息，参考场数量信息表示用于当前场的两个参考场；

解码比特流中一个宏块级处的第二元素，第二元素包括用于当前场中的当前宏块的宏块模式信息；以及

解码宏块级处的第三元素，第三元素包括用于当前宏块的运动矢量数据，其中，运动矢量数据包括不同的运动矢量数据；以及

为宏块处理运动矢量数据，所述处理的步骤包括：

为当前宏块确定第一和第二运动矢量预测值，其中，第一运动矢量预测值是奇数场预测值而第二运动矢量预测值是偶数场预测值；以及

至少部分地基于第一和第二运动矢量预测值并至少部分地基于差分运动矢量数据，为当前宏块重新构造运动矢量。

43. 一种解码隔行扫描的 P 场的当前宏块的方法，其特征在于，所述方法包括：

解码运动矢量数据，运动矢量数据共同地用信号表示水平差分运动矢量分量、垂直差分运动矢量分量和表示是否使用主导的或非主导的运动矢量预测值的标志；以及

至少部分地基于解码的运动矢量数据，重新构造运动矢量。

44. 如权利要求 43 所述的方法，其特征在于，运动矢量是用于当前宏块的。

45. 如权利要求 43 所述的方法，其特征在于，运动矢量是用于当前宏块的一个块的。

46. 如权利要求 43 所述的方法，其特征在于，所述重新构造的步骤包括：
基于标志在使用主导运动矢量预测值或非主导运动矢量预测值之间选择；
将水平差分运动矢量分量与选择的预测值的水平分量结合起来；以及
将垂直差分运动矢量分量与选择的预测值的水平分量结合起来。

用于前向预测的隔行扫描视频帧的场的预测运动矢量

相关申请信息

本发明要求 2003 年九月 7 日提交的美国临时专利申请号 60/501,081, 标题为“Video Encoding and Decoding Tools and Techniques(视频编码和解码工具与技术)”的利益, 通过引用将它包括在此。

技术领域

描述用于隔行扫描视频编码和解码的技术与工具。例如, 视频编码器/解码器为隔行扫描视频帧的场计算运动矢量预测值。

背景技术

数字视频消耗大量的存储器和传输容量。典型的未加工的数字视频序列每秒包括 15 或 30 个图象。每个图象可以包括几万或几十万像素(pixel)(也称为 pels(象素))。每个像素代表图象中的一个微小元素。在未加工的形式中, 计算机通常用 24 比特或更多的比特来表示一个像素。因而, 典型的未加工的数字视频序列的每秒比特数, 或者比特率可以是 5 百万比特/秒或更多。

许多计算机和计算机网络缺乏资源来处理未加工的数字视频。因此, 工程师使用压缩(也称为编码(coding 或 encoding))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害, 但对比特率的减少受到视频复杂性的限制。或者, 压缩可以是有损的, 其中视频的质量受到损害, 但对比特率的减少更有效。解压缩与压缩相反。

一般而言, 视频压缩技术包括“帧内(intra)”压缩和“帧间(inter)”或预测压缩。帧内压缩技术压缩单独的图片, 一般称为 I 图片或关键图片。帧间压缩技术参考先前和/或后面的图片来压缩图片, 且一般称为预测图片、P 图片或 B 图片。

I. 在 Windows Media Video(视窗媒体视频)版本 8 和 9 中的帧间压缩

Microsoft(微软)公司的 Windows Media Video 版本 8[“WMV8”]包括视频编

码器和视频解码器。WMV8 编码器使用帧内和帧间压缩，WMV8 解码器使用帧内和帧间解压缩。Windows Media Video 版本 9[“WMV9”]的早期版本对于许多操作使用相似的结构。

WMV8 编码器中的帧间压缩使用基于分块的运动补偿预测编码，后跟随剩余误差(residual error)变换编码。图 1 和 2 例示在 WMV8 编码器中对于预测帧的基于分块的帧间压缩。特别地，图 1 例示预测帧 110 的运动估计，而图 2 例示预测帧的运动补偿块的预测剩余部分(prediction residual)的压缩。

例如，在图 1 中，WMV8 编码器为预测帧 110 中的宏块(macroblock)115 计算运动矢量。为了计算运动矢量，编码器搜索参考帧 130 的搜索区 135。在搜索区 135 内，编码器将来自预测帧 110 的宏块 115 与各种候选宏块比较，以便找出一个良好匹配的候选宏块。编码器输出指定用于匹配宏块的(熵编码的)运动矢量的信息。

由于运动矢量值常常与在空间上环绕运动矢量的值有关系，用于传输运动矢量信息的数据压缩可以通过选择来自相邻宏块的运动矢量预测值并使用该预测值为当前宏块预测运动矢量来完成。编码器可以编码运动矢量与预测值之间的差分。在通过将差分加到预测值上来重新构造运动矢量之后，解码器使用运动矢量，用来自在参考帧 130 的信息来为宏块 115 计算预测宏块，其中参考帧 130 是先前重新构造的、在编码器和解码器可得到的帧。预测很少是完美的，因此编码器通常编码预测宏块与宏块 115 本身之间像素差的块(也称为误差或剩余块)。

图 2 例示在 WMV8 编码器中误差块 235 的计算与编码的例子。误差块 235 是预测块 215 与原来的当前块 225 之间的差。编码器将 DCT 240 应用于误差块 235，得到一个 8x8 的系数块 245。编码器随后量化 250DCT 系数，得到 8x8 最化的 DCT 系数块 255。编码器将 8x8 块 255 扫描 260 到一维阵列 265 中，使得系数通常从最低频率到最高频率排序。编码器使用游程长度编码(run length encoding)270 的变体来熵编码经扫描的系数。编码器从一个或多个行程/级/最后(run/level/last)表 275 选择一个熵代码并输出这个熵代码。

图 3 示出用于帧间编码块的相应解码过程 300。在图 3 中概括地说，使用具有一个或多个游程/级/最后表 315 的可变长度解码(variable length decoding)310 和游程长度解码(run length decoding)320，解码器解码(310, 320)表示预测剩余部分的经熵编码的信息。解码器将存储经熵解码的信息的一

维阵列 325 逆扫描 330 到二维块 335 中。解码器逆量化并逆离散余弦变换(一起, 340)数据, 得到重新构造的误差块 345。在一个独立的运动补偿路径中, 解码器使用用于离参考帧的位移的运动矢量信息 355 来计算预测的块 365。解码器将预测块 365 与重新构造的误差块 345 结合起来, 来形成重新构造的块 375。

原来的与重新构造的帧之间的改变量就是失真, 而编码帧所需要的位数表示帧的速率。失真量约略与速率成反比。

II. 隔行扫描的视频和逐行扫描的视频

典型的隔行扫描视频帧由不同时间开始的两个场组成。例如, 参考图 4, 隔行扫描的视频帧 400 包括顶场 410 和底场 420。通常, 在一个时间(时间 t) 开始扫描偶数行(顶场)并在一个不同(一般稍后)时间(例如时间 $t+1$) 开始扫描奇数行(底场)。这个时序可以在存在运动的隔行扫描的视频帧的区域中产生锯齿化牙状特点, 因为在不同时间开始扫描两个场。因此, 可以按照场结构重新安排隔行扫描的视频帧, 将奇数行一起分组在一个场中, 而将偶数行一起分组在另一个场中。这种安排, 称为场编码, 在高度运动的图象中对于减少这样的锯齿化边缘人为假象是有用的。另一方面, 在静止区域中, 可在没有这样的重新安排的情况下, 更有效地保持隔行扫描视频帧中的图象细节。因此, 在静止或低度运动的隔行扫描视频帧中常常使用帧编码, 其中保持原来交替的场行安排。

典型的逐行扫描视频帧由一个具有非交替行的内容的帧组成。与隔行扫描的视频相比, 逐行扫描的视频不将视频帧划分成独立的场, 而是在单一时间开始从左至右、由顶至底扫描整个帧。

III. 在 WMV9 的早期版本中隔行扫描 P 帧编码与解码

Windows Media Video 版本 9[“WMV9”]的早期版本, 使用隔行扫描 P 帧编码和解码。在 WMV9 的这些早期版本中, 隔行扫描的 P 帧可以包含以场模式或者以帧模式编码的宏块, 或者被跳过的宏块, 其中通常在一个接一个宏块基础上作出判决。两个运动矢量与每一个场编码的宏块相关联, 而一个运动矢量与每一个帧编码的宏块相关联。编码器联合地编码宏块中块的运动信息, 包括水平和垂直运动矢量差分分量, 有可能连同其它信令(signaling)信息。

在编码器中，通过计算运动矢量与运动矢量预测值之间的差分来编码运动矢量，其中运动矢量预测值是根据相邻的运动矢量来计算的。而且，在解码器中，通过将运动矢量差分加到运动矢量预测值上来重新构造运动矢量，其中运动矢量预测值再次是根据相邻的运动矢量计算的(这次是在解码器中)。

图 5、6 和 7 示出在 WMV9 的早期版本中的隔行扫描的 P 帧中，分别用于帧编码的宏块和场编码的宏块的运动矢量预测值的候选预测值的例子。图 5 示出用于隔行扫描的 P 帧中的内部位置中当前的帧编码宏块的候选预测值 A、B 和 C(不是宏块行中第一或最后一个宏块，不是在顶行中)。可以从不同于那些标注 A、B 和 C 的候选方向获得预测(例如在诸如在当前宏块是一行或顶行中的第一个宏块或最后一个宏块时这样的特殊情况下，因为对于这样的情况某些预测值是不能得到的)。对于当前帧编码的宏块，取决于相邻的宏块是场编码的还是帧编码的来不同地计算预测值候选。对于相邻的帧编码宏块，简单地取运动矢量作为预测值候选。对于相邻的场编码宏块，通过对顶与底场运动矢量取平均来确定候选运动矢量。

图 6 和 7 示出用于在宏块行中但不在顶行中、不是第一或最后一个宏块的场编码宏块中的当前场的候选预测值 A、B 和 C。在图 6 中，当前场是底场，并使用相邻宏块中的底场运动矢量作为候选预测值。在图 7 中，当前场是顶场，并使用顶场运动矢量作为候选预测值。因而，对于当前的场编码宏块中的每一个场，用于每一个场的运动矢量预测值候选的数量最多是三，其中每一个来自同一场类型(例如顶或底)的每一个候选作为当前场。

用于当前宏块或当前宏块的场的预测值是根据候选预测值来选择的，并根据预测值来计算运动矢量差分。可以通过在编码器或解码器任意一边将运动矢量差分加到选择的运动矢量预测值上，来重新构造运动矢量。通常，由经编码的运动信息来重新构造亮度运动矢量，而由重新构造的亮度运动矢量来导出色度运动矢量。

IV. 用于视频压缩和解压缩的标准

除 WMV8 和 WMV9 的早期版本之外，若干国际标准涉及视频压缩和解压缩。这些标准包括运动图片专家组[“MPEG”]1、2 和 4 标准和来自国际电信联合会[“ITU”]的 H.261、H.262、H.263 和 H.264 标准。用于在国际标准中实现数字视频序列的数据压缩的一种基本方法是要减少图片之间的时间冗余(temporal

redundancy)。这些流行的压缩方案(MPEG-1、MPEG-2、MPEG-4、H.261、H.263等)使用运动估计和补偿。例如,将当前帧被划分成统一的正方形区域(例如块和/或宏块)。为每一当前区域通过发送用于该区域的运动矢量信息来指定匹配的区域。运动矢量表示在先前编码(和重新构造)的帧中要用作当前区域的预测值的区域的位置。导出当前区域与参考帧中的区域之间一个接一个像素的差,称为误差信号。这个误差信号通常具有比原来的信号低的熵。因此,可以用较低的速率来编码该信息。如在WMV8和WMV9的早期版本中,由于运动矢量值常常与在空间环绕的运动矢量有关系,用于表示运动矢量信息的数据的压缩可以通过编码当前运动矢量与基于先前编码的相邻运动矢量的预测值之间的差分来完成。

此外,有些国际标准描述在隔行扫描视频帧中的运动估计和补偿。H.262标准允许隔行扫描的视频帧编码为单一帧或为两个场,其中可以自适应地在一帧接一帧的基础上选择帧编码或场编码。H.262标准描述基于场的预测,它是只使用参考帧的一个场的预测模式。H.262标准还描述双主预测(dual-prime prediction),这是一种预测模式,其中为隔行扫描的P图象中一个16x16块平均两个前向的基于场的预测。H.262标准的7.6节描述“场预测”,包括在两个参考场之间选择用于隔行扫描视频帧的当前场的宏块的运动补偿。7.6.3节描述运动矢量预测和重新构造,其中用于给定宏块的重新构造的运动矢量变成用于随后被编码/被解码的宏块的运动矢量预测值。这样的运动矢量预测在许多情况下未能恰当地为隔行扫描视频帧的场的宏块预测运动矢量。

给出对数字视频的视频压缩和解压缩的急迫重要性,不会惊讶视频压缩和解压缩是高度发展的领域。无论前面的视频压缩和解压缩技术有什么好处,但是,它们不具有下面的技术和工具的优点。

发明内容

总而言之,详细的描述是针对用于编码和解码隔行扫描的视频中预测视频图象的各种技术和工具的。所述的技术包括为隔行扫描视频帧的块或宏块计算运动矢量预测值的技术。这些技术在许多情况下提高运动矢量预测值的准确性,从而减少了与编码用于隔行扫描视频帧的场的运动矢量信息相关联的比特率。可以组合地或独立地使用各种技术和工具。

在第一方面,编码器/解码器为隔行扫描的P场的一部分(例如块或宏块)

的运动矢量计算运动矢量预测值，包括在为该部分使用相同极性运动矢量预测值与使用相反极性运动矢量预测值之间选择。编码器/解码器至少部分地基于为运动矢量计算的运动矢量预测值来处理运动矢量。该处理可以包括在编码过程中计算运动矢量与运动矢量预测值之间的运动矢量差分，和在解码过程中由运动矢量差分与运动矢量预测值重新构造运动矢量。该选择可以至少部分地基于在该部分周围的近邻的相反极性运动矢量的计数和/或近邻的相同极性运动矢量的计数。该选择还可以至少部分地基于比特流中单个比特信号。该选择可以包括至少部分地基于在该部分周围的近邻的相反极性运动矢量的计数和/或近邻的相同极性运动矢量的计数来确定主导预测值，并基于主要预测值和比特流中的单个比特信号来选择相同极性运动矢量预测值或者相反极性运动矢量预测值。

该选择还可以包括，如果用于该部分的运动矢量参考相同极性场，则选择相同极性运动矢量预测值，或者如果用于该部分的运动矢量参考相反极性场，则选择相反极性运动矢量预测值。

可以作为在该部分周围的近邻的多个相同极性的运动矢量预测值候选的中值来计算相同极性运动矢量预测值，并可以作为在该部分周围的近邻的多个相反极性的运动矢量预测值的中值来计算相反极性运动矢量预测值。相同极性运动矢量预测值候选可以通过定标相反极性运动矢量预测值候选来导出，并且相反极性运动矢量预测值可以通过定标相同极性运动矢量预测值候选来导出。

在另一方面，编码器/解码器至少部分地基于多个相邻的运动矢量为隔行扫描的P场的一部分的运动矢量计算运动矢量预测值，其中，多个相邻的运动矢量包括一个或多个相反极性运动矢量和一个或多个相同极性运动矢量。编码器/解码器至少部分地基于为运动矢量计算的运动矢量预测值来处理运动矢量。可以作为多个相同极性运动矢量预测值候选的中值来计算相同极性运动矢量预测值，其中，至少基于定标来自一个或多个相反极性运动矢量之一的值来导出多个候选的至少一个，以及可以作为多个相反极性运动矢量预测值候选的中值来计算相反极性运动矢量预测值，其中，至少基于定标来自一个或多个相同极性运动矢量之一的值来导出多个候选的至少一个。

在另一方面，编码器/解码器处理隔行扫描的预测视频帧，这是通过为视频帧中当前场中的宏块确定运动矢量预测值候选的第一集合，运动矢量预测值候选的第一集合引用具有相对于当前场的相同极性的第一参考场。至少部分地

基于运动矢量预测值候选的第一集合中的一个或多个，为宏块计算第一运动矢量预测值。为宏块确定运动矢量预测值候选的第二集合，运动矢量预测值候选的第二集合引用具有相对于当前场的相反极性的第二参考场。至少部分地基于运动矢量预测值候选的第二集合的一个或多个，为宏块计算第二运动矢量预测值。

在该处理中使用第一和第二运动矢量预测值，诸如通过选择第一和第二运动矢量预测值之一并基于选择的运动矢量预测值为宏块计算运动矢量差分，和/或选择第一和第二运动矢量预测值之一并将选择的预测值与运动矢量差分结合起来，为宏块重新构造运动矢量。可以熵编码运动矢量差分。

在另一方面，通过定标具有不同于第一极性的第二极性的实际运动矢量值来计算第一极性运动矢量预测值候选。当为隔行扫描的P场的一部分的运动矢量计算运动矢量预测值时，使用第一极性运动矢量预测值候选。在编码过程中计算运动矢量与运动矢量预测值之间的运动矢量差分。在解码过程中由运动矢量差分与运动矢量预测值来重新构造运动矢量。第一极性运动矢量预测值候选具有与该部分相反的场极性，其中，实际运动矢量值具有与该部分相同的场极性，以及其中，调整定标以从相同极性定标成相反极性。或者，第一极性运动矢量预测值候选可以具有与该部分相同的场极性，其中，实际运动矢量值具有与该部分相反的场极性，以及其中，调整定标以从相反极性定标成相同极性。定标至少部分地基于参考帧距离。

在另一方面，编码器/解码器为视频帧中当前场中的宏块确定第一运动矢量预测值候选，第一运动矢量预测值候选具有第一极性。编码器/解码器为宏块确定第二运动矢量预测值候选，其中，第二运动矢量预测值候选是由第一运动矢量预测值候选使用定标操作来导出的。第二运动矢量预测值候选具有不同于第一极性的极性。

在另一方面，编码器/解码器为视频帧中当前场中的宏块确定第一运动矢量预测值候选，第一运动矢量预测值候选引用具有第一极性和参考距离的场。编码器/解码器为宏块确定第二运动矢量预测值候选，其中，第二运动矢量预测值候选是由第一运动矢量预测值候选使用至少部分地基于参考距离变化的定标操作来导出的。第二运动矢量预测值候选具有不同于第一极性的极性。

在另一方面，解码器解码比特流中的一个场层次处的第一元素，和比特流中一个宏块层次处的第二元素与第三元素。第一元素包括用于视频帧中当前场

的参考场数量信息，它表示用于当前场的两个参考场。第二元素包括用于当前场中的当前宏块的宏块模式信息。第三元素包括用于当前宏块的运动矢量数据，其中，运动矢量数据包括差分运动矢量数据。解码器为宏块处理运动矢量数据。该处理包括为当前宏块确定第一和第二运动矢量预测值，以及至少部分地基于第一和第二运动矢量预测值和至少部分地基于差分运动矢量数据，为当前宏块重新构造运动矢量。第一运动矢量预测值是奇数场预测值而第二运动矢量预测值是偶数场预测值。

在另一方面，解码器解码运动矢量数据，运动矢量数据共同地用信号表示水平差分运动矢量分量、垂直差分运动矢量分量和表示是否使用主导的或非主导的运动矢量预测值的标志。解码器至少部分地基于经解码的运动矢量数据，重新构造运动矢量。该重新构造可以包括基于该标志在使用主导运动矢量预测值或非主导运动矢量预测值之间选择，将水平差分运动矢量分量与选择的预测值的水平分量结合起来，以及将垂直差分运动矢量分量与选择的预测值的水平分量结合起来。

通过下面参考附图进行的不同实施例的详细描述，将使另外的特征和优点显而易见。

附图说明

图 1 示出在按照背景技术的视频编码器中的运动估计。

图 2 示出在按照背景技术的视频编码器中对 8x8 的预测剩余部分块进行基于块的压缩。

图 3 示出在按照背景技术的视频编码器中对 8x8 的预测剩余部分块进行基于块的解压缩。

图 4 示出按照背景技术的隔行扫描视频帧。

图 5 示出 WMV9 的早期版本中用于当前的帧编码宏块的候选运动矢量预测值。

图 6 和 7 示出 WMV9 的早期版本中用于当前的场编码宏块的候选运动矢量预测值。

图 8 是合适的计算环境的方框图，可结合该环境实现几个所述的实施例。

图 9 是一般化的视频编码器系统的方框图，可结合该系统实现几个所述实施例。

图 10 是一般化的视频编码器系统的方框图，可结合该系统实现几个所述实施例。

图 11 是在几个所述的实施例中使用的宏块格式的图。

图 12A 是隔行扫描视频帧的部分的图，示出顶场和底场的交替行。图 12B 是为编码/解码而组织为帧的隔行扫描视频帧的图，而图 12C 是为编码/解码而组织为场的隔行扫描视频帧。

图 13 和 14 是代码列表，示出分别用于 3 个值的中值和 4 个值的中值计算的伪代码。

图 15 和 16 示出隔行扫描的 P 场，每个场具有两个参考场。

图 17 示出对于当前与参考场极性的不同组合，运动矢量的垂直分量与相应空间位置之间的关系。

图 18 是流程图，示出用于为具有两个可能的参考场的隔行扫描的 P 场选择运动矢量预测值。

图 19 示出用于当前宏块的由三个候选运动矢量预测值组成的两个集合。

图 20A-20F 是代码列表，示出用于计算二个参考隔行扫描的 P 场中的运动矢量预测值的伪代码。

图 21A-21B 是代码列表，示出用于定标来自一个场的预测值来导出来自另一个场的预测值。

图 22 和 23 是示出与不同参考帧距离相关联的定标操作值的表。

图 24 示出在组合的实现中用于隔行扫描的 P 场的帧层比特流语法。

图 25 示出在组合的实现中用于隔行扫描的 P 场的场层比特流语法。

图 26 示出在组合的实现中用于隔行扫描的 P 场的宏块层比特流语法。

图 27A 和 27B 是代码列表，示出例示为一个参考隔行扫描的 P 场解码运动矢量差分的伪代码。

图 28A 和 28B 是代码列表，示出例示为两个参考隔行扫描的 P 场解码运动矢量差分 and 主导/非主导预测值信息的伪代码。

图 29A 和 29B 示出用于隔行扫描的 P 场中的 1MV 宏块的候选运动矢量预测值的宏块的位置。

图 30A 和 30B 示出用于在混合的 1MV/4MV 隔行扫描的 P 场中的 1MV 宏块的候选运动矢量预测值的块的位置。

图 31A、31B、32A、32B、33 和 34 示出用于在混合的 1MV/4MV 隔行扫描的

P 场中的 4MV 宏块中不同位置的块的候选运动矢量预测值的块的位置。

图 35A—35F 是代码列表，示出在组合的实现中用于计算两个参考隔行扫描的 P 场中的运动矢量预测值的伪代码。

图 36 是代码列表，示出用于确定两个参考隔行扫描的 P 场中的参考场的伪代码。

具体实施方式

本申请涉及用于隔行扫描的视频的有效压缩和解压缩的技术和工具。在所述的各种实施例中，例如，视频编码器和解码器在编码和解码隔行扫描视频帧中的预测场时结合用于预测运动矢量的技术，以及与包括不同层(layer)或级(level)(例如，序列级，图象(picture)/图象(image)级，场级，宏块级和/或块级)的比特流格式或语法一起使用的信令技术。例如在数字视频广播系统(例如，电缆、卫星、DSL 等)中，可以使用这些技术和工具。

特别地，所述的技术和工具提高了用于隔行扫描视频帧的正向预测场的块和/或宏块的运动矢量预测值的质量，从而允许运动矢量被更有效地编码。例如，描述用于使用周围的块和/或宏块的相邻运动矢量为当前场的块和/或宏块产生预测值运动矢量的技术，其中相邻运动矢量共同地参考两个场中的一个或两个作为参考。在所述技术和工具中实现的创新包括但不限于下列内容：

1) 为隔行扫描的 P 场(interlaced P-field)中的块或宏块产生两个运动矢量预测值：一个运动矢量预测值用于“偶数”参考场而一个运动矢量预测值用于“奇数”参考场。编码器/解码器为当前块或宏块考虑达六个运动矢量预测值候选(三个用于偶数运动矢量预测值，三个用于奇数运动矢量预测值)。多达六个运动矢量预测值候选的使用允许进行比现有方法更好的运动矢量预测。

2) 使用来自当前运动矢量引用的参考场的运动矢量预测值：如果当前运动矢量引用参考帧的相应场中的区域(指当前场与参考场是相同极性的)，则使用来自那个参考场的运动矢量预测值。同样，如果使用相反极性场作为参考，则使用来自那个参考场的运动矢量预测值。这允许进行比现有方法更好的运动矢量预测值。

3) 使用定标操作为现有运动矢量候选的相反极性产生运动矢量预测值候选。编码器/解码器取用来自候选块/宏块位置的运动矢量，其中该运动矢量或者引用奇数场或者引用偶数场。编码器/解码器随后定标运动矢量来导出其它

极性的运动矢量预测值候选。例如，如果一个毗邻宏块的实际运动矢量引用奇数场，则使用那个值作为来自毗邻宏块的奇数运动矢量预测值候选，并使用经定标的运动矢量值(从实际值导出的)作为来自毗邻宏块的偶数运动矢量预测值候选。尽管运动矢量预测值候选只是从三个块/宏块位置(具有三个运动矢量值)获得的，但编码器/解码器的不同极性运动矢量预测值候选的推导给出总数六个候选以供从中选择。

4)在导出运动矢量预测值候选时使用取决于参考帧距离的定标操作。编码器/解码器使用当前场与两个参考场之间的相对时间距离来执行定标操作，定标操作从来自另一个场的现有运动矢量候选为缺少的极性导出预测值候选。考虑相对瞬时距离以执行定标的编码器/解码器，可以产生比假定不变的参考距离的编码器/解码器更准确的预测值。

对于在此所述的实现的各种可替换方案是可能的。例如，参考流程图描述的技术可以通过改变流程图所示阶段的顺序、通过重复或省略某些阶段等来改变。作为另一个例子，尽管某些实现是参考特定的宏块和/或块格式来描述的，但也可以使用其它格式。而且，参考隔行扫描的P场类型预测描述的技术和工具也可应用于其它类型的预测。

可以组合地或独立地使用各种技术和工具。不同的实施例实现所述技术和工具中的一个或多个。在视频编码器或解码器中，或者在不是明确地受限于视频编码或解码的某些其它系统中，可以使用在此描述的技术和工具。

I. 计算环境

图8例示合适的计算环境800的一般化例子，在该环境中可实现几个所述的实施例。计算环境800不是想要建议对使用范围或功能的任何限制，而是可在各种各样的通用或专用计算环境中实现这些技术和工具。

参考图8，计算环境800包括至少一个处理单元810和存储器820。在图8中，这个最基本的配置830包括在虚线中。处理单元810执行计算机可执行指令，并可以是真实的或虚拟的处理器。在多处理器系统中，多个处理单元执行计算机可执行指令以增加处理能力。存储器820可以是易失性存储器(例如，寄存器、高速缓存、RAM)，非易失性存储器(例如，ROM、EEPROM、闪存等)，或者两者的组合。存储器820存储实现具有隔行扫描的P场的二个参考场运动矢量预测值的视频编码器或解码器的软件880。

计算环境可具有另外的特征。例如，计算环境 800 包括存储器 840，一个或多个输入设备 850，一个或多个输出设备 860，以及一个或多个通信连接 870。互连机制(未示出)诸如总线、控制器或网络将计算环境 800 的组件互相连接。一般地，操作系统软件(未示出)为在计算环境 800 中执行的其它软件提供操作环境，并协调计算环境 800 的组件的活动。

存储器 840 可以是可移动的或不可移动的，并包括磁盘、磁带或盒、CD-ROM、DVD 或任何可用于存储信息并在计算环境 800 内可以访问的其它介质。存储器 840 存储用于实现视频编码器或解码器的软件 880 的指令。

输入设备 850 可以是接触式输入设备诸如键盘、鼠标、笔或轨迹球，语音输入设备，扫描设备或给计算环境 800 提供输入的其它设备。对于音频或视频编码，输入设备 850 可以是声卡，视频卡，TV 调谐卡，或以模拟或数字形式接收音频或视频输入的相似设备，或将音频或视频样本读入计算环境 800 的 CD-ROM 或 CD-RW。输出设备 860 可以是显示器，打印机，扬声器，CD-写入器，或从计算环境 800 提供输出的其它设备。

通信连接 870 允许在通信介质上与另外的计算实体通信。通信介质传送信息，诸如计算机可执行指令、音频或视频输入或输出，或在经调制的数据信号中的其它数据。经调制的数据信号是一种信号，具有以将信息编码在该信号的这样一种方式设置或改变的其一个或多个特性。作为例子，而非限制，通信介质包括以电子、光、红外线、RF、声音或其它载波实现的有线或无线技术。

可以在计算机可读介质的一般环境中描述本技术和工具。计算机可读介质是可以在计算环境内访问的任何可用介质。作为例子，而非限制性地，就计算环境 800 而言，计算机可读介质包括存储器 820、存储装置 840、通信介质和上述的任何组合。

可以在计算机可执行指令的一般环境中描述本技术和工具，诸如包括在计算环境中在目标真实的(target real)或虚拟(virtual)的处理器上执行的程序模块中的计算机可执行指令。通常，程序模块包括例程、程序、库、对象、类、组件、数据结构等，它们执行特定的任务或实现特定的抽象数据类型。在各种实施例中按需要，可在程序模块之间组合或分开程序模块的功能。用于程序模块的计算机可执行指令可在本地或分布式计算环境内执行。

为了表达，详细描述使用象“估计(estimate)”、“补偿(compensate)”、“预测(predict)”和“应用(apply)”这样的术语来描述计算环境中的计算机操作。

这些术语是对由计算机执行的操作的高级抽象，且不应该与由人执行的动作混淆。相应于这些术语的实际计算机操作取决于实现而变化。

II. 一般化的视频编码器和解码器

图 9 是一般化的视频编码器 900 的方框图，可结合它实现所述的实施例。图 10 是一般化的视频解码器 1000 的方框图，可结合它实现所述的实施例。

在编码器 900 与解码器 1000 内的模块之间所示的关系表示在编码器和解码器中的一般信息流；为了简单，未示出其它关系。特别地，图 9 和 10 通常不示出表示用于视频序列、图象、宏块、块等的编码器设置、模式、表等的边信息(side information)。这样的边信息是在输出比特流中发送的，一般在熵编码边信息之后。输出比特流的格式可以是 Windows Media Video 版本 9 格式或其它格式。

编码器 900 和解码器 1000 处理视频图象，视频图象可以是视频帧、视频场或帧和场的组合。在图象或宏块级的比特流语法和语义可取决于使用帧还是使用场。可以改变宏块组织和整个时序。编码器 900 和解码器 1000 是基于块的，并使用 4:2:0 宏块格式用于帧，其中每个宏块包括四个 8x8 亮度块(有时视为一个 16x16 宏块)和两个 8x8 色度块。对于场，可使用相同或不同的宏块组织和格式。可在不同阶段进一步子划分 8x8 块，例如在频率变换和熵编码阶段。在下面更详细地描述示例视频帧组织。可供替换地，编码器 900 和解码器 1000 是基于对象的，使用不同的宏块或块格式，或者对不同于 8x8 块和 16x16 宏块的大小或配置的像素执行操作。

取决于实现和所希望的压缩类型，可增加、省略编码器或解码器的模块，将它们划分成多个模块，与其它模块组合起来，和/或用相似的模块代替它们。在可替换的实施例中，具有不同模块和/或模块的其它配置的编码器或解码器执行一个或多个所述技术。

A. 视频帧组织

在某些实现中，编码器 900 和解码器 1000 处理按如下组织的视频帧。帧包含视频信号的空间信息的行。对于逐行扫描的视频，这些行包含从一个时刻开始并继续通过至帧的底部的连续行的样本。逐行扫描的视频帧被划分成宏块，诸如图 11 中所示的宏块 1100。宏块 1100 包括四个 8x8 亮度块(Y1 至 Y4)

和两个 8x8 色度块，后者与四个亮度块位置在一起，但在水平与垂直方向只有一半分辨率，按照常规的 4:2:0 的宏块格式。可在不同阶段进一步细分 8x8 块，例如，在频率变换和熵编码阶段。逐行 I 帧是帧内编码的逐行扫描视频帧。逐行扫描的 P 帧是使用前向预测编码的逐行扫描视频帧，而逐行扫描的 B 帧是使用双向预测编码的逐行扫描视频帧。逐行扫描的 P 和 B 帧可包括帧内编码的宏块和不同类型的预测宏块。

对于隔行扫描的视频，帧由两个场即顶场和底场组成。这些场之一比另一个晚开始一个场周期。图 12a 示出隔行扫描的视频帧 1200 的部分，在隔行扫描的视频帧 1200 的左上部分包括顶场和底场的交替行。

图 12b 示出为编码/解码而组织为帧 1230 的图 12a 的隔行扫描视频帧 1200。隔行扫描视频帧 1200 已被划分成宏块，诸如宏块 1231 和 1232，它们使用如在图 11 中所示的 4:2:0 格式。在亮度平面中，每个宏块 1231、1232 包括来自顶场的 8 行，与来自底场的 8 交替总数 16 行，且每一行 16 像素长。（没有示出宏块 1231、1232 内亮度块和色度块的实际组织和布置，且实际上对于不同的编码选择可以变化）。在给定的宏块内，可共同地或者独立地以任何各种各样的相位编码顶场信息和底场信息。隔行扫描的 I 帧是隔行扫描视频帧的两个帧内编码的场，其中宏块包括两个场的信息。隔行扫描的 P 帧是使用正向预测编码的隔行扫描视频帧的两个场，而隔行扫描的 B 帧是使用双向预测编码的隔行扫描视频帧的两个场，其中宏块包括两个场的信息。隔行扫描的 P 和 B 帧可包括帧内编码的宏块和不同类型的预测宏块。

图 12c 示出为编码/解码而组织为场 1260 的图 12a 的隔行扫描视频帧 1200。隔行扫描视频帧 1200 的两个场的每一个被划分成宏块。顶场被划分成宏块，诸如宏块 1261，而底场被划分成宏块，诸如宏块 1262。（再一次，宏块使用如在图 11 中所示的 4:2:0 格式，但未示出宏块内亮度块和色度块的组织 and 布置）。在亮度平面中，宏块 1261 包括来自顶场的 16 行，而宏块 1262 包括来自底场的 16 行，每一行 16 像素长。隔行扫描的 I 场是隔行扫描视频帧的单一、独立表示的场。隔行扫描的 P 帧是使用正向预测编码的隔行扫描视频帧的单一、独立表示的场，而隔行扫描的 B 场是使用双向预测编码的隔行扫描视频帧的单一、独立表示的场。隔行扫描的 P 和 B 场可包括帧内编码的宏块和不同类型的预测宏块。

术语图片 (picture) 指源、编码的或重新构造的图象 (image) 数据。对于逐

行扫描视频，图片是逐行扫描视频帧。对于隔行扫描的视频，图片可以指隔行扫描的视频帧，帧的顶场，或帧的底场，取决于上下文。

B. 视频编码器

图 9 是一般化的视频编码器系统 900 的方框图。编码器系统 900 接收包括当前图片 905 在内的视频图片序列，并产生压缩的视频信息 995 作为输出。视频编码器的特定实施例一般使用一般化编码器 900 的变体或补充版本。

编码器系统 900 压缩预测图片和关键图片(key picture)。为了表示，图 9 示出关键图片通过编码器系统 900 的路径和预测图片的路径。编码器系统 900 的许多组件用于压缩关键图片和预测图片两者。由那些组件执行的准确操作可以根据正在压缩的信息类型变化。

预测图片(也称为 p 图片，对于双向预测称为 b-图片，或帧间编码的图片)是按照来自一个或多个其它图片(一般称为参考图片或锚(anchor))的预测值(或差)来表示的。预测剩余部分(prediction residual)是所预测的与原来的图片之间的差。相反，在不参考其它图片的情况下压缩关键图片(也称为 I 图片，帧间编码的图片)。

如果当前图片 905 是正向预测的图片，则运动估计器(motion estimator)910 根据一个或多个参考图片来估计宏块或当前图片 905 的像素的其它集合的运动，参考图片是在图片存储器(picture store)920、922 中缓冲的重新构造的先前的图片。如果当前图片 905 是双向预测的图片(B 图片)，则运动估计器 910 根据两个或多个重新构造的参考图片来估计当前图片 905 中的运动。一般地，运动估计器根据至少一个在时间上在前的参考图片和至少一个在时间上在后的参考图片来估计 B 图片中的运动。因此，编码器系统 900 可以使用独立的存储器 920 和 922 用于后向和正向参考图片。关于双向预测图片的更多信息，见 2003 年七月 18 日提交的美国专利申请序列号 10/622,378，标题为“Advanced Bi-Directional Predictive Coding of Video Frames(视频帧的高级双向预测编码)”。

运动估计器 910 可以按像素、 $\frac{1}{2}$ 像素、 $\frac{1}{4}$ 像素或者其它增量来估计运动，并可以在一个接一个图片基础或其它基础上切换运动估计的分辨率。运动估计的分辨率可以水平地和垂直地相同或不同。运动估计器 910 作为边信息输出运动信息 915，诸如差分运动矢量信息。编码器 900 通过例如为运动矢量计算一

个或多个预测值，计算运动矢量与预测值的差分并熵编码差分来编码运动信息。为了重新构造运动矢量，运动补偿器 930 将预测值与差分运动矢量信息组合起来。下面描述用于计算运动矢量预测值、计算差分运动矢量和为隔行扫描的 P 场重新构造运动矢量的各种技术。

运动补偿器 930 将重新构造的运动矢量应用于重新构造的图片 925，以形成经运动补偿的当前图片 935。但是预测很少是完美的，而经运动补偿的当前图片 935 与原来的当前图片 905 之间的差是预测剩余部分 945。可供替换地，运动估计器和运动补偿器应用另外类型的运动估计/补偿。

频率变换器 960 将空间域视频信息转换成频域(即光谱)数据。对于基于块的视频图片，频率变换器 960 将离散余弦变换[“DCT”]，DCT 的变体，或者其它块变换应用于像素数据或预测剩余部分数据块，产生频率变换系数块。可供替换地，频率变换器 960 应用另一种常规频率变换诸如傅立叶变换或使用小波或子带分析。频率变换器 960 可应用 8x8、8x4、4x8、4x4 或其它尺寸的频率变换。

量化器 970 随后量化光谱数据系数块。量化器以在一个接一个图片基础或其它基础上变化的步长，对光谱数据应用均匀的标量量化。可供替换地，量化器对光谱数据系数应用另一种类型的量化，例如非均匀的、矢量的或非自适应的量化，或者在不使用频率变换的编码器系统中直接量化空间域数据。除自适应量化之外，编码器 900 可以使用帧丢弃(frame dropping)、自适应滤波或其它技术用于速率控制。

如果在预测图片中给定的宏块没有某些类型的信息(例如，没有用于宏块的运动信息和没有剩余部分信息)，则编码器 900 可将宏块编码为跳过的宏块(skipped macroblock)。如果有，则编码器在压缩的视频信息 995 的输出比特流中用信号表示跳过的宏块。

当需要重新构造的当前图片用于后续的运动估计/补偿时，逆量化器 976 对量化的光谱数据系数执行逆量化。逆频率变换器 966 随后执行频率变换器 960 的逆操作，产生(为预测图片)重新构造的预测剩余部分或重新构造的关键图片。如果当前图片 905 是关键图片，则取重新构造的关键图片作为重新构造的当前图片(未示出)。如果当前图片 905 是预测图片，则将重新构造的预测剩余部分加到经运动补偿的当前图片 935 上，以形成重新构造的当前图片。图片存储器 920、922 之一缓冲在预测下一个图片时使用的重新构造的当前图片。在

某些实施例中，编码器将解块(de-blocking)的滤波器应用于重新构造的图片以自适应地平滑图片中的不连续。

熵编码器 980 压缩量化器 970 的输出和某个边信息(例如，对于运动信息 915，量化步长)。典型的熵编码技术包括算术编码(arithmetic coding)，差分编码(differential coding)，霍夫曼编码(Huffman coding)，行程长度编码(run length coding)，LZ 编码(LZ coding)，词典编码(dictionary coding)，和上述的组合。熵编码器 980 一般使用不同编码技术用于不同种类的信息(例如，DC 系数，AC 系数，不同种类的边信息)，并可以从特定编码技术内多个编码表中选择。

熵编码器 980 提供压缩视频信息 995 给多路复用器[“MUX”]990。MUX 990 可包括缓冲器，并可将缓冲器级指示器(buffer level indicator)反馈回比特率自适应模块用于速率控制。在 MUX 990 之前或之后，可以为在网络上传输，信道编码压缩的视频信息 995。信道编码(channel coding)可以将差错检测和校正数据应用于压缩的视频信息 995。

C. 视频解码器

图 10 是一般的视频解码器系统 1000 的方框图。解码器系统 1000 接收用于视频预测值的压缩序列的信息 1095，并产生包括重新构造的图片 1005 的输出。视频解码器的特定实施例一般使用通用的解码器 1000 的变体或补充版本。

解码器系统 1000 解压缩预测图片和关键图片。为了表示，图 10 示出关键图片通过解码器系统 1000 的路径和正向预测图片的路径。解码器系统 1000 的许多组件用于解压缩关键图片和预测图片两者。由那些组件执行的准确操作可以根据正在解压缩的信息的类型变化。

DEMUX 1090 接收用于压缩的视频序列的信息 1095 并使接收的信息可用于熵解码器 1080。DEMUX 1090 可包括抖动缓冲器(jitter buffer)和其它缓冲器。在 DEMUX 1090 之前或之后，压缩的视频信息可以被信道解码并为差错检测和校正对其处理。

熵解码器 1080 熵解码经熵编码的量化的数据和经熵编码的边信息(例如，对于运动信息 1015，量化步长)，一般应用编码器中所执行的熵编码的逆操作。熵解码技术包括算术解码(arithmetic decoding)，差分解码(differential decoding)，霍夫曼编码(Huffman decoding)，行程长度解码(run length

decoding), LZ 解码(LZ decoding), 词典解码(dictionary decoding), 以及上述的组合。熵解码器 1080 一般为不同类型的信息使用不同的解码技术(例如, DC 系数, AC 系数, 不同类型的边信息), 并可以从特定编码技术内的多个编码表中选择。

解码器 1000 解码运动信息 1015, 例如通过为运动矢量计算一个或多个预测值, 熵解码差分运动矢量, 以及将经解码的差分运动矢量与预测值组合起来重新构造运动矢量。下面描述用于为隔行扫描的 P 帧计算运动矢量预测值、计算差分运动矢量和重新构造运动矢量的各种技术。

运动补偿器 1030 将运动信息 1015 应用于一个或多个参考图片 1025 以形成正在重新构造的图片 1005 的预测 1035。例如, 运动补偿器 1030 使用一个或多个宏块运动矢量来查找参考图片 1025 中的宏块。一个或多个图片存储器(例如, 图片存储器 1020、1022)存储先前重新构造的图片用作参考图片。一般地, B 图片具有多于一个参考图片(例如, 至少一个在时间上在先的参考图片和至少一个在时间上在后的参考图片)。因此, 解码器系统 1000 可以使用分开的图片存储器 1020 和 1022 用于后向和前向参考图片。运动补偿器 1030 可以以像素、 $\frac{1}{2}$ 像素、 $\frac{1}{4}$ 像素或其它增量来补偿运动, 并可以在一个接一个图片基础或其它基础上切换运动补偿的分辨率。运动补偿的分辨率可以在水平或垂直上相同或不同。可供替换地, 运动补偿器应用另一种类型的运动补偿。通过运动补偿器的预测很少是完美的, 因此解码器 1000 还重新构造预测剩余部分。

逆量化器 1070 逆量化经熵解码的数据。一般而言, 逆量化器以在一个接一个图片基础或其它基础上变化的步长, 对经熵解码的数据应用均匀的、标量的逆量化。可供替换地, 逆量化器对数据应用其它类型的逆量化, 例如, 非均匀的、矢量的或非自适应的量化, 或者在不使用逆频率变换的解码器系统中直接逆量化空间域数据。

逆频率变换器 1060 将量化的、频域数据转换成空间域视频信息。对于基于块的视频图片, 逆频率变换器 1060 对频率变换系数的块应用逆 DCT["IDCT"]、IDCT 的变体或其它逆块变换, 分别为关键图片或预测图片产生像素数据或预测剩余部分数据。可供替换地, 逆频率变换器 1060 应用另外的常规逆频率变换诸如逆傅立叶变换或使用小波或子带综合。逆频率变换器 1060 可应用 8x8、8x4、4x8、4x4 或其它大小的逆频率变换。

对于预测图片, 解码器 1000 将重新构造的预测剩余部分 1045 与运动补偿

预测 1035 结合起来以形成重新构造的图片 1005。当解码器需要重新构造的图片 1005 用于随后的运动补偿，图片存储器之一(例如图片存储器 1020)缓冲重新构造的 1005 以在预测下一个图片时使用。在某些实施例中，解码器 1000 对重新构造的图片应用解块滤波器(de-blocking filter)以自适应地平滑图片中的不连续。

III. 运动矢量预测值

可以使用用于宏块(或块)的运动矢量来预测那些宏块(或块)的因果关系的近邻中的运动矢量。例如，编码器/解码器可以从用于相邻的候选宏块的运动矢量之中为当前宏块选择运动矢量预测值，并预测地使用该运动矢量预测值为当前宏块编码运动矢量。编码器/解码器可以使用三个值的中值(median-of-three)预测，四个值的中值(median-of-four)预测，或某种其它技术来来自相邻宏块的候选运动矢量之中为当前宏块确定运动矢量预测值。在图 13 的伪代码 1300 中描述用于三个值的中值预测的过程。在图 14 的伪代码 1400 中描述用于四个值的中值预测的过程。

IV. 用于隔行扫描图片的场编码

典型的隔行扫描视频帧由在不同时间扫描的两个场(例如，顶场和底场)组成。一般而言，通过将场编码在一起(“帧模式”编码)来编码隔行扫描视频帧的静止区域更有效。另一方面，通过独立地编码场(“场模式”编码)来编码隔行扫描视频帧的移动区域常常更有效，因为两个场倾向于不同的运动。正向预测的隔行扫描视频帧可被编码为两个独立的正向预测场--隔行扫描的 P 场。独立地为正向预测隔行扫描视频帧编码场可能是有效的，例如，当遍及隔行扫描的视频帧有高度运动，并因此在场之间有更多的差异。

A. 用于隔行扫描的 P 场的参考场

隔行扫描的 P 场引用一个或多个其它场(一般为先前的场，可以编码在比特流中或者可以不编码在比特流中)。例如，在某些实现中，隔行扫描的 P 场可具有一个或两个参考场。如果隔行扫描的 P 场具有两个参考场，则用于 P 场的块或宏块的特定运动矢量参考这两个参考帧中所选择的一个。图 15 和 16 示出具有两个参考场的隔行扫描的 P 场的例子。在图 15 中，当前场 1510 参考在

时间上在前的帧中的顶场 1520 和底场 1530。由于场 1540 和 1550 是隔行扫描的 B 场，它们不用作为参考场。在图 16 中，当前场 1610 参考紧接在包含当前场 1610 的隔行扫描视频帧之前的预测帧中的顶场 1620 和底场 1630。在其它情况下，隔行扫描的 P 场引用单个场，例如，最近的或次最近的 I 场或 P 场。

可供替换地，隔行扫描的 P 场可使用来自从不同类型或时间位置的帧得到的其它场的场作为参考场。

B. 场图片坐标系统和场极性

运动矢量单位可以用像素/子-像素(sub-pixel)单位或场单位来表示。例如，如果运动矢量的垂直分量表示六个四分之一像素单位的位移，这表示一又二分之一场行的位移，因为在场中每行一个像素高。

图 17 示出在一个实现中运动矢量的垂直分量与空间位置之间的关系。在图 17 中所示的例子示出三个不同情景 1710、1720 和 1730，用于当前和参考场类型(例如顶和底)的三种不同组合。如果当前和参考场的场类型不同，则极性“相反”。如果场类型相同，则极性“相同”。对于每一种情景，图 17 示出当前场中的像素的一个垂直列和参考场中的像素的第二垂直列。实际上，这两个列是水平对齐的。圆圈表示实际的整数像素位置而 X 表示内插的二分之一或四分之一像素位置。水平分量值(未示出)不需要考虑因隔行扫描引起的任何偏移，因为相应的场是水平对齐的。负值表示进一步向上偏移，且在相反的方向中，如所示的正值垂直偏移。

在情景 1710 中，极性“相反”。当前场是顶场而参考场是底场。相对于当前场，参考场的位置因隔行扫描在向下方向中偏移二分之一像素。因而，垂直运动矢量分量值 0 表示参考场中的位置，它在当前场中的位置之下偏移二分之一像素，作为默认的“无运动”值。垂直分量值+2 表示在当前场中位置之下偏移整个像素(以绝对值)的位置，它是参考场中的内插值，而垂直分量+4 表示在当前场中位置之下偏移一又二分之一像素(以绝对值)的位置，它是参考场中的实际值。

在情景 1720 中，极性也“相反”。当前场是底场而参考场是顶场。相对于当前场，参考场的位置因隔行扫描在向上方向中偏移二分之一像素。因而，垂直运动矢量分量 0 表示参考场中高于当前场中位置二分之一像素的位置(以绝对值)，垂直分量值+2 表示与当前场中位置相同水平(以绝对值)的位置，而垂

直分量+4表示在当前场中位置之下偏移二分之一像素的位置(以绝对值)。

在情景 1730 中,极性“相同”,没有应用垂直偏移,因为当前场的位置与相对于参考场的位置相同。

可供替换地,按照不同的惯例表示运动矢量的位移。

V. 在用于预测编码/解码隔行扫描的 P 场的运动矢量预测方面的创新

所述实施例包括用于编码和解码隔行扫描的视频(例如隔行扫描的 P 场)的技术和工具。所述技术和工具可以彼此结合或者结合其它技术和工具使用,或者可以独立地使用。

特别地,所述技术和工具规定例如为隔行扫描的 P 场的块和/或宏块产生运动矢量预测值的方法,其中隔行扫描的 P 场使用两个场作为参考用于运动补偿预测编码。所述技术和工具实现一个或多个创新,包括但不限于下列内容:

1. 产生两个运动矢量预测值—一个用于偶数场参考,一个用于奇数场参考。每个预测值是从三个先前编码的、候选的相邻运动矢量导出的。

2. 使用来自相同参考场的运动矢量预测值作为当前运动矢量(都参考相同的参考场)。

3. 定标一个场的实际运动矢量预测值候选值,为其它场产生运动矢量预测值候选用于运动矢量预测。

4. 使用定标操作(它使用当前场与两个参考场之间的相对时间距离),由另一种极性的现有运动矢量预测值候选,为一种参考极性导出运动矢量预测值候选。

在隔行扫描的视频中,可不使用运动预测(帧内或 I 场)、使用前向运行预测(P 场)或使用双向预测(B 场)来编码场。所述技术和工具包括为隔行扫描的 P 场的部分计算运动矢量预测值,尤其在可以参考两个最近的(以显示顺序)I 或 P 场中任何一个时发生运动补偿的情况下。假定两个参考场是相反极性的,即指一个参考场表示隔行扫描视频帧的奇数行,而另一个参考场表示相同或不同的隔行扫描视频帧的偶数行。这是在图 24 中表示的信令协议的情况,它示出用于隔行扫描的图片的场图片类型的语法元素(FPTYPE)。或者,例如,再次参考图 15,当前场 1510 参考顶场 1520 和底场 1530,它们分别表示视频帧的奇数和偶数行。

A. 在两个参考场图片中的运动矢量预测

用两个参考场 P-场，当前场可以在相同的时间方向上引用两个场（例如两个最近的在前参考场）。在两个参考场隔行扫描的 P 场的情况下，编码器和解码器为块或宏块的运动矢量在两个运动矢量预测值之间选择。在某些实施例中，一个预测值用于与当前场相同极性的参考场，而另一个用于相反极性的参考场。其它极性组合也有可能。

在某些实施例中，编码器/解码器通过查找奇数场预测值和偶数场为当前块或宏块计算运动矢量预测值，并选择一个预测值来处理宏块。例如，图 18 示出一种技术 1800，用于为具有两个可能的参考场的隔行扫描的 P 场的块或宏块计算运动矢量预测值。

在 1810，编码器/解码器确定奇数场运动矢量预测值和偶数场运动矢量预测值。运动矢量预测值之一因而具有与当前场相同的极性，而另一个运动矢量预测值具有相反极性。

在 1820，编码器/解码器从奇数场运动矢量预测值与偶数场运动矢量预测值之中选择运动矢量预测值。例如，编码器在运动矢量预测值之间基于哪一个给出更好的预测的运动矢量预测值来选择。或者，编码器选择参考相同参考场的运动矢量预测值作为当前正在预测的运动矢量。编码器使用简单的选择信号或使用更复杂的结合上下文信息(contextual information)以提高编码效率的信令，用信号表示要使用哪一个运动矢量预测值。上下文信息可表示奇数场或偶数场中的哪一个、或者相同极性场或相反极性场中的哪一个是在块或宏块周围的近邻中主要使用的。解码器基于选择信号和/或上下文信息来选择要使用哪一个运动矢量预测值。

在 1830，编码器/解码器使用所选择的运动矢量预测值来处理运动矢量。例如，编码器编码运动矢量与运动矢量预测值之间的差分。或者，解码器通过将运动矢量差分与运动矢量预测值结合起来解码运动矢量。

可供替换地，编码器和/或解码器可跳过确定奇数场运动矢量预测值或确定偶数场运动矢量预测值。例如，如果编码器确定将为用于特定块或宏块的运动补偿使用奇数场，则编码器只确定奇数场运动矢量预测值。或者，如果解码器根据上下文和/或用信号表示的信息确定将为运动补偿使用奇数场，则解码器只确定奇数场运动矢量预测值。因此，编码器和解码器可避免不必要的操作。

在一个实现中，解码器使用下列技术来为当前隔行扫描的 P 场确定运动矢

量预测值：

对于隔行扫描的 P 场中每一个具有运动矢量的块或宏块，获得两组三个候选的运动矢量预测值。在图 19 中示出相对于当前宏块 1900 的相邻宏块，从它们获得这些候选的运动矢量预测值。这些候选中的三个来自于偶数参考场，三个来自于奇数参考场。由于在每个候选方向 (A、B 或 C) 中相邻宏块将或者是帧内编码的或者具有实际的运动矢量(它或者引用偶数场或者引用奇数场)，因此有必要导出其它场的运动矢量。例如，对于给定的宏块，假定预测值 A 具有引用奇数场的运动矢量。在这种情况下，“偶数场”预测值候选 A 是从“奇数场”预测值候选 A 的运动矢量导出的。这个推导是使用定标操作来完成的。(例如，见下面图 21A 和 21B 的说明)。可供替换地，推导是以另一种方式完成的。

一旦已获得三个奇数场候选运动矢量预测值，就使用中值操作 (median operation) 来从三个奇数场候选中导出一个奇数场运动矢量预测值。同样，一旦已获得三个偶数场候选运动矢量预测值，就使用中值操作从三个偶数场候选中导出一个偶数场运动矢量预测值。可供替换地，使用另一种机制基于候选场运动矢量预测值来选择场运动矢量预测值。解码器决定使用偶数场还是使用奇数场作为运动矢量预测值(例如，通过选择主导预测值)，并使用偶数或奇数运动矢量预测值来重新构造运动矢量。

在图 20A-20F 中的伪代码 2000 例示用于从如在图 19 中安排的预测值 A、B、C 产生运动矢量预测值的过程。尽管图 19 示出在当前隔行扫描的 P 场的中间的一个典型宏块的近邻，但图 20A-20F 的伪代码 2000 处理宏块位置的各种特殊情况。此外，可使用伪代码 2000 为在各种位置的块的运动矢量计算运动矢量预测值。

在伪代码 2000 中，应该理解，术语“相同场”和“相反场”是相对于当前正在编码或解码的场而言的。例如，如果当前场是偶数场，则“相同场”是偶数参考场，而“相反场”是奇数参考场。伪代码 2000 中的变量 `samefieldpred_x` 和 `samefieldpred_y` 表示来自相同场的运动矢量预测值的水平和垂直分量，而变量 `oppositefieldpred_x` 和 `oppositefieldpred_y` 表示来自相反场的运动矢量预测值的水平和垂直分量。变量 `samecount` 和 `oppositecount` 分别跟踪有多少用于当前块或宏块的近邻的运动矢量引用对于当前场的“相同”的极性参考场和有多少引用“相反”极性参考场。在伪代码的开始处将变量 `samecount` 和 `oppositecount` 初始化为 0。

使用在伪代码 2000 中提到的定标操作 `scaleforsame()` 和 `scaleforopposite()` 从近邻的实际运动矢量值为“其它”场导出运动矢量预测值候选。定标操作是由实现决定的。下面参考图 21A、21B、22 和 23 描述示例定标操作。可供替换地，例如，使用其它定标操作来补偿诸如在图 17 中所示的垂直位移。

图 20A 和 20B 示出用于为典型的内部块或宏块计算运动矢量预测值。将用于“帧内”近邻的运动矢量设置为 0。对于每一近邻，设置相同场运动矢量预测值和相反场运动矢量预测值，其中一个是由近邻的运动矢量的实际值设置的，而另一个是从它导出的。为相同场运动矢量预测值和相反场运动矢量预测值计算候选的中值，并根据 `samecount` 和 `oppositecount` 确定“主导(dominant)”预测值。变量 `dominatepredictor` 表示哪一个场包含主导运动矢量预测值。如果运动矢量预测值具有与三个候选预测值中大多数相同的极性，则它是主导的。(用信号表示的值 `predictor_flag`，连同运动矢量差分数据一起被解码，表示使用主导还是非主导预测值)。

在图 20C 中的伪代码处理每行只有一个宏块的隔行扫描的 P 场中的宏块的情况，其中的宏块没有近邻 B 和 C。在图 20D 和 20E 中的伪代码处理在隔行扫描的 P 场左边缘的块或宏块的情况，其中的块或宏块没有近邻 C。这里，如果运动矢量预测值具有与多于两个候选预测值的相同极性，则它是主导的，而在平分(tie)的情况下，奇数场运动矢量预测值是主导的。最后，在图 20F 中的伪代码处理例如隔行扫描的 P 场的首行中的宏块的情况。

B. 用于从另一个场运动矢量预测值导出一个场运动矢量预测值的定标

在一个实施例中，编码器/解码器使用在图 21A 和 21B 的伪代码中例示的定标操作从另一个场运动矢量预测值导出一个场运动矢量预测值。`SCALEOPP`、`SCALESAME1`、`SCALESAME2`、`SCALEZONE1_X`、`SCALEZONE2_Y`、`ZONE1OFFSET_X` 和 `ZONE1OFFSET_Y` 的值是由实现决定的。图 22 的表 2200 中和图 23 的表 2300 中所示的两个可能的值集合，前者用于当前场是隔行扫描的视频帧中的第一个场，而后者用于当前场是隔行扫描视频帧中的第二个场。在表 2200 和 2300 中，参考帧距离是作为当前场与包含在时间上最远的参考场之间的 B 图片(即包括两个 B 场的视频帧)的数量来定义的。

在表 2200 与表 2300 中所示的例子中，N 的值取决于运动矢量范围。例如，

扩展的运动矢量范围可以通过语法元素 EXTENDED_MV=1 来用信号表示。如果 EXTENDED_MV=1, 则 MVRANGE 语法元素存在于图片首部中并用信号表示运动矢量范围。如果 EXTENDED_MV=0, 则使用默认的运动矢量范围。表 1 示出 N 与 MVRANGE 之间的关系。

表 1: 在图 22 和 23 中 N 的推导

MVRANGE(运动矢量范围)	N
0 或默认值	1
10	2
110	8
111	16

在表 2200 和 2300 中的值可以根据实现来修改。

可供替换地, 不使用扩展的运动矢量范围信息, 或者以某种其它方法来执行定标。

VI. 组合的实现

现在描述用于比特流语法和解码器的详细的组合实现, 之外还描述具有与主组合实现微小差别的可替换组合实现。

A. 比特流语法

在各种组合实现中, 以具有多个层(layer)(例如, 序列、帧、场、宏块、块和/或子块层)的比特流的形式提供用于隔行扫描的 P 场的的数据。用于包括隔行扫描的 P 场的每一帧的数据由帧首部和跟随其后的用于场层的数据组成。在图 24 中示出构成用于包括隔行扫描的 P 场的帧的帧首部的比特流元素。在图 25 中示出构成用于隔行扫描的 P 场的场首部的比特流元素。在图 26 中示出构成用于隔行扫描的 P 场的宏块层的比特流元素。下面的章节描述在涉及用信号表示隔行扫描的 P 场的块或宏块的运动矢量预测值的帧、场和宏块层中所选择的比特流元素。

I. 选择的帧层元素

图 24 示出在组合实现中用于包括隔行扫描的 P 场的帧的帧级 (frame-level) 比特流语法。下面描述具体的比特流元素。

帧编码模式 (Frame Coding Mode) (FCM) (可变大小)

FCM 是可变长度代码字 (variable length codeword) [“VLC”], 用于表示图片片码类型。FCM 取下面在图 2 中所示的用于帧编码模式的值:

表 2: 帧编码模式 VLC

FCM 值	帧编码模式
0	逐行扫描
10	帧-隔行扫描
11	场-隔行扫描

场图片类型 (Field Picture Type) (FPTYPE) (3 位)

FPTYPE 是在用于包括隔行扫描的 P 场的帧的帧首部中存在的三比特语法元素。FPTYPE 取按照下面表 3 用于不同组合的场类型的值。

表 3: 场图片类型 FLC

FPTYPE FLC	第一场类型	第二场类型
000	I	I
001	I	P
010	P	I
011	P	P
100	B	B
101	B	BI
110	BI	B
111	BI	BI

P 参考距离 (P Reference Distance) (REFDIST) (可变大小)

REFDIST 是可变大小的语法元素。这个元素表示当前帧与参考帧之间的帧

数量。表 4 示出用于编码 REF DIST 值的 VLC。

表 4: REF DIST VLC 表

参考帧距离	VLC 代码字(二进制)	VLC 大小
0	00	2
1	01	2
2	10	2
N	11[(N-3) 1s]0	N

表 4 中最后一行表示用于表示比 2 大的参考帧距离的代码字。它们被编码为(二进制)11, 跟随着 N-3 个 1, 其中 N 是参考帧距离。代码字中的最后位是 0。例如:

N=3, VLC 代码字=110, VLC 大小=3

N=4, VLC 代码字=1110, VLC 大小=4

N=5, VLC 代码字=11110, VLC 大小=5

在可替换的组合实现中, 在用于隔行扫描的 P 场的场层的开始处代替在用于包括隔行扫描的 P 场的帧的帧层处用信号表示图片类型信息, 并省略 P 参考距离。

2. 选择的场层元素

图 25 示出在组合实现中用于隔行扫描的 P 场的场级(field-level)比特流语法。下面描述具体的比特流元素。

参考图片的数量(Number of Reference Pictures) (NUMREF) (1 比特)

NUMREF 语法元素是一个一比特语法元素, 表示当前场引用一个还是两个先前的参考场图片。如果 NUMREF=0, 则当前 P 场图片可只引用一个场。在这种情况下, 在图片层比特流中跟随着 REFFIELD 语法元素。对于具有两个参考场的隔行扫描的 P 场, NUMREF=1。

参考场图片指示器(Reference Field Picture Indicator) (REFFIELD) (1 比特)

如果 NUMREF=0, REFFIELD 是在隔行扫描的 P 场图片首部中存在的一个 1 位语法元素。REFFIELD 表示使用哪一个先前解码的场作为参考。如果 REFFIELD=0, 则使用在时间上最近的(以显示顺序)I 或 P 场作为参考。如果 REFFIELD=1, 则使用在时间上次最近的 I 或 P 场图片作为参考。

扩展的 MV 范围标志(Extended MV Range Flag) (MVRANGE) (可变大小)

MVRANGE 是在序列层 EXTENDED_MV 位设置为 1 时存在的可变大小的语法元素。MVRANGE VLC 表示运动矢量范围。

扩展的差分 MV 范围标志(Extended Differential MV Range Flag) (DMVRANGE) (可变大小)

如果序列级语法元素 EXTENDED_DMV=1, 则存在 DMVRANGE, 它是可变大小的语法元素。DMVRANGE VLC 表示运动矢量差分范围。

运动矢量模式(Motion Vector Mode) (MVMODE) (可变大小或 1 比特)

MVMODE 语法元素用信号表示四种运动矢量编码模式之一或一种强度补偿模式。若干后续的元素提供另外的运动矢量模式和/或强度补偿信息。

宏块模式表(Macroblock Mode Table) (MBMODETAB) (2 或 3 比特)

MBMODETAB 语法元素是定长字段。对于隔行扫描的 P 场, MBMODETAB 是 3 比特值, 表示使用八个霍夫曼表中的哪一个来解码宏块层中的宏块模式语法元素(MBMODE)。

运动矢量表(Motion Vector Table) (MVTAB) (2 或 3 比特)

MVTAB 语法元素是一个 2 或 3 位值。对于其中 NUMREF=1 的隔行扫描的 P 场, MVTAB 是 3 比特语法元素, 表示使用八个隔行扫描的霍夫曼表中的哪些来解码运动矢量数据。

4MV 块图案表(4MV Block Pattern Table) (4MVBPTAB) (2 比特)

4MVBPTAB 语法元素是一个 2 比特值。对于隔行扫描的 P 场, 只有当 MVMODE(或 MVMODE2, 如果 MVMODE 设置为强度补偿)表示图片是“混合的 MV”类

型时才提供它。4MVBPTAB 语法元素用信号表示使用四个霍夫曼表中的哪一个来解码 4MV 宏块中的 4MV 块图案(4MVBP)语法元素。

3. 选择的宏块层元素

图 26 示出在组合实现中用于隔行扫描的 P 场的宏块级(macroblock-level)比特流语法。下面描述具体的比特流元素。用于宏块的数据由宏块头部和跟随其后的块层数据构成。

宏块模式(Macroblock Mode)(MBMODE)(可变大小)

MBMODE 语法元素表示宏块类型(1MV、4MV 或帧内), 并存在 CBP 标志和运动矢量数据, 如在下面 VI. B. 3. 节中详细描述。

运动矢量数据(Motion Vector Data)(MVDATA)(可变大小)

MVDATA 是可变大小的语法元素, 它编码用于宏块的运动矢量的差分, 在下面 VI. B. 3. 节中详细描述它的解码。

4MV 块图案(4MV Block Pattern)(4MVBP)(4 比特)

4MVBP 语法元素表示 4 个亮度块中哪一个包含非零运动矢量差分, 在下面 VI. B. 3. 节中详细描述其使用。

块级运动矢量数据(Block-level Motion Vector)(BLKMVDATA)(可变大小)

BLKMVDATA 是可变大小的语法元素, 包含用于块的运动信息, 并存在于 4MV 宏块中。

混合运动矢量预测(Hybrid Motion Vector Prediction)(HYBRIDPRED)(1 比特)

HYBRIDPRED 是每运动矢量 1 比特语法元素。如果预测值显式地编码在比特流中, 则提供一个比特表示使用预测值 A 还是预测值 C 作为运动矢量预测值。

B. 解码隔行扫描的 P 场

下面的章节描述用于在组合实现中解码隔行扫描的 P 场的过程。

1. 帧/场层解码

参考图片

P 场图片可引用一个或两个先前编码的场。在图片层中的 NUMREF 语法元素是一个一比特语法元素，表示当前场可引用一个还是两个先前的参考场图片。如果 NUMREF=0，则当前隔行扫描的 P 场图片的块和宏块可只引用一个场。如果 NUMREF=1，则当前隔行扫描的 P 场图片的块或宏块可使用两个在时间上最近的（以显示顺序）I 或 P 场图片中的任何一个作为参考。

图 15 和 16 示出两个参考 P-场的例子。

P 场图片类型

隔行扫描的 P 场图片可以两种类型之一：1MV 或混合的 MV (Mixed-MV)。下面的章节描述每种类型。在 1MV P 场中，每个运动补偿的宏块使用单个运动矢量来表示用于在宏块中所有 6 个块的预测的块的位移。1MV 模式是由 MVMODE 和 MVMODE2 图片层语法元素用信号表示的。

在混合的 MV P 场中，每个运动补偿的宏块可编码为 1MV 或 4MV 宏块。在 4MV 宏块中，4 个亮度块的每一个具有与其相关联的运动矢量。在混合的 MV P 场中，用于每一个宏块的 1MV 或 4MV 是由 MBMODE 语法元素在每个宏块表示的。混合的 MV 模式是由 MVMODE 和 MVMODE2 图片层语法元素用信号表示的。

2. 宏块层解码

在隔行扫描的 P 场中的宏块可以是 3 种可能类型之一：1MV、4MV 和帧内 (Intra)。宏块类型是由宏块层中的 MBMODE 语法元素用信号表示的。下面的章节描述 1MV 和 4MV 类型以及如何用信号表示它们。

1MV 宏块

1MV 宏块可在 1MV 和混合的 MV P 场图片中出现。在 1MV 宏块中，对于宏块中所有 6 个宏块，单个运动矢量代表当前和参考图片之间的位移。对于 1MV 宏块，在宏块层中的 MBMODE 语法元素表示三项内容：

- 1) 宏块类型是 1MV
- 2) CBPCY 语法元素是否存在

3) MVDATA 语法元素是否存在

如果 MBMODE 语法元素表示 CBPCY 语法元素存在, 则 CBPCY 语法元素存在于相应位置中的宏块层中。CBPCY 表示 6 个块中哪些被编码在块层中。如果 MBMODE 语法元素表示 CBPCY 语法元素不存在, 则假定 CBPCY 等于 0 且不存在用于宏块中 6 个块的任何一个的块数据。

如果 MBMODE 语法元素表示 MVDATA 语法元素存在, 则 MVDATA 存在于相应位置中的宏块层中。MVDATA 语法元素编码运动矢量差分。将运动矢量差分与运动矢量预测值结合起来重新构造运动矢量。如果 MBMODE 语法元素表示 MVDATA 语法元素不存在, 则假定运动矢量差分是零并因此运动矢量等于运动矢量预测值。

4MV 宏块

4MV 宏块只可在混合的 MV 图片中出现。在 4MV 宏块中, 在宏块中的 4 个亮度块的每一个具有相关联的运动矢量, 它表示用于那个块的当前和参考图片之间的位移。用于色度块的位移是从 4 个亮度运动矢量导出的。

对于 4MV 宏块, 宏块层中的 MBMODE 语法元素表示三项内容:

- 1) 宏块类型是 4MV
- 2) CBPCY 语法元素是否存在

4MVBP 语法元素表示 4 个亮度块中的哪些包含非零运动矢量差分。4MVBP 语法元素解码成 0 与 15 之间的值。对于 4MVBP 中的 4 个比特位置 (bit position) 的每一个, 值 0 表示不存在用于那个块的运动矢量差分 (BLKMVDATA) 并假定运动矢量差分是 0。值 1 表示在相应位置中存在用于那个块的运动矢量差分 (BLKMVDATA)。例如, 如果 4MVBP 解码成值 1100 (二进制), 则比特流包含用于块 0 和 1 的 BLKMVDATA 而没有用于块 2 和 3 的 BLKMVDATA。

在可替换的实现中, MBMODE 语法元素可以表示 4MVBP 语法元素是否存在; 如果 MBMODE 语法元素表示 4MVBP 语法元素不存在, 则假定用于所有 4 个亮度块的运动矢量差分数据 (BLKMVDATA) 存在。

取决于 MVMODE/MVMODE2 语法元素表示混合的 MV 还是表示全部 1MV (all-1MV), MBMODE 用信号表示如下的信息。表 5 示出 MBMODE 如何用信号表示关于在全部 1MV 图片中的宏块的信息。

表 5: 在全部-1MV 图片中的宏块模式

索引	宏块类型	CBP 存在	MV 存在
0	帧内	否	无效
1	帧内	是	无效
2	1MV	否	否
3	1MV	否	是
4	1MV	是	否
5	1MV	是	是

表 6 示出 MBMODE 如何用信号表示关于混合的 MV 图片中的宏块的信息。

表 6: 在混合的 1MV 图片中的宏块模式

索引	宏块类型	CBP 存在	MV 存在
0	帧内	否	无效
1	帧内	是	无效
2	1MV	否	否
3	1MV	否	是
4	1MV	是	否
5	1MV	是	是
6	4MV	否	无效
7	4MV	是	无效

使用 8 个表之一来用信号表示 MBMODE。在图片层处通过 MBMODETAB 语法元素用信号表示该表。

3. 运动矢量解码过程

下列章节描述用于 P 场图片的块或宏块的运动矢量解码过程。

解码运动矢量差分

MVDATA 或 BLKMVDATA 语法元素为宏块中的宏块或块编码运动信息。1MV 宏

块具有单个 MVDATA 语法元素，而 4MV 宏块可具有零与四个之间的 BLKMVDATA 元素。下面的章节描述如何为一个参考 (one-reference) (图片层语法元素 NUMREF=0) 和两个参考 (two-reference) (图片层语法元素 NUMREF=1) 情况计算运动矢量差分。

在一个参考场图片中的运动矢量差分

在只有一个参考场的场图片中，在宏块层中每个 MVDATA 或 BLKMVDATA 语法元素共同地编码两项内容：1) 水平运动矢量差分分量和 2) 垂直运动矢量差分分量。

MVDATA 或 BLKMVDATA 语法元素是可变长的霍夫曼码字 (Huffman codeword) 和跟随着的定长码字。霍夫曼码字的值确定定长码字的大小。在图片层中的 MVTAB 语法元素指定用于解码可变大小的码字的霍夫曼表。

在图 27A 中的伪代码 2700 例示如何为一个参考场解码运动矢量差分。在伪 2700 中计算值 dmv_x 和 dmv_y 。按如下定义这些值：

dmv_x : 差分水平运动矢量分量，

dmv_y : 差分垂直运动矢量分量，

k_x, k_y : 用于长运动矢量的固定长度，

k_x 和 k_y 取决于如由 MVRANGE 符号定义的运动矢量范围。

表 7: 由 MVRANGE 指定的 k_x 和 k_y

MVRANGE	k_x	k_y	range_x	range_y
0 (默认值)	9	8	256	128
10	10	9	512	256
110	12	10	2048	512
111	13	11	4096	1024

$extend_x$: 用于水平运动矢量差分的扩展范围，

$extend_y$: 用于垂直运动矢量差分的扩展范围，

$extend_x$ 和 $extend_y$ 是从 DMVRANGE 图片场语法元素导出的。如果 DMVRANGE 表示使用用于水平分量的扩展范围，则 $extend_x=1$ 。否则 $extend_x=0$ 。同样，如果 DMVRANGE 表示使用用于垂直分量的扩展范围，则 $extend_y=1$ ，否则 $extend_y=0$ 。

`offset_table`(偏移表)是在伪代码 2700 中使用的数组, 并如在图 27A 中所示的那样定义。

在图 27B 中的伪代码 2710 例示在可替换的组合实现中如何为一个参考场编码运动矢量差分。伪代码 2710 以不同的方式解码运动矢量差分。例如, 伪代码 2710 省略了扩展的运动矢量差分范围的处理。

在两个参考场图片中的运动矢量差分

在使用场图片编码隔行扫描帧时出现两个参考场图片。序列的每个帧被分成两个场, 每个场使用实际上是逐行扫描的编码路径来编码每个场。场图片常常具有两个参考场, 并在下面描述在这种情况下场图片运动矢量的编码。

在具有两个参考场的场图片中, 在宏块层中的每个 `MVDATA` 或 `BLKMVDATA` 语法元素共同地编码三项内容: 1) 水平运动矢量差分分量, 2) 垂直运动矢量差分分量和 3) 是否使用主导或非主导的预测值, 即由运动矢量引用两个场中的哪一个。

`MVDATA` 或 `BLKMVDATA` 语法元素是变长的霍夫曼码字并跟随着的定长码字。霍夫曼码字的值确定定长码字的大小。在图片层中的 `MVTAB` 语法元素指定用于解码可变大小的码字的霍夫曼表。

在图 28A 中的伪代码 2800 例示如何编码运动矢量差分和主导/非主导的预测值信息。

在图 28A 中的伪代码 2800 中计算值 `predictor_flag`、`dmv_x` 和 `dmv_y`。除在图 27A 中的伪代码 2700 中所示的变量和数组之外, 伪代码 2800 包含变量 `predictor_flag`, 它是表示使用主导还是非主导运动矢量预测值的二进制标志 (0=使用主导预测值, 1=使用非主导的预测值), 以及表 `size_table`, 它是如在图 28 中所示的那样定义的数组。

在图 28B 中的伪代码 2810 例示在可替换的组合实现中如何为两个参考场解码运动矢量差分。伪代码 2810 以不同的方法解码运动矢量差分。例如, 伪代码 2810 省略了扩展的运动矢量差分范围的处理。

运动矢量预测值

通过将前一节中计算的运动矢量差分加到运动矢量预测值上来计算运动矢量。预测值是由多达三个相邻的运动矢量来计算的。下面的章节描述在这个

组合实现中如何为 1MV P 场图片和混合的 MV P 场图片计算运动矢量预测值。

在 1MV 隔行扫描的 P 场中的运动矢量预测值

图 29A 和 29B 示出为隔行扫描的 P 场的 1MV 宏块的候选运动矢量预测值考虑的宏块位置。在 1MV 隔行扫描的 P 场中，从左边、上部和右上的宏块取候选预测值，除了在宏块是行中最后一个宏块的情况之外。在这种情况下，预测值 B 是从左上的宏块取得的，而不是右上。对于帧是一个宏块宽的特殊情况，预测值总是预测值 A(上部的预测值)。当前宏块在首行中的特殊情况(没有 A 和 B 预测值，或者根本没有预测值)，参考上面图 20A-20F 来处理。

在混合的 MV P 图片中的运动矢量预测值

图 30A-34 示出为混合的 MV P 场图片中的 1MV 或 4MV 宏块的多达 3 个候选运动矢量预测值考虑的宏块位置。在下列图中，较大的正方形是宏块边界，而较小的正方形是块边界。对于帧是一个宏块宽的特殊情况，预测值总是预测值 A(上部预测值)。当前块或宏块在顶行中的特殊情况是参考上面图 20A-20F 来处理的。

图 30A 和 30B 示出为混合的 1MV/4MV 隔行扫描的 P 场中的 1MV 当前宏块的候选运动矢量预测值考虑的块位置。相邻的宏块可以是 1MV 或 4MV 宏块。图 30A 和 30B 示出假定近邻是 4MV 时(即预测值 A 是在当前宏块之上的宏块中的块 2 的运动矢量,而预测值 C 是紧接着当前宏块的左边的宏块中的块 1 的运动矢量)候选运动矢量的位置。如果任何近邻是 1MV 宏块，则取在图 29A 和 29B 中所示的运动矢量预测值作为用于整个宏块的矢量。如图 30B 所示，如果宏块是行中最后的宏块，则预测值 B 来自左上的宏块中的块 3 而不是来自右上的宏块中的块 2，否则反之。

图 31A-34 示出为在 4MV 宏块中的 4 个亮度块的每一个的候选运动矢量预测值考虑的块位置。图 31A 和 31B 示出为在混合的 1MV/4MV 隔行扫描的 P 场中的 4MV 宏块中在位置 0 处的块的候选运动矢量预测值考虑的块位置；图 32A 和 32B 示出为在混合的隔行扫描的 P 场中的 4MV 宏块中在位置 1 处的块的候选运动矢量预测值考虑的块位置；图 33 示出为在混合的 1MV/4MV 隔行扫描的 P 场中的 4MV 宏块中在位置 2 处的块的候选运动矢量预测值考虑的块位置；以及图 34 示出为在混合的 1MV/4MV 隔行扫描的 P 场中的 4MV 宏块中在位置 3 处的块的

候选运动矢量预测值考虑的块位置。再一次，如果近邻是 1MV 宏块，则为宏块的块使用用于宏块的运动矢量预测值。

对于宏块是行中第一个宏块的情况，与行中其余宏块的块 0 不同地处理块 0 的预测值 B。在这种情况下，从紧接在当前宏块之上的宏块中的块 3 而不是从当前宏块之上和左边的宏块中的块 3 来取预测值 B，否则反之。同样，对于宏块是行中最后一个宏块，不同地处理块 1 的预测值 B。在这种情况下，从紧接在当前宏块之上的宏块中的块 2 而不是从当前宏块之上和右边的宏块中的块 2 来取预测值 B，否则反之。如果宏块在第一宏块列中，则将块 0 和 2 的预测值 C 设置为等于 0。

主导和非主导的 MV 预测值

在两个参考场 P 场图片中，对于每个帧间编码的宏块，导出两个运动矢量预测值。一个来自主导场，而另一个来自非主导场。主导场被认为是包含近邻的实际值运动矢量预测值候选的大部分的场。在平分的情况下，用于相反场的运动矢量预测值被认为是主导预测值。在主导/非主导预测值的计算中不考虑帧内编码的宏块。如果所有候选预测值宏块是帧内编码的，则主导和非主导运动矢量预测值设置为零，并从相反的场取主导预测值。

计算运动矢量预测值

如果 NUMREF=1，则当前场图片可参考两个最近的场图片，且为块或宏块的每个运动矢量计算两个运动矢量预测值。在图 35A-35F 中的伪代码 3500 描述在组合实现中如何为两个参考情况计算运动矢量预测值。（在图 20A-20F 中的伪代码 2000 描述在另一个实现中如何为两个参考情况计算运动矢量预测值）。在两个参考图片 (NUMREF=1) 中，当前场可引用两个最近的场。一个预测值用于相同极性的参考场，而另一个用于具有相反极性的参考场。

在组合实现中的定标操作

图 21A-B 是代码图，示出用于定标来自一个场的预测值来导出来自另一个场的预测值的伪代码 2100。在图 22 的表 2200 中（对于当前场是第一个场的情况）和在图 23 的表 2300 中（对于当前场是第二个场的情况），示出在这个组合实现中的 SCALEOPP、SCALESAME1、SCALESAME2、SCALEZONE1_X、SCALEZONE1_Y、

ZONE1OFFSET_X 和 ZONE1OFFSET_Y 的值。在图片首部中的 REFDIST 中编码参考帧距离。参考帧距离是 REFDIST+1。

重新构造运动矢量

下面的章节描述如何为 1MV 和 4MV 宏块重新构造亮度和色度运动矢量。在重新构造运动矢量之后，随后可使用它作为相邻的运动矢量来为邻近的宏块预测运动矢量。如果运动矢量用于两个参考场隔行扫描的 P 场中的块或宏块，则运动矢量将具有相关联的“相同”或“相反”的极性，且可用于为运动矢量预测值的其它场极性导出运动矢量预测值。

亮度运动矢量重新构造

在所有情况(1MV 和 4MV 宏块)下，通过如下将差分加到预测值上来重新构造亮度运动矢量：

$$mv_x=(dmv_x + predictor_x)smod\ range_x$$

$$mv_y=(dmv_y + predictor_y)smod\ range_y$$

模操作“smod”是带符号模，按如下定义：

$$A\ smod\ b = ((A + b) \% (2*b)) - b$$

这确保重新构造的矢量是有效的。(A smod b)位于-b 与 b-1 之内。range_x 和 range_y 取决于 MVRANGE。

如果隔行扫描的 P 场图片使用两个参考图片(NUMREF=1)，则将在解码运动矢量差分之后导出的 predictor_flag 与从运动矢量预测值导出的 dominatpredictor 值结合起来以确定使用哪一个场作为参考。在图 36 中的伪代码 3600 描述如何确定参考场。

在 1MV 宏块中，将存在用于 4 个构成宏块的亮度分量的块的单个运动矢量。如果 MBMODE 语法元素表示在宏块层中不存在 MV 数据，则 dmv_x=0 和 dmv_y=0(mv_x=predictor_x 且 mv_y=predictor_y)。

在 4MV 宏块中，在宏块中的每一个帧间编码的亮度块将具有其自己的运动矢量。因此在每个 4MV 宏块中将存在 0 与 4 个之间的亮度运动矢量。如果 4MVBP 语法元素表示不存在用于块的运动矢量信息，则对于那个块 dmv_x=0 和 dmv_y(mv_x=predictor_x 且 mv_y=predictor_y)。

色度运动矢量重新构造

色度运动矢量是从亮度运动矢量导出的。在可替换的实现中，对于 4MV 宏块，基于亮度块或场的状态作出关于将色度块作为帧间还是帧内编码的决定。

已经参考各种实施例描述和例示了我的发明的原理，将认识到，可以在不脱离这样的原理的情况下在安排和细节上修改各种实施例。应该理解，在此描述的程序、过程或方法与任何特定类型的计算环境无关，或者不受限于它们，除非另有说明。可与各种类型的通用或专用计算环境一起使用或执行按照在此描述的教导的操作。在软件中所示的实施例的元素可在硬件中实现，反之亦然。

由于许多可能的实施例可应用我的发明原理，我要求我的发明所有这样的实施例都落在所附的权利要求书及其等价物的范围和精神之中。

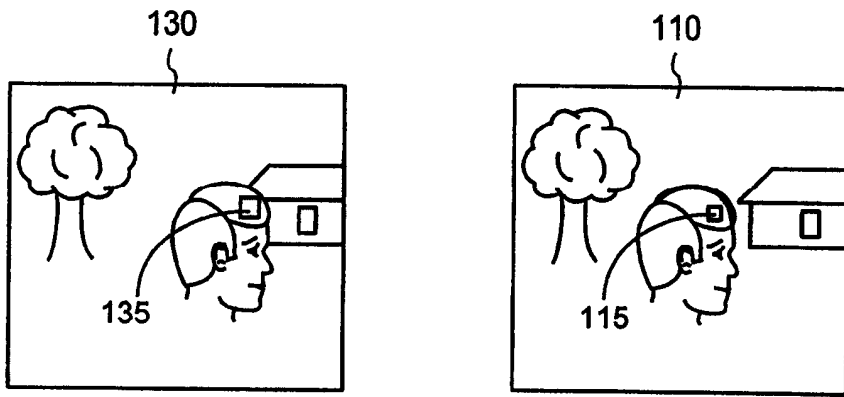


图 1
背景技术

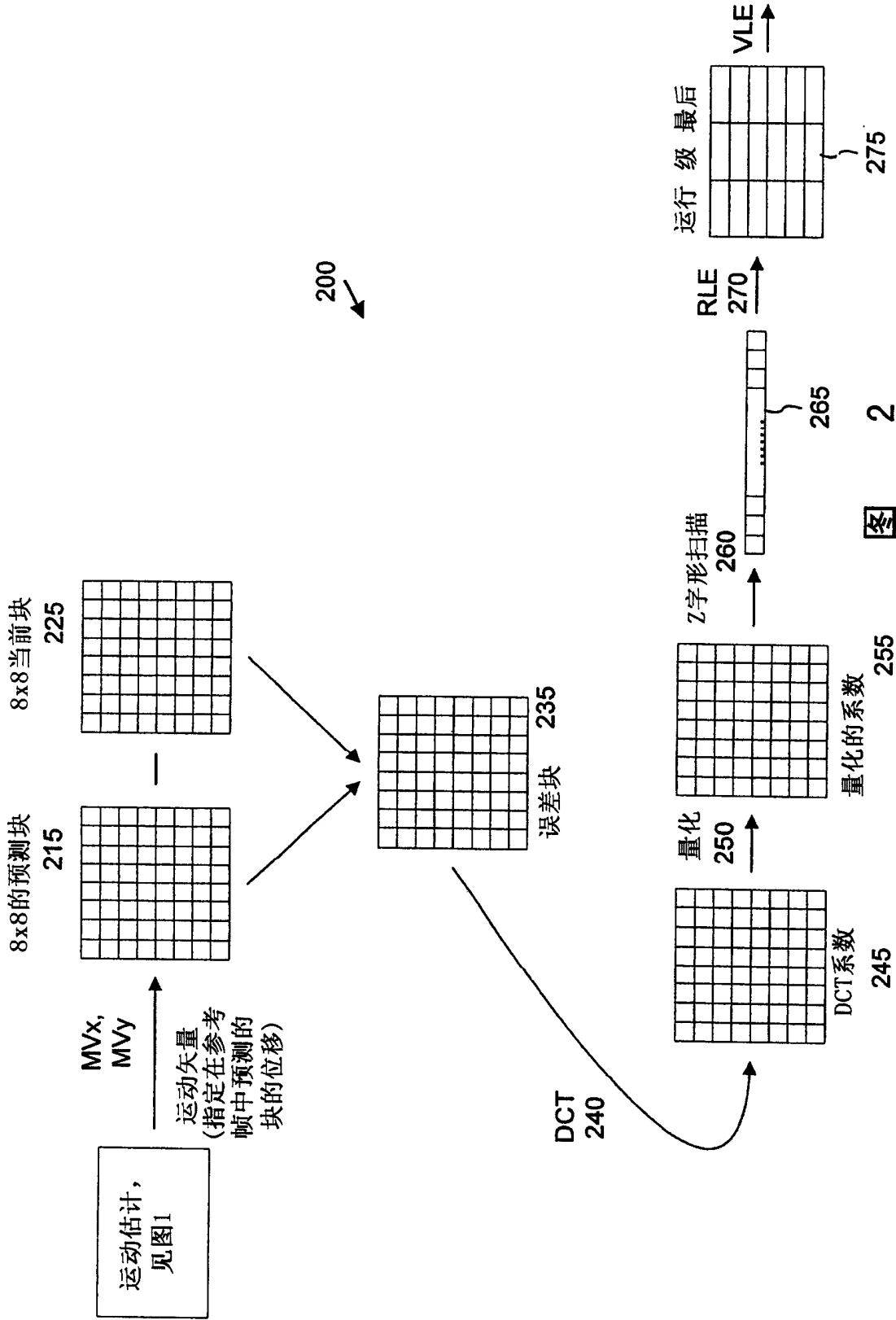


图 2 背景技术

200

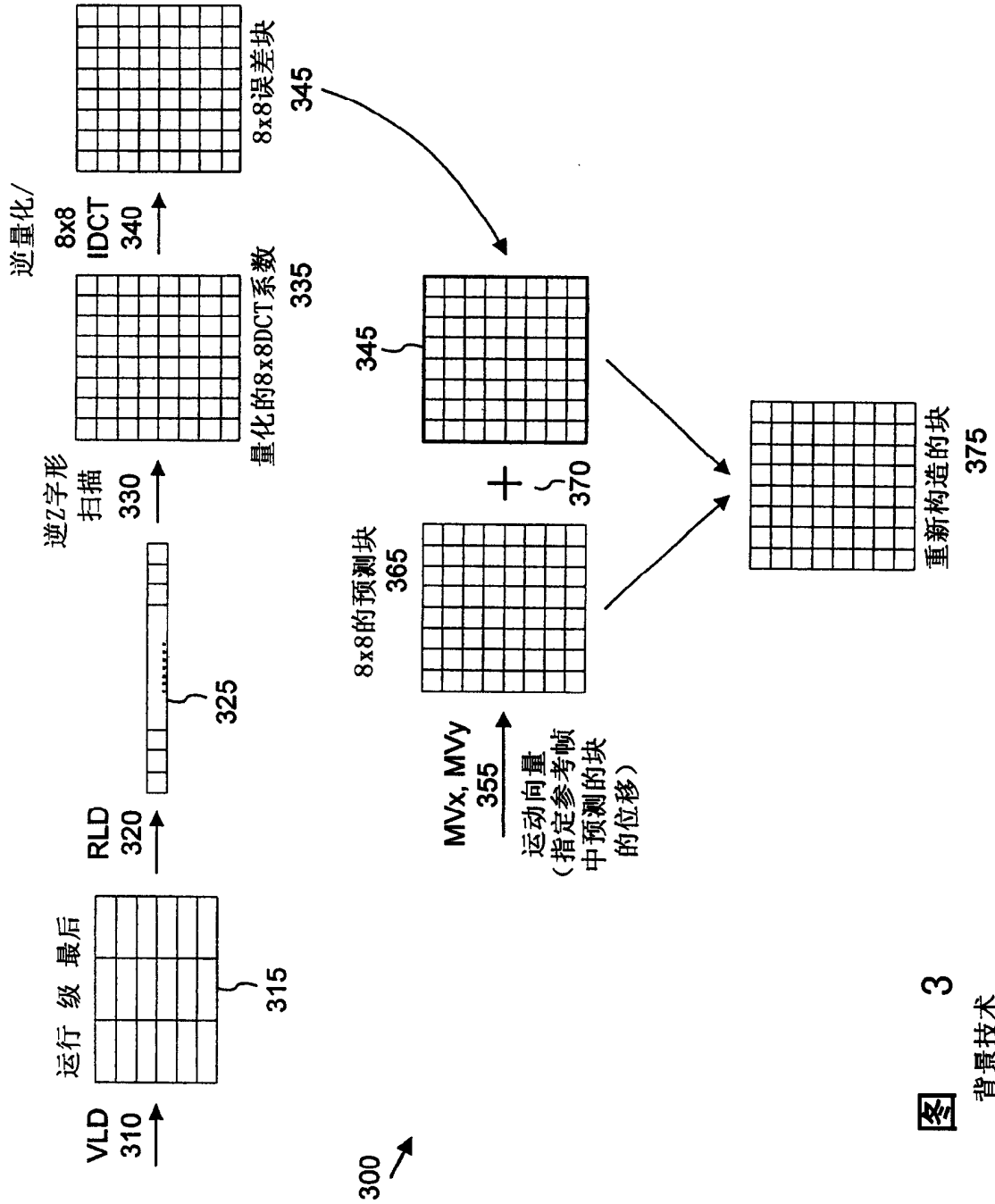


图 3
背景技术

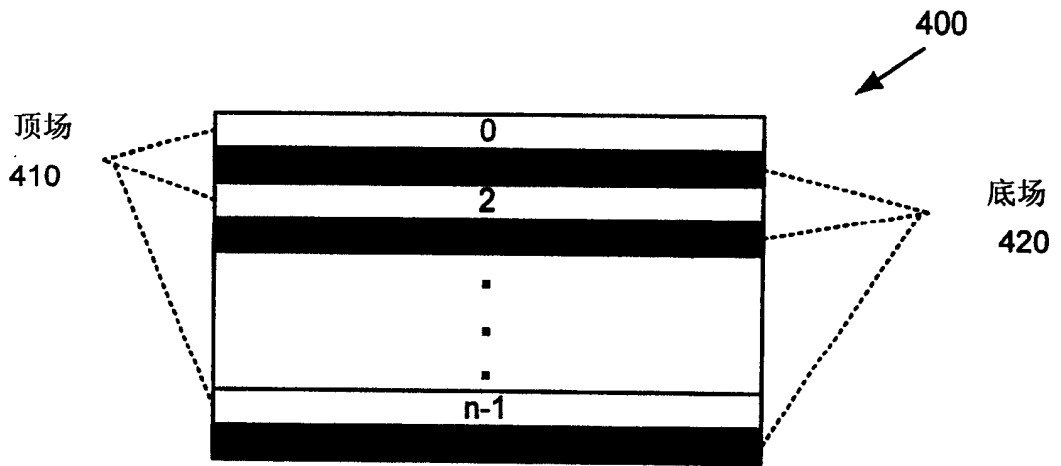


图 4
背景技术

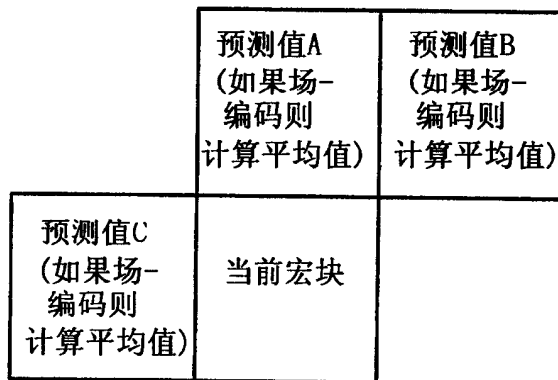


图 5
背景技术

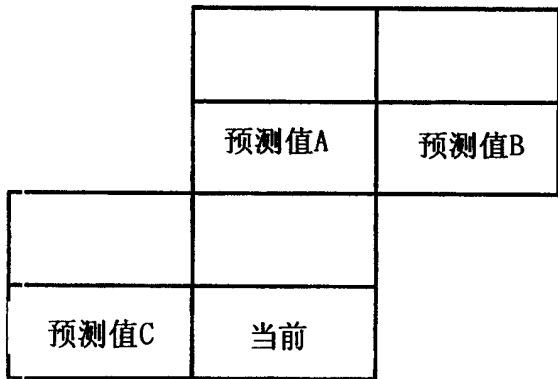


图 6
背景技术

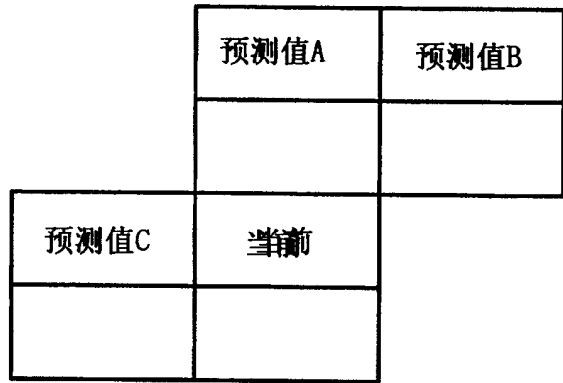
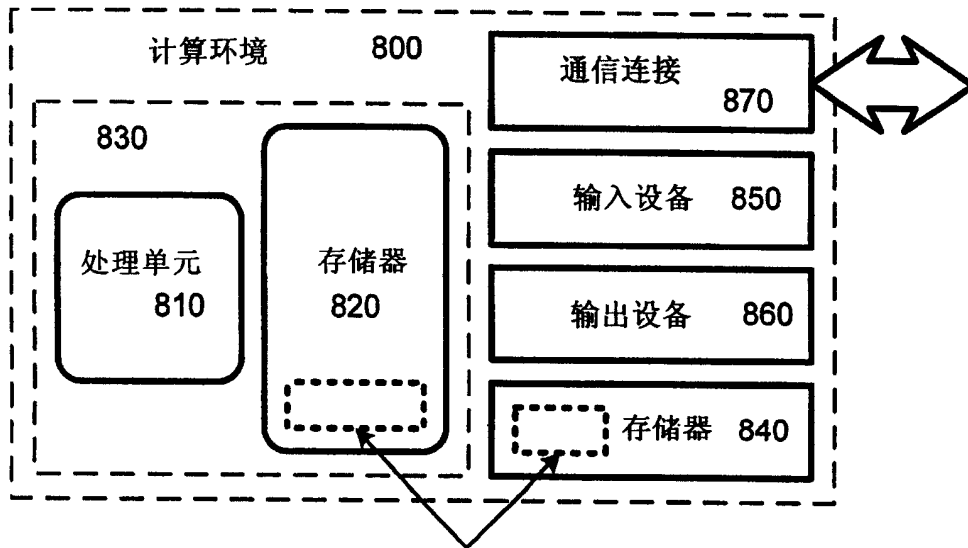
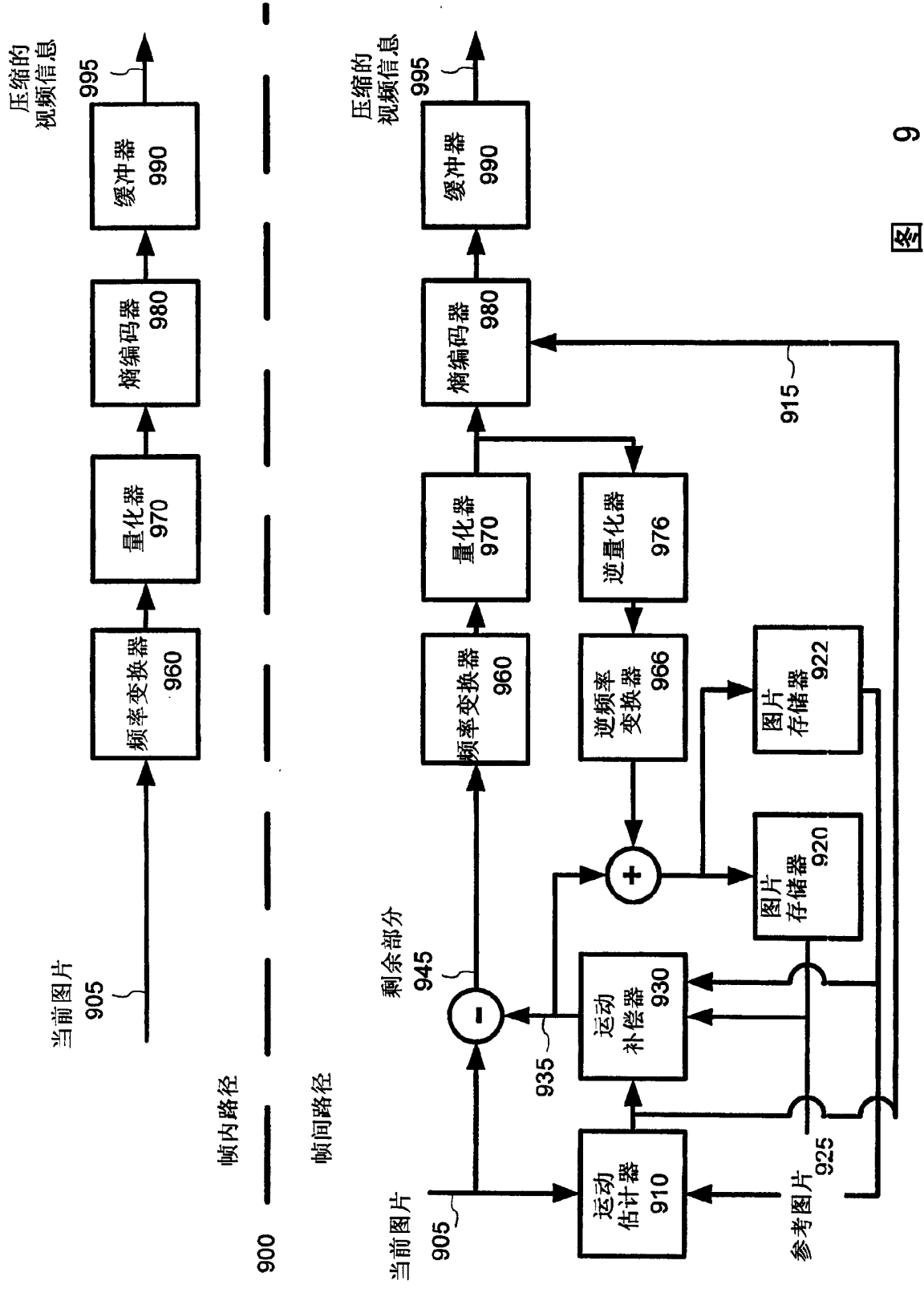


图 7
背景技术



实现具有用于隔行扫描的P场的两个参考场运动向量
预测值的视频编码器或解码器软件880

图 8



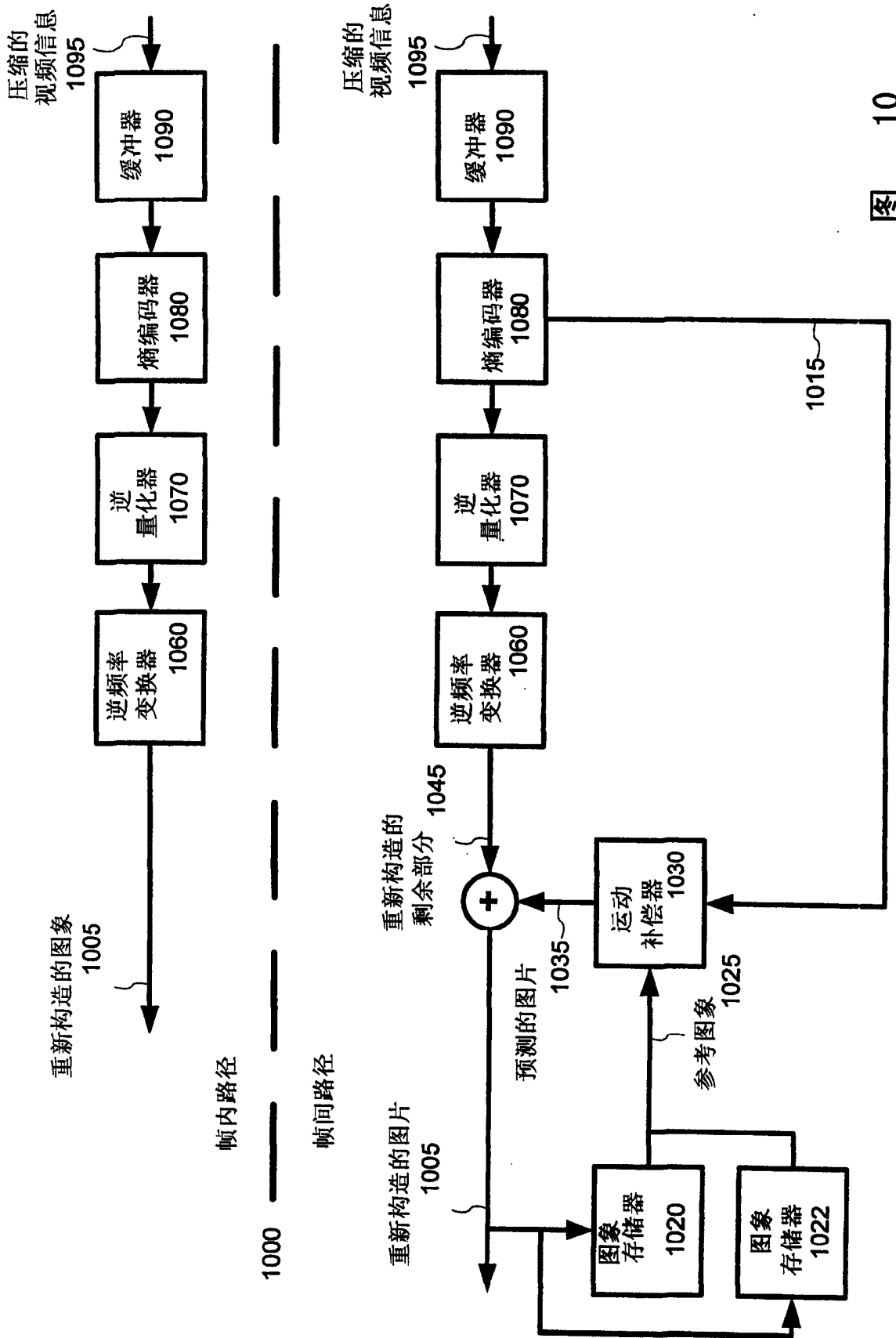


图 10

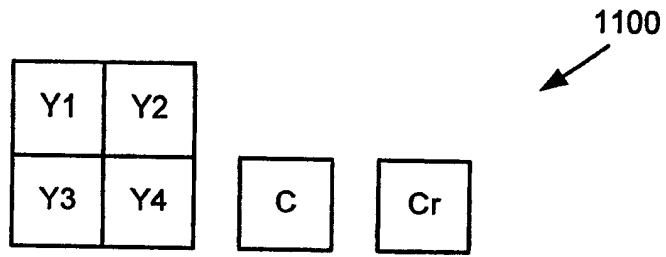


图 11

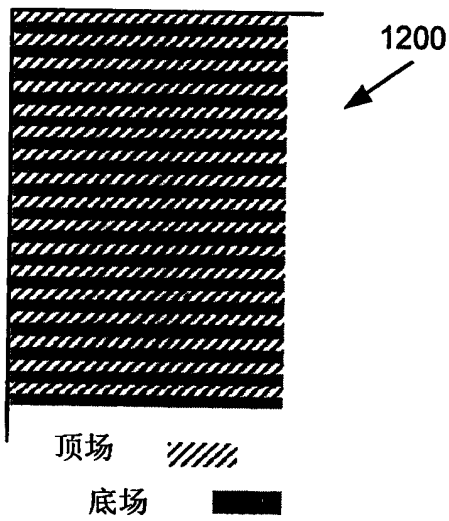


图 12A

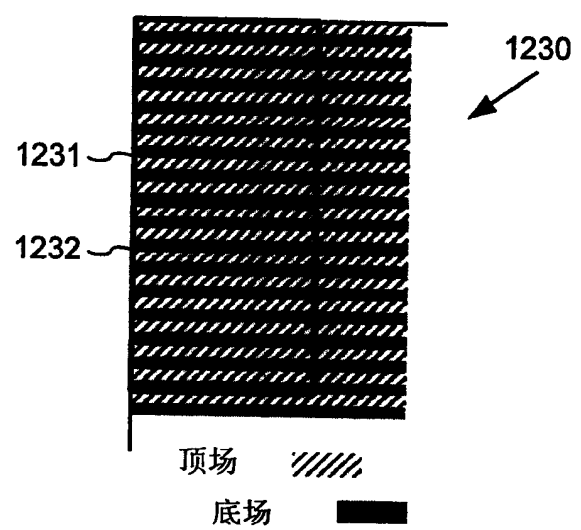


图 12B

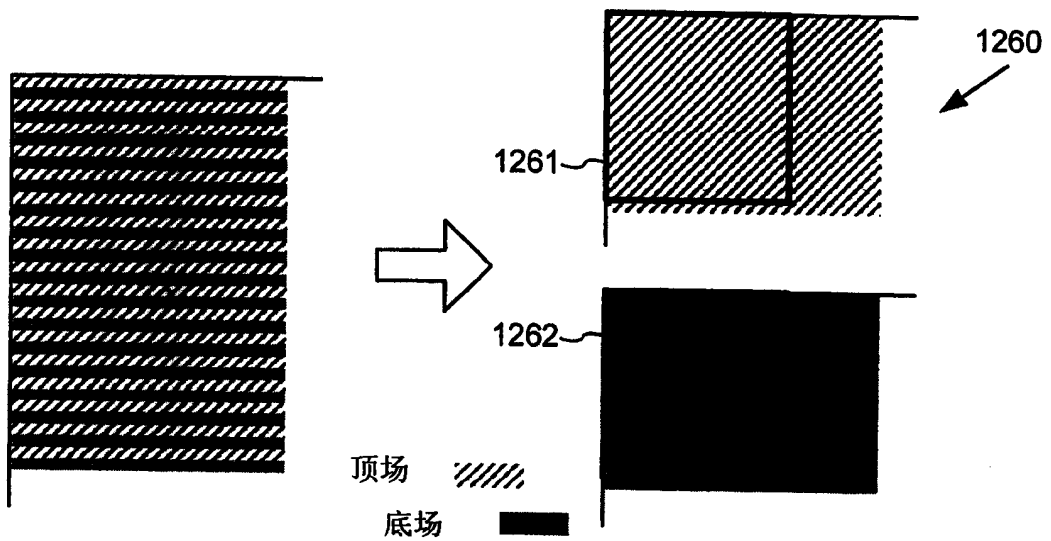


图 12C

```
Median3 (a, b, c) {  
    if (a > b) {  
        if (b > c)  
            median = b  
        else if (a > c)  
            median = c  
        else  
            median = a  
    }  
    else if (a > c)  
        median = a  
    else if (b > c)  
        median = c  
    else median = b  
    return median  
}
```

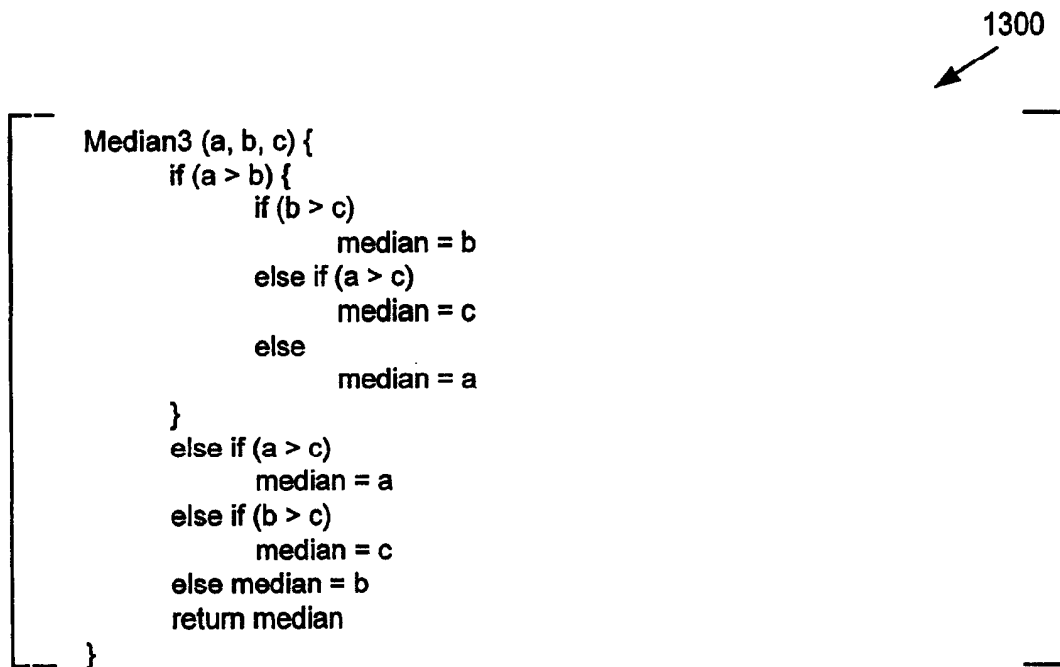


图 13

```
Median4 (a, b, c, d) {  
    max = min = a  
    if (b > max)  
        max = b  
    else if (b < min)  
        min = b  
    if (c > max)  
        max = c  
    else if (c < min)  
        min = c  
    if (d > max)  
        max = d  
    else if (d < min)  
        min = d  
  
    median = (a + b + c + d - max - min) / 2  
    return median  
}
```

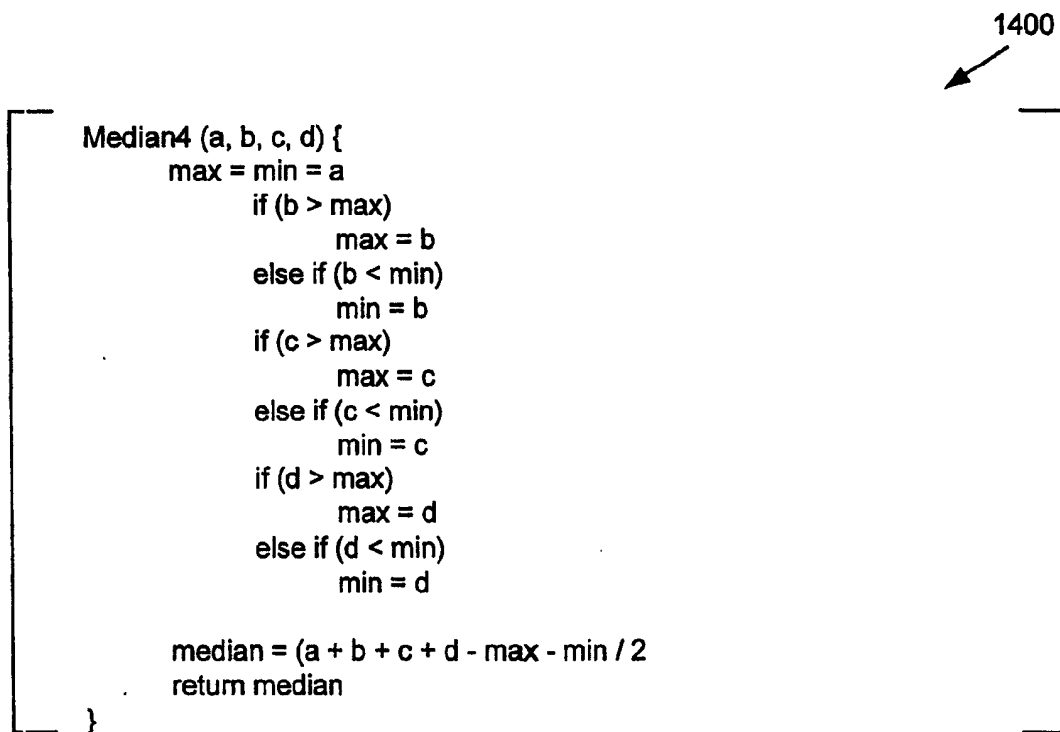


图 14

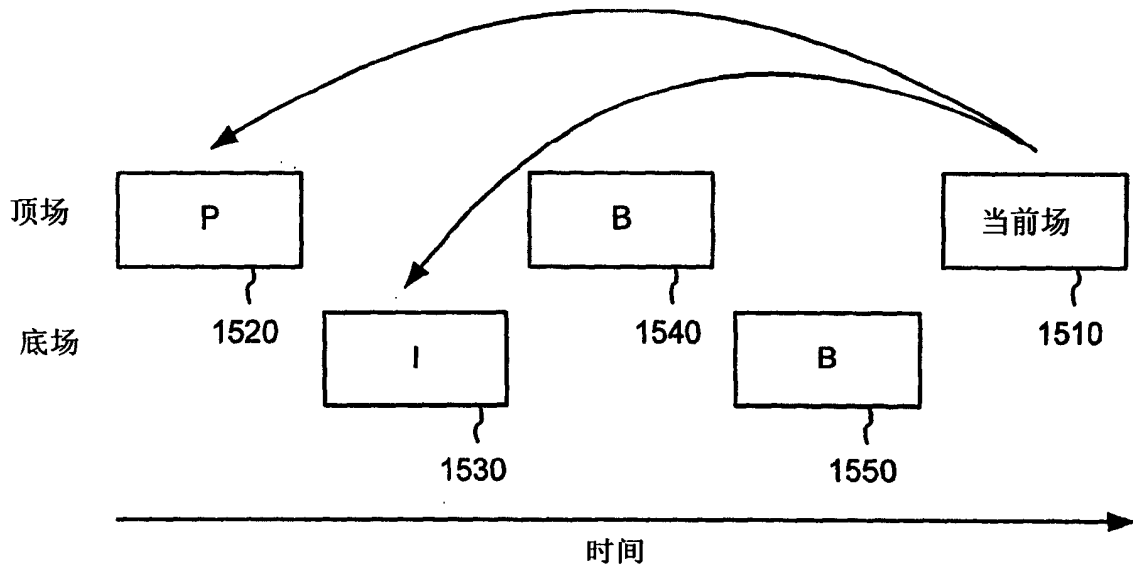


图 15

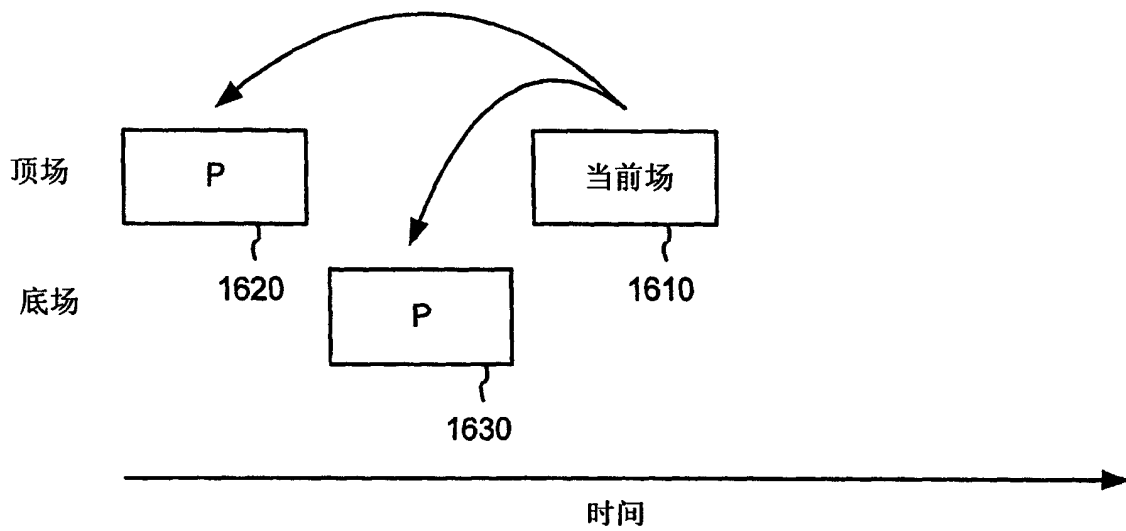


图 16

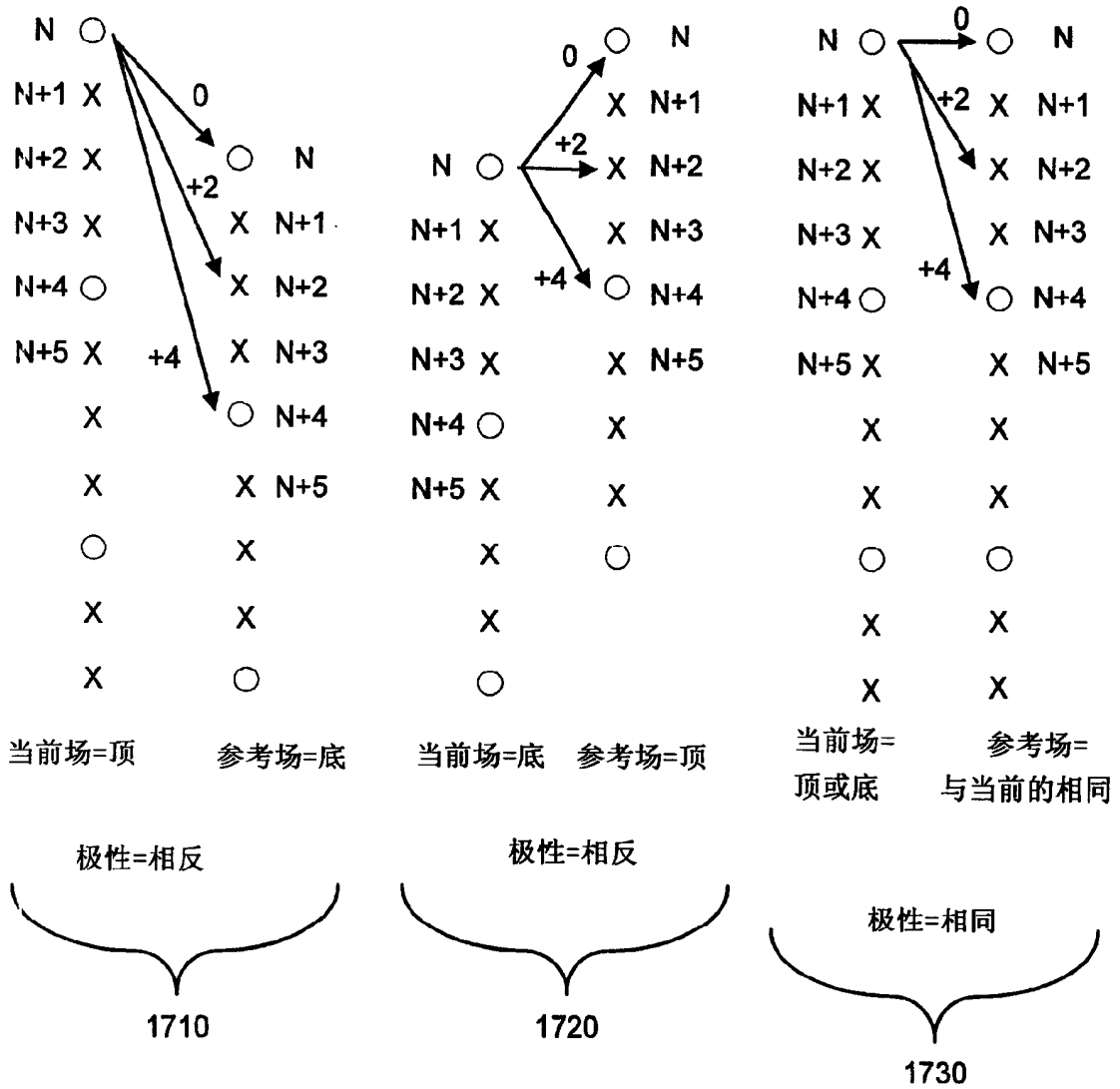


图 17

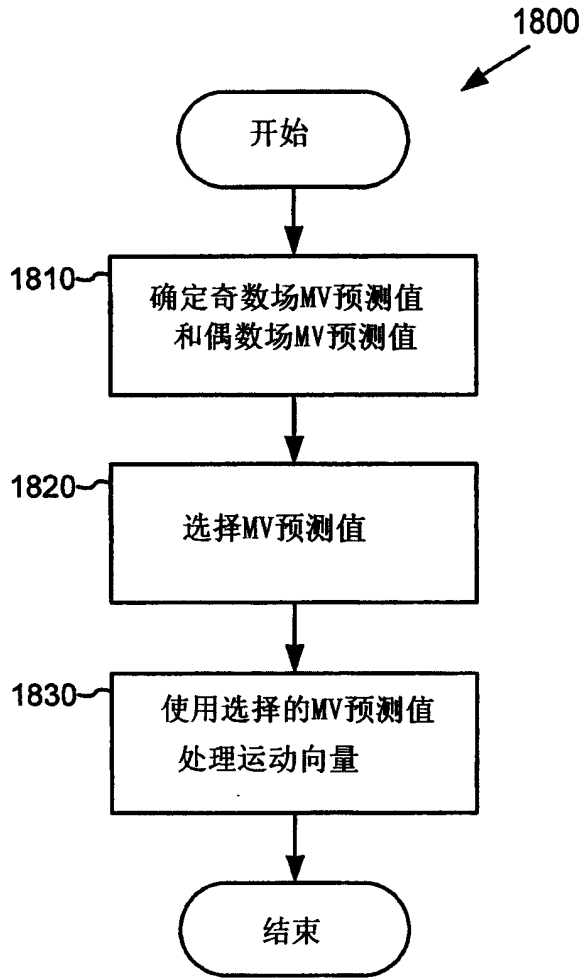


图 18

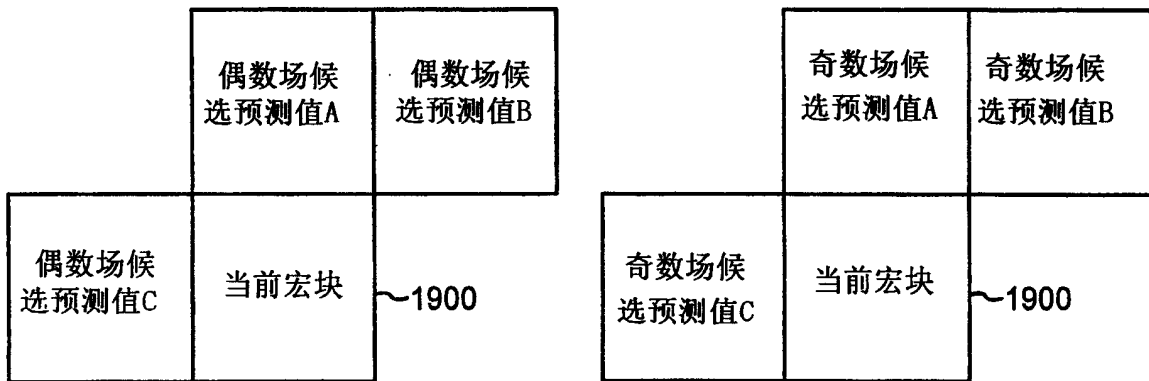


图 19

2000
↙

```
if (predictorA is not out of bounds) {
  if (predictorC is not out of bounds) {
    if (predictorA is intra) {
      predictorA_x = 0
      predictorA_y = 0
    }
    if (predictorB is intra) {
      predictorB_x = 0
      predictorB_y = 0
    }
    if (predictorC is intra) {
      predictorC_x = 0
      predictorC_y = 0
    }
    if (predictorA is from same field) {
      samecount = samecount + 1
      samefieldpredA_x = predictorA_x
      samefieldpredA_y = predictorA_y
      oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
      oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
    }
    else {
      oppositecount = oppositecount + 1
      oppositefieldpredA_x = predictorA_x
      oppositefieldpredA_y = predictorA_y
      samefieldpredA_x = scaleforsame_x(predictorA_x)
      samefieldpredA_y = scaleforsame_y(predictorA_y)
    }
    if (predictorB is from same field) {
      samecount = samecount + 1
      samefieldpredB_x = predictorB_x
      samefieldpredB_y = predictorB_y
      oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
      oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
    }
    else {
      oppositecount = oppositecount + 1
      oppositefieldpredB_x = predictorB_x
      oppositefieldpredB_y = predictorB_y
      samefieldpredB_x = scaleforsame_x(predictorB_x)
      samefieldpredB_y = scaleforsame_y(predictorB_y)
    }
  }
}
```

图 20A

2000
↙

```
if (predictorC is from same field) {
    samecount = samecount + 1
    samefieldpredC_x = predictorC_x
    samefieldpredC_y = predictorC_y
    oppositefieldpredC_x = scaleforopposite_x(predictorC_x)
    oppositefieldpredC_y = scaleforopposite_y(predictorC_y)
}
else {
    oppositecount = oppositecount + 1
    oppositefieldpredC_x = predictorC_x
    oppositefieldpredC_y = predictorC_y
    samefieldpredC_x = scaleforsame_x(predictorC_x)
    samefieldpredC_y = scaleforsame_y(predictorC_y)
}
samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, samefieldpredC_x)
samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, samefieldpredC_y)
oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, oppositefieldpredC_x)
oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, oppositefieldpredC_y)

if (samecount > oppositecount)
    dominantpredictor = samefield
else
    dominantpredictor = oppositefield
}
```

图 20B

2000
↙

```
else {
    // predictorC is out of bounds
    if (only 1 macroblock per row) {
        if (predictorA is intra) {
            samefieldpred_x = oppositefieldpred_x = 0
            samefieldpred_y = oppositefieldpred_y = 0
            dominantpredictor = oppositefield
        }
        else {
            // Use predictorA
            if (predictorA is from same field) {
                samefieldpred_x = predictorA_x
                samefieldpred_y = predictorA_y
                oppositefieldpred_x = scaleforopposite_x(predictorA_x)
                oppositefieldpred_y = scaleforopposite_y(predictorA_y)
                dominantpredictor = samefield
            }
            else {
                oppositefieldpred_x = predictorA_x
                oppositefieldpred_y = predictorA_y
                samefieldpred_x = scaleforsame_x(predictorA_x)
                samefieldpred_y = scaleforsame_y(predictorA_y)
                dominantpredictor = oppositefield
            }
        }
    }
}
```

图 20C

2000
↙

```
else {
    // PredictorC is out of bounds, use PredictorA and PredictorB
    predictorC_x = 0
    predictorC_y = 0
    if (predictorA is intra) {
        predictorA_x = 0
        predictorA_y = 0
    }
    if (predictorB is intra) {
        predictorB_x = 0
        predictorB_y = 0
    }
    if (predictorC is intra) {
        predictorC_x = 0
        predictorC_y = 0
    }
    if (predictorA is from same field) {
        samecount = samecount + 1
        samefieldpredA_x = predictorA_x
        samefieldpredA_y = predictorA_y
        oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
        oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
    }
    else {
        oppositecount = oppositecount + 1
        oppositefieldpredA_x = predictorA_x
        oppositefieldpredA_y = predictorA_y
        samefieldpredA_x = scaleforsame_x(predictorA_x)
        samefieldpredA_y = scaleforsame_y(predictorA_y)
    }
}
```

图 20D

2000
↙

```
if (predictorB is from same field) {
    samecount = samecount + 1
    samefieldpredB_x = predictorB_x
    samefieldpredB_y = predictorB_y
    oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
    oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
}
else {
    oppositecount = oppositecount + 1
    oppositefieldpredB_x = predictorB_x
    oppositefieldpredB_y = predictorB_y
    samefieldpredB_x = scaleforsame_x(predictorB_x)
    samefieldpredB_y = scaleforsame_y(predictorB_y)
}
samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, 0)
samefieldpred_y =
    median (samefieldpredA_y, samefieldpredA_y, 0)
oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, 0)
oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, 0)
if (samecount > oppositecount)
    dominantpredictor = samefield
else
    dominantpredictor = oppositefield
}
```

图 20E

2000
↙

```
else {  
    // Predictor A is out of bounds  
    if (predictorC is out of bounds) {  
        samefieldpred_x = oppositefieldpred_x = 0  
        samefieldpred_y = oppositefieldpred_y = 0  
        dominantpredictor = oppositefield  
    }  
    else {  
        // Use predictorC  
        if (predictorC is from same field) {  
            samefieldpred_x = predictorC_x  
            samefieldpred_y = predictorC_y  
            oppositefieldpred_x = scaleforopposite_x(predictorC_x)  
            oppositefieldpred_y = scaleforopposite_y(predictorC_y)  
            dominantpredictor = samefield  
        }  
        else {  
            oppositefieldpred_x = predictorC_x  
            oppositefieldpred_y = predictorC_y  
            samefieldpred_x = scaleforsame_x(predictorC_x)  
            samefieldpred_y = scaleforsame_y(predictorC_y)  
            dominantpredictor = oppositefield  
        }  
    }  
}
```

图 20F

```
scaleforopposite_x(n) {  
    int scaledvalue  
    scaledvalue = (n * SCALEOPP) >> 8  
    return scaledvalue  
}  
  
scaleforopposite_y(n) {  
    int scaledvalue  
    if (current field is top)  
        scaledvalue = ((n * SCALEOPP) >> 8) - 2  
    else //current field is bottom  
        scaledvalue = ((n * SCALEOPP) >> 8) + 2  
    return scaledvalue  
}
```

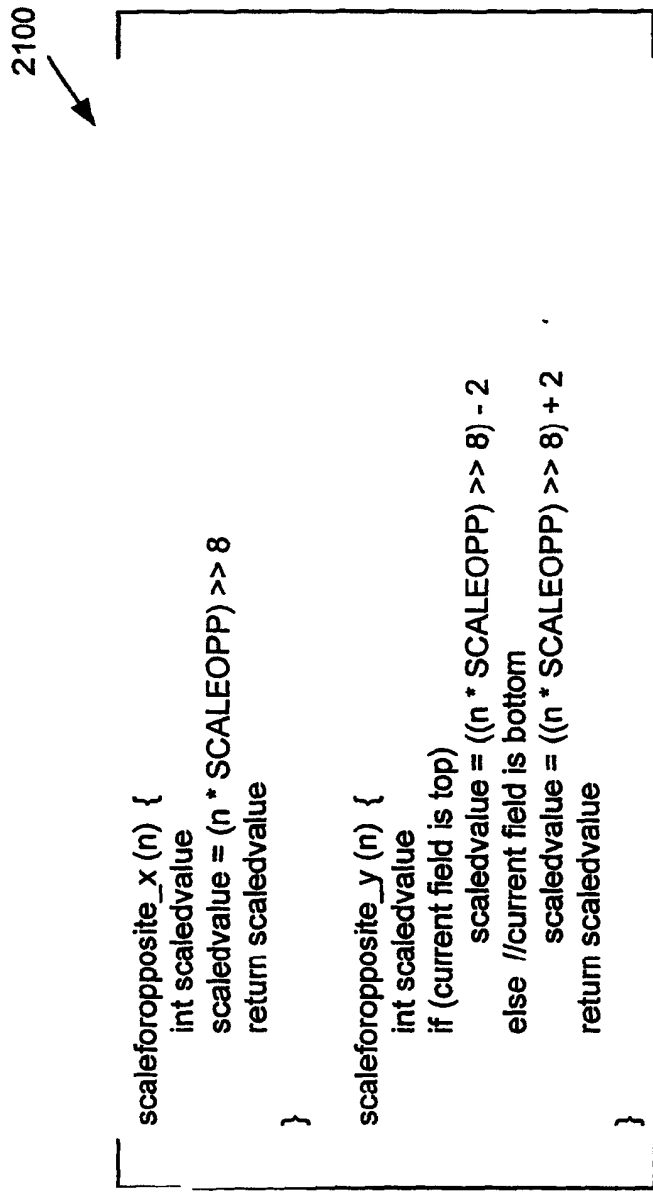


图 21A

2100
↙

```
scaleforsame_x (n) {
  if (abs (n) < SCALEZONE1_X)
    scaledvalue = (n * SCALESAME1) >> 8
  else {
    if (n < 0)
      scaledvalue = ((n * SCALESAME2) >> 8) - ZONE1OFFSET_X
    else
      scaledvalue = ((n * SCALESAME2) >> 8) + ZONE1OFFSET_X
  }
  return scaledvalue
}

scaleforsame_y (n) {
  if (current field is top) {
    if (abs (n) < SCALEZONE1_Y)
      scaledvalue = ((n + 2) * SCALESAME1) >> 8
    else {
      if (n < 0)
        scaledvalue = (((n + 2) * SCALESAME2) >> 8) - ZONE1OFFSET_Y
      else
        scaledvalue = (((n + 2) * SCALESAME2) >> 8) + ZONE1OFFSET_Y
    }
  }
  else { //current field is bottom
    if (abs (n) < SCALEZONE1_Y)
      scaledvalue = ((n - 2) * SCALESAME1) >> 8
    else {
      if (n < 0)
        scaledvalue = (((n - 2) * SCALESAME2) >> 8) - ZONE1OFFSET_Y
      else
        scaledvalue = (((n - 2) * SCALESAME2) >> 8) + ZONE1OFFSET_Y
    }
  }
  return scaledvalue
}
```

图 21B

2200
↙

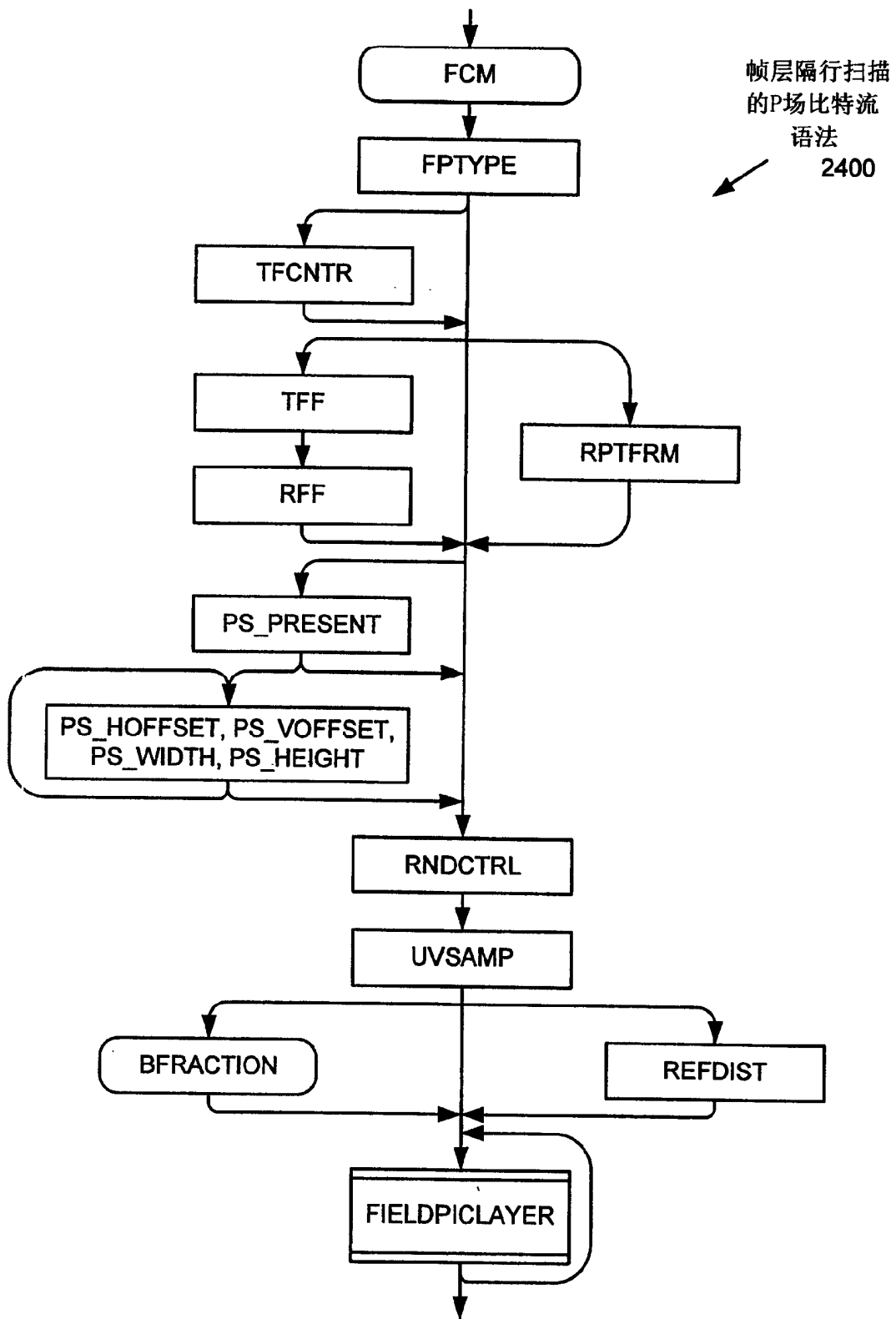
	参考帧距离			
	1	2	3	4或更大
SCALEOPP	128	192	213	224
SCALESAME1	512	341	307	293
SCALESAME2	219	236	242	245
SCALEZONE1_X	32 * N	48 * N	53 * N	56 * N
SCALEZONE1_Y	8 * N	12 * N	13 * N	14 * N
ZONE1OFFSET_X	37 * N	20 * N	14 * N	11 * N
ZONE1OFFSET_Y	10 * N	5 * N	4 * N	3 * N

图 22

2300
↙

	参考帧距离			
	1	2	3	4或更大
SCALEOPP	128	64	43	32
SCALESAME1	512	1024	1536	2048
SCALESAME2	219	204	200	198
SCALEZONE1_X	32 * N	16 * N	11 * N	8 * N
SCALEZONE1_Y	8 * N	4 * N	3 * N	2 * N
ZONE1OFFSET_X	37 * N	52 * N	56 * N	11 * N
ZONE1OFFSET_Y	10 * N	5 * N	4 * N	3 * N

图 23



帧层隔行扫描
的P场比特流
语法
2400

图 24

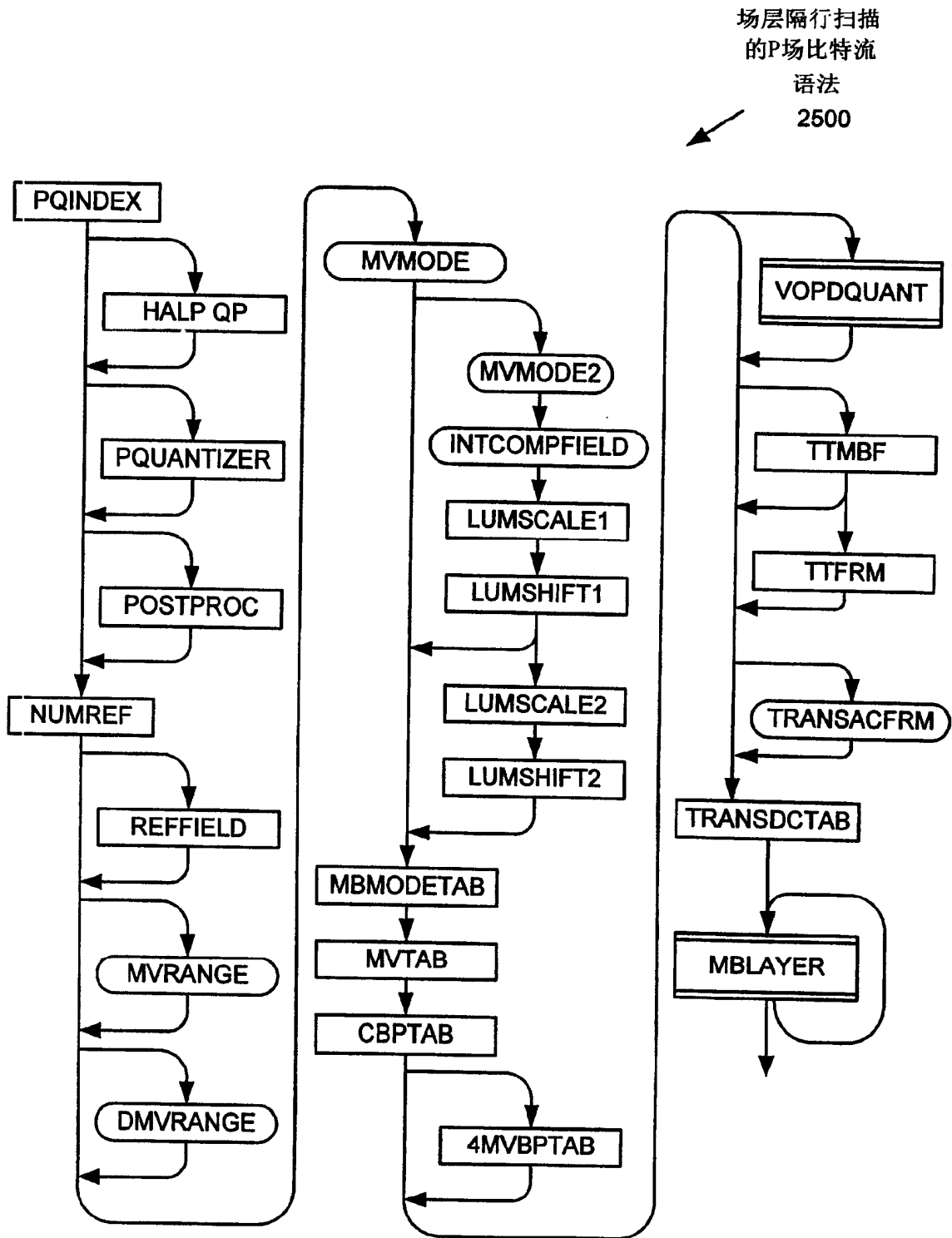


图 25

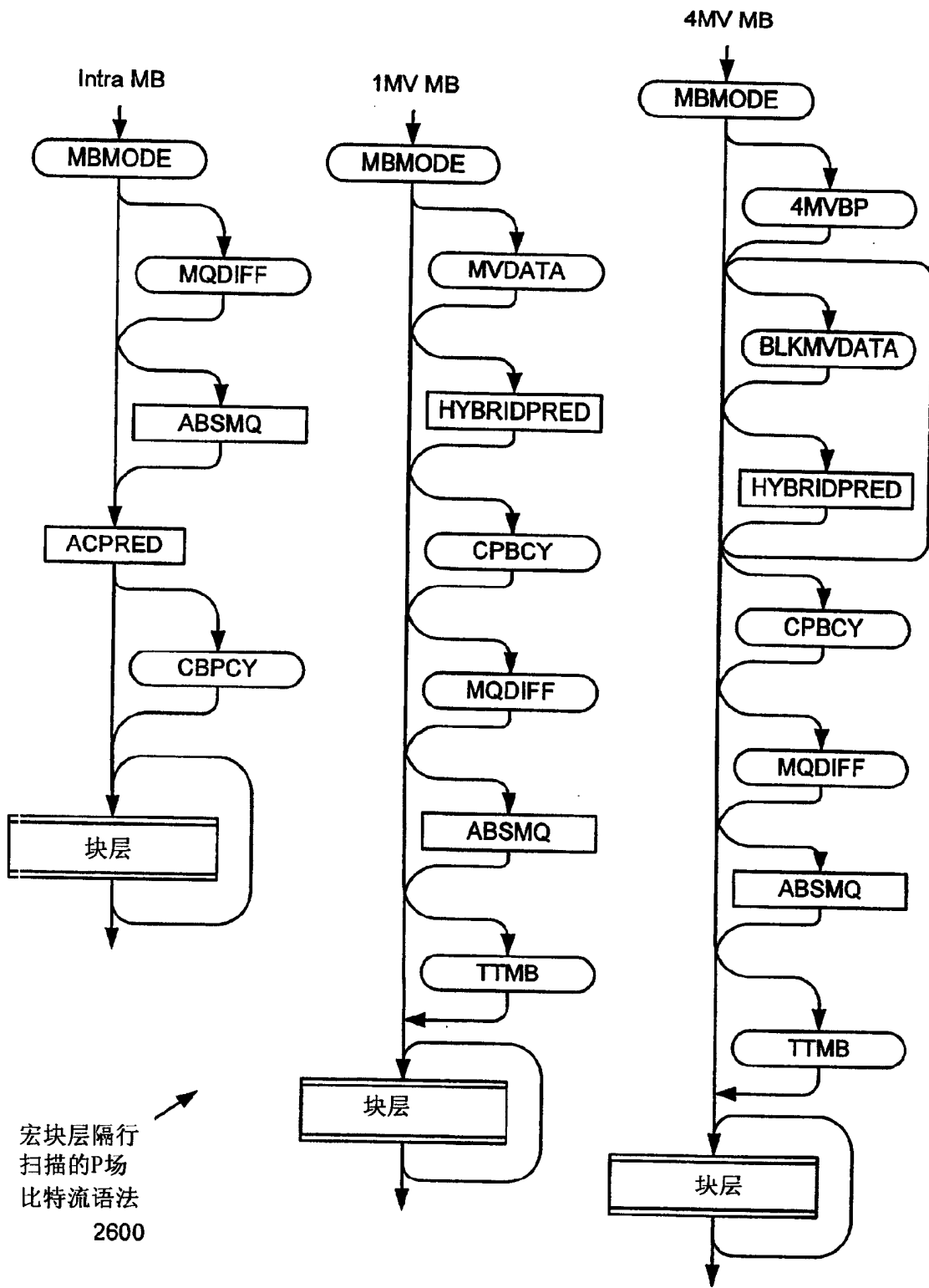


图 26

2700
↙

```

offset_table1[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128,}
offset_table2[9] = {0, 1, 3, 7, 15, 31, 63, 127, 255}
index = vlc_decode() // Use the Huffman table indicated by MVTAB in
the picture layer
if (index == 71)
{
    dmv_x = get_bits(k_x)
    dmv_y = get_bits(k_y)
}
else
{
    if (extend_x == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) % 9
    if (index1 != 0)
    {
        val = get_bits (index1 + extend_x)
        sign = 0 - (val & 1)
        dmv_x = sign ^ ((val >> 1) + offset_table[index1])
        dmv_x = dmv_x - sign
    }
    else
        dmv_x = 0
    if (extend_y == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) / 9
    if (index1 != 0)
    {
        val = get_bits (index1 + extend_y)
        sign = 0 - (val & 1)
        dmv_y = sign ^ ((val >> 1) + offset_table[index1])
        dmv_y = dmv_y - sign
    }
    else
        dmv_y = 0
}
}

```

图 27A

2710

```
offset_table[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128}
index = vlc_decode() // Use the Huffman table indicated by MVTAB in
the picture layer
if (index == 0) {
    dmv_x = 1 - 2 * get_bits(1)
    dmv_y = 0
}
if (index == 125)
{
    dmv_x = get_bits(k_x - halfpel_flag)
    dmv_y = get_bits(k_y - halfpel_flag)
}
else
{
    index1 = (index + 1) % 9
    val = get_bits (index1)
    sign = 0 - (val & 1)
    dmv_x = sign ^ ((val >> 1) + offset_table[index1])
    dmv_x = dmv_x - sign

    index1 = (index + 1) / 9
    val = get_bits (index1)
    sign = 0 - (val & 1)
    dmv_y = sign ^ ((val >> 1) + offset_table[index1])
    dmv_y = dmv_y - sign
}
```

图 27B

2800
↙

```

size_table[16] = {0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7}
offset_table1[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128}
offset_table2[9] = {0, 1, 3, 7, 15, 31, 63, 127, 255}
index = vlc_decode() // Use the Huffman table indicated by MVTAB
if (index == 125)
{
    dmv_x = get_bits(k_x)
    dmv_y = get_bits(k_y)
    predictor_flag = dmv_y & 1
    dmv_y = dmv_y >> 1
}
else
{
    if (extend_x == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) % 9
    if (index1 != 0)
    {
        val = get_bits (index1 + extend_x)
        sign = 0 - (val & 1)
        dmv_x = sign ^ ((val >> 1) + offset_table[index1])
        dmv_x = dmv_x - sign
    }
    else
        dmv_x = 0
    if (extend_y == 1)
        offset_table = offset_table2
    else
        offset_table = offset_table1
    index1 = (index + 1) / 9
    if (index1 != 0)
    {
        val = get_bits (size_table[index1 + 2 * extend_y])
        sign = 0 - (val & 1)
        dmv_y = sign ^ ((val >> 1) + offset_table[index1 >> 1])
        dmv_y = dmv_y - sign
        predictor_flag = index1 & 1
    }
    else
    {
        dmv_y = 0
        predictor_flag = 0
    }
}
}

```

图 28A

2810
↙

```

size_table[14] = {0, 0, 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6}
offset_table[9] = {0, 1, 2, 4, 8, 16, 32, 64, 128}
index = vlc_decode() // Use the Huffman table indicated by MVTAB in
the picture layer
if (index == 0) {
    dmv_x = 1 - 2 * get_bits(1)
    dmv_y = 0
    predictor_flag = 0
}
if (index == 125)
{
    dmv_x = get_bits(k_x - halfpel_flag)
    dmv_y = get_bits(k_y - halfpel_flag)
    predictor_flag = dmv_y & 1
    dmv_y = dmv_y >> 1
}
else
{
    index1 = (index + 1) % 9
    val = get_bits (index1)
    sign = 0 - (val & 1)
    dmv_x = sign ^ ((val >> 1) + offset_table[index1])
    dmv_x = dmv_x - sign

    index1 = (index + 1) / 9
    val = get_bits (size_table[index1])
    sign = 0 - (val & 1)
    dmv_y = sign ^ ((val >> 1) + offset_table[index1])
    dmv_y = dmv_y - sign
    predictor_flag = index1 & 1
}
}

```

图 28B

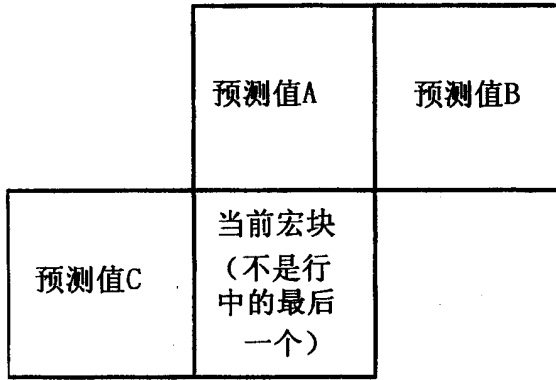


图 29A

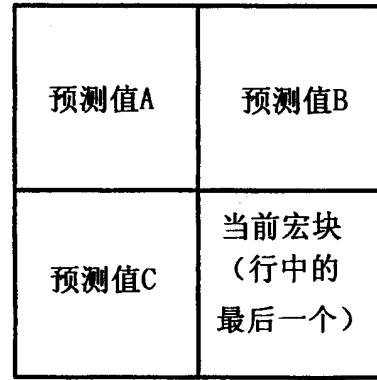


图 29B

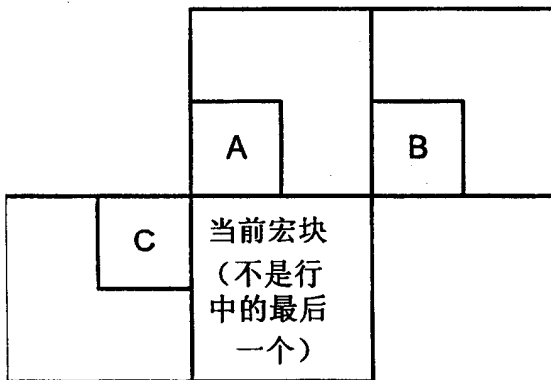


图 30A

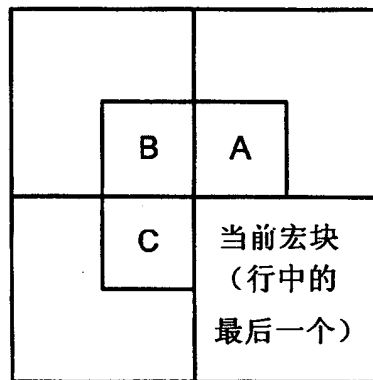


图 30B

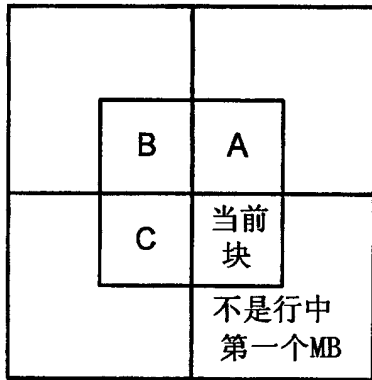


图 31A

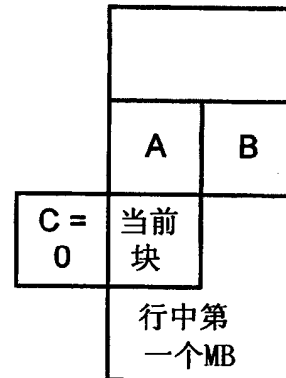


图 31B

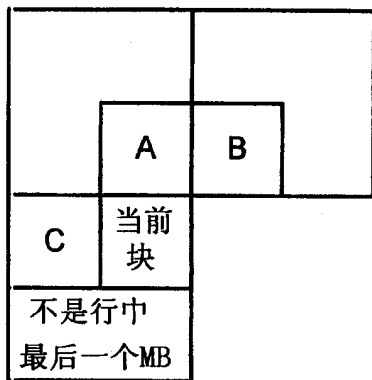


图 32A

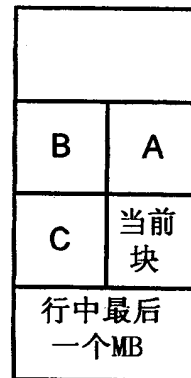


图 32B

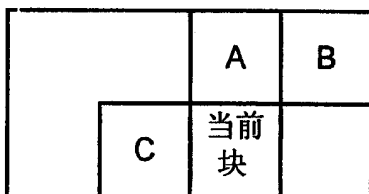


图 33

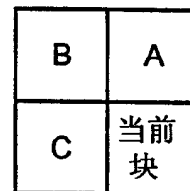


图 34

3500

```

samecount = 0;
oppositecount = 0;
if (predictorA is not out of bounds) {
  if (predictorC is not out of bounds) {
    if (predictorA is intra) {
      samefieldpred_x = oppositefieldpred_x = samefieldpredA_x = oppositefieldpredA_x = 0
      samefieldpred_y = oppositefieldpred_y = samefieldpredA_y = oppositefieldpredA_y = 0
    }
    if (predictorB is intra) {
      samefieldpred_x = oppositefieldpred_x = samefieldpredB_x = oppositefieldpredB_x = 0
      samefieldpred_y = oppositefieldpred_y = samefieldpredB_y = oppositefieldpredB_y = 0
    }
    if (predictorC is intra) {
      samefieldpred_x = oppositefieldpred_x = samefieldpredC_x = oppositefieldpredC_x = 0
      samefieldpred_y = oppositefieldpred_y = samefieldpredC_y = oppositefieldpredC_y = 0
    }
  }
  if (predictorA is not intra) {
    if (predictorA is from same field) {
      samecount = samecount + 1
      samefieldpred_x = samefieldpredA_x = predictorA_x
      samefieldpred_y = samefieldpredA_y = predictorA_y
      oppositefieldpred_x = oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
      oppositefieldpred_y = oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
    }
    else {
      oppositecount = oppositecount + 1
      oppositefieldpred_x = oppositefieldpredA_x = predictorA_x
      oppositefieldpred_y = oppositefieldpredA_y = predictorA_y
      samefieldpred_x = samefieldpredA_x = scaleforsame_x(predictorA_x)
      samefieldpred_y = samefieldpredA_y = scaleforsame_y(predictorA_y)
    }
  }
}
if (predictorB is not intra) {
  if (predictorB is from same field) {
    samecount = samecount + 1
    samefieldpred_x = samefieldpredB_x = predictorB_x
    samefieldpred_y = samefieldpredB_y = predictorB_y
    oppositefieldpred_x = oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
    oppositefieldpred_y = oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpred_x = oppositefieldpredB_x = predictorB_x
    oppositefieldpred_y = oppositefieldpredB_y = predictorB_y
    samefieldpred_x = samefieldpredB_x = scaleforsame_x(predictorB_x)
    samefieldpred_y = samefieldpredB_y = scaleforsame_y(predictorB_y)
  }
}
}

```

图 35A

3500
↙

```
if (predictorC is not intra) {
  if (predictorC is from same field) {
    samecount = samecount + 1
    samefieldpred_x = samefieldpredC_x = predictorC_x
    samefieldpred_y = samefieldpredC_y = predictorC_y
    oppositefieldpred_x = oppositefieldpredC_x = scaleforopposite_x(predictorC_x)
    oppositefieldpred_y = oppositefieldpredC_y = scaleforopposite_y(predictorC_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpred_x = oppositefieldpredC_x = predictorC_x
    oppositefieldpred_y = oppositefieldpredC_y = predictorC_y
    samefieldpred_x = samefieldpredC_x = scaleforsame_x(predictorC_x)
    samefieldpred_y = samefieldpredC_y = scaleforsame_y(predictorC_y)
  }
}
if ((samecount + oppositecount) > 1) {
  samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, samefieldpredC_x)
  samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, samefieldpredC_y)
  oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, oppositefieldpredC_x)
  oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, oppositefieldpredC_y)
}
if (samecount > oppositecount)
  dominantpredictor = samefield
else
  dominantpredictor = oppositefield
}
```

图 35B

3500
↙

```
else {
  // predictorC is out of bounds
  if (only 1 macroblock per row) {
    if (predictorA is intra) {
      samefieldpred_x = oppositefieldpred_x = 0
      samefieldpred_y = oppositefieldpred_y = 0
      dominantpredictor = oppositefield
    }
    else {
      // Use predictorA
      if (predictorA is from same field) {
        samefieldpred_x = predictorA_x
        samefieldpred_y = predictorA_y
        oppositefieldpred_x = scaleforopposite_x(predictorA_x)
        oppositefieldpred_y = scaleforopposite_y(predictorA_y)
        dominantpredictor = samefield
      }
      else {
        oppositefieldpred_x = predictorA_x
        oppositefieldpred_y = predictorA_y
        samefieldpred_x = scaleforsame_x(predictorA_x)
        samefieldpred_y = scaleforsame_y(predictorA_y)
        dominantpredictor = oppositefield
      }
    }
  }
}
```

图 35C

3500
↙

```

else {
// Predictor C is out of bounds, use Predictor and PredictorB
predictorC_x = 0
predictorC_y = 0
if (predictorA is intra) {
samefieldpred_x = oppositefieldpred_x = samefieldpredA_x = oppositefieldpredA_x = 0
samefieldpred_y = oppositefieldpred_y = samefieldpredA_y = oppositefieldpredA_y = 0
}
if (predictorB is intra) {
samefieldpred_x = oppositefieldpred_x = samefieldpredB_x = oppositefieldpredB_x = 0
samefieldpred_y = oppositefieldpred_y = samefieldpredB_y = oppositefieldpredB_y = 0
}
if (predictorC is intra) {
samefieldpred_x = oppositefieldpred_x = samefieldpredC_x = oppositefieldpredC_x = 0
samefieldpred_y = oppositefieldpred_y = samefieldpredC_y = oppositefieldpredC_y = 0
}
if (predictorA is not intra) {
if (predictorA is from same field) {
samecount = samecount + 1
samefieldpred_x = samefieldpredA_x = predictorA_x
samefieldpred_y = samefieldpredA_y = predictorA_y
oppositefieldpred_x = oppositefieldpredA_x = scaleforopposite_x(predictorA_x)
oppositefieldpred_y = oppositefieldpredA_y = scaleforopposite_y(predictorA_y)
}
else {
oppositecount = oppositecount + 1
oppositefieldpred_x = oppositefieldpredA_x = predictorA_x
oppositefieldpred_y = oppositefieldpredA_y = predictorA_y
samefieldpred_x = samefieldpredA_x = scaleforsame_x(predictorA_x)
samefieldpred_y = samefieldpredA_y = scaleforsame_y(predictorA_y)
}
}
}

```

图 35D

3500
↙

```
if (predictorB is not intra) {
  if (predictorB is from same field) {
    samecount = samecount + 1
    samefieldpred_x = samefieldpredB_x = predictorB_x
    samefieldpred_y = samefieldpredB_y = predictorB_y
    oppositefieldpred_x = oppositefieldpredB_x = scaleforopposite_x(predictorB_x)
    oppositefieldpred_y = oppositefieldpredB_y = scaleforopposite_y(predictorB_y)
  }
  else {
    oppositecount = oppositecount + 1
    oppositefieldpred_x = oppositefieldpredB_x = predictorB_x
    oppositefieldpred_y = oppositefieldpredB_y = predictorB_y
    samefieldpred_x = samefieldpredB_x = scaleforsame_x(predictorB_x)
    samefieldpred_y = samefieldpredB_y = scaleforsame_y(predictorB_y)
  }
}
if ((samecount + oppositecount) > 1) {
  samefieldpred_x =
    median (samefieldpredA_x, samefieldpredB_x, samefieldpredC_x)
  samefieldpred_y =
    median (samefieldpredA_y, samefieldpredB_y, samefieldpredC_y)
  oppositefieldpred_x =
    median (oppositefieldpredA_x, oppositefieldpredB_x, oppositefieldpredC_x)
  oppositefieldpred_y =
    median (oppositefieldpredA_y, oppositefieldpredB_y, oppositefieldpredC_y)
}
if (samecount > oppositecount)
  dominantpredictor = samefield
else
  dominantpredictor = oppositelfield
}
}
```

图 35E

3500
↙

```
else {  
    // Predictor A is out of bounds  
    if (predictorC is out of bounds) {  
        samefieldpred_x = oppositefieldpred_x = 0  
        samefieldpred_y = oppositefieldpred_y = 0  
        dominantpredictor = oppositefield  
    }  
    else {  
        // Use predictorC  
        if (predictorC is from same field) {  
            samefieldpred_x = predictorC_x  
            samefieldpred_y = predictorC_y  
            oppositefieldpred_x = scaleforopposite_x(predictorC_x)  
            oppositefieldpred_y = scaleforopposite_y(predictorC_y)  
            dominantpredictor = samefield  
        }  
        else {  
            oppositefieldpred_x = predictorC_x  
            oppositefieldpred_y = predictorC_y  
            samefieldpred_x = scaleforsame_x(predictorC_x)  
            samefieldpred_y = scaleforsame_y(predictorC_y)  
            dominantpredictor = oppositefield  
        }  
    }  
}
```

图 35F

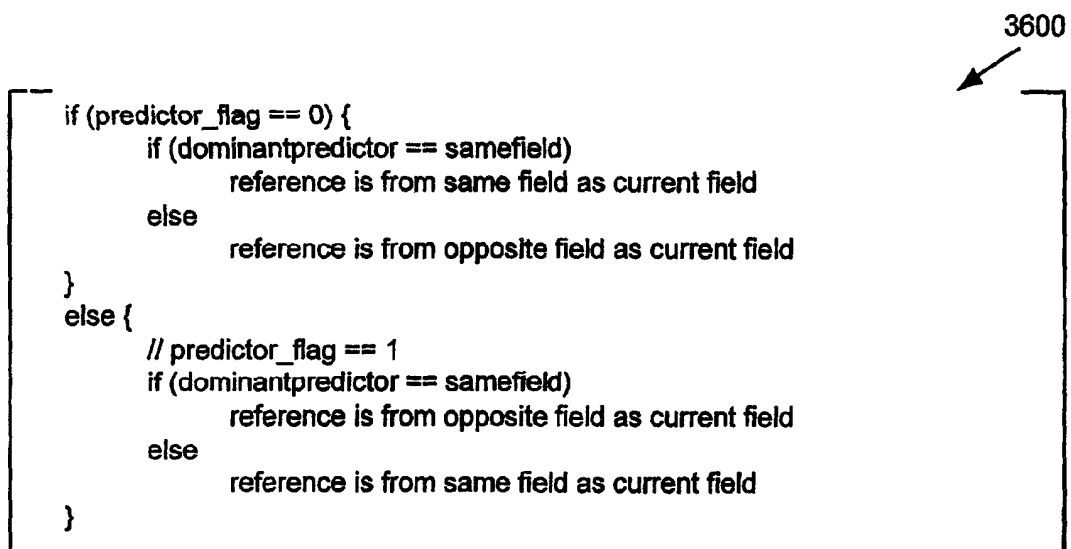


图 36