



## (12) 发明专利

(10) 授权公告号 CN 113225560 B

(45) 授权公告日 2023.05.23

(21) 申请号 202110443353.9

(74) 专利代理机构 北京派特恩知识产权代理有

(22) 申请日 2019.09.20

限公司 11270

(65) 同一申请的已公布的文献号

专利代理人 吴薇薇 张颖玲

申请公布号 CN 113225560 A

(51) Int.CI.

H04N 19/159 (2014.01)

(43) 申请公布日 2021.08.06

H04N 19/176 (2014.01)

(30) 优先权数据

H04N 19/94 (2014.01)

10-2018-0114342 2018.09.21 KR

H04N 19/96 (2014.01)

10-2018-0114343 2018.09.21 KR

10-2018-0114344 2018.09.21 KR

(62) 分案原申请数据

201980054695.2 2019.09.20

(56) 对比文件

(73) 专利权人 OPPO广东移动通信有限公司

US 2018192069 A1, 2018.07.05

地址 523860 广东省东莞市长安镇乌沙海  
滨路18号

CN 108432250 A, 2018.08.21

(72) 发明人 李培根

WO 2018155983 A1, 2018.08.30

Minhua Zhou 等.Non-CE4: A study on the affine merge mode.《Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11》.2018, 第1–10页.

审查员 王瑞

权利要求书6页 说明书48页 附图24页

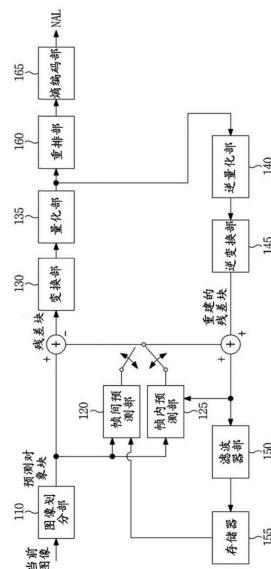
(54) 发明名称

视频编码/解码方法、视频编码/解码装置和存储介质

(57) 摘要

本发明涉及视频解码方法、视频编码方法、视频解码装置、视频编码装置和存储介质。本发明的视频解码方法包括以下步骤：在当前块中生成合并候选列表；指定所述合并候选列表中所包含的多个合并候选中的任一个；基于所述指定的合并候选的第一仿射种子矢量和第二仿射种子矢量导出所述当前块的第一仿射种子矢量和第二仿射种子矢量；利用所述当前块的所述第一仿射种子矢量和所述第二仿射种子矢量导出所述当前块的子块的仿射矢量；以及基于所述仿射矢量对所述子块执行运动补偿预测。

CN 113225560 B



1. 一种视频解码方法,包括:

生成当前块的合并候选列表;指定所述合并候选列表中包括的多个合并候选之一;

基于所指定的合并候选的第一仿射种子矢量和第二仿射种子矢量,导出所述当前块的第一仿射种子矢量和第二仿射种子矢量;

通过使用所述当前块的所述第一仿射种子矢量和所述第二仿射种子矢量导出所述当前块中的子块的仿射矢量,其中,所述子块是大小小于所述当前块的大小的区域;以及

基于所述仿射矢量对所述子块执行运动补偿预测,其中,

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于与所述当前块相邻的相邻块的运动信息导出的;

当所述相邻块被包括在与所述当前块的编码树单元不同的编码树单元中时,所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的左下子块和右下子块的运动矢量导出的。

2. 根据权利要求1所述的视频解码方法,其中,

所述左下子块包括位于所述相邻块的左下角的左下参考样本,并且所述右下子块包括位于所述相邻块的右下角的右下参考样本。

3. 根据权利要求2所述的视频解码方法,其中,

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的,

并且所述比例因子是基于通过左下参考样本与右下参考样本之间的水平距离与偏移量相加获得的值导出的。

4. 根据权利要求2所述的视频解码方法,其中,

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的,并且所述比例因子是基于所述右下参考样本右侧相邻的相邻样本与所述左下参考样本之间的距离导出的。

5. 根据权利要求1所述的视频解码方法,其中,

所述合并候选列表包括第一合并候选和第二合并候选,所述第一合并候选是基于被确定为位于所述当前块上方的上相邻块中的第一可用块导出的,并且所述第二合并候选是基于被确定为位于所述当前块左侧的左相邻块中的第一可用块导出的。

6. 根据权利要求1所述的视频解码方法,其中,

当所述相邻块被包括在与所述当前块的编码树单元相同的编码树单元中时,

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的第一仿射种子矢量和第二仿射种子矢量导出的。

7. 根据权利要求1所述的视频解码方法,其中,使用与仿射相邻块的左下控制点相关的第三仿射种子矢量和与仿射相邻块的右下控制点相关的第四仿射种子矢量来导出当前块的第一仿射种子矢量和第二仿射种子矢量,

$$v_{0x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x}$$

$$v_{0y} = \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(x_0-x_{n2}) - \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(y_0-y_{n2}) + nv_{2y}$$

$$v_{1x} = \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(x_1-x_{n2}) - \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(y_1-y_{n2}) + nv_{2x}$$

$$v_{1y} = \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(x_1-x_{n2}) - \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(y_1-y_{n2}) + nv_{2y}$$

$(x_{n2}, y_{n2})$  表示仿射相邻块的左下控制点的坐标，并且  $(x_{n3}, y_{n3})$  表示仿射相邻块的右下控制点的坐标；

$(x_0, y_0)$  表示当前块的左上控制点的坐标，并且  $(x_1, y_1)$  表示当前块的右上控制点的坐标；

$(nv_{2x}, nv_{2y})$  表示仿射相邻块的左下控制点的第三仿射种子矢量，并且  $(nv_{3x}, nv_{3y})$  表示仿射相邻块的右下控制点的第四仿射种子矢量；

$(v_{0x}, v_{0y})$  表示当前块的左上控制点的第一仿射种子矢量，并且  $(v_{1x}, v_{1y})$  表示当前块的右上控制点的第二仿射种子矢量。

#### 8. 一种视频编码方法，包括：

生成当前块的合并候选列表；指定所述合并候选列表中包括的多个合并候选之一；

基于所指定的合并候选的第一仿射种子矢量和第二仿射种子矢量，导出所述当前块的第一仿射种子矢量和第二仿射种子矢量；

通过使用所述当前块的所述第一仿射种子矢量和所述第二仿射种子矢量导出所述当前块中的子块的仿射矢量，其中，所述子块是大小小于所述当前块的大小的区域；以及

基于所述仿射矢量对所述子块执行运动补偿预测，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于与所述当前块相邻的相邻块的运动信息导出的；

当所述相邻块被包括在与所述当前块的编码树单元不同的编码树单元中时，所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的左下子块和右下子块的运动矢量导出的。

#### 9. 根据权利要求8所述的视频编码方法，其中，

所述左下子块包括位于所述相邻块的左下角的左下参考样本，并且所述右下子块包括位于所述相邻块的右下角的右下参考样本。

#### 10. 根据权利要求9所述的视频编码方法，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的，

并且所述比例因子是基于通过左下参考样本与右下参考样本之间的水平距离与偏移量相加获得的值导出的。

#### 11. 根据权利要求9所述的视频编码方法，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的，

并且所述比例因子是基于所述右下参考样本右侧相邻的相邻样本与所述左下参考样

本之间的距离导出的。

12. 根据权利要求8所述的视频编码方法,其中,

所述合并候选列表包括第一合并候选和第二合并候选,所述第一合并候选是基于被确定为位于所述当前块上方的上相邻块中的第一可用块导出的,并且所述第二合并候选是基于被确定为位于所述当前块左侧的左相邻块中的第一可用块导出的。

13. 根据权利要求8所述的视频编码方法,其中,

当所述相邻块被包括在与所述当前块的编码树单元相同的编码树单元中时,

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的第一仿射种子矢量和第二仿射种子矢量导出的。

14. 根据权利要求8所述的视频编码方法,其中,使用与仿射相邻块的左下控制点相关的第三仿射种子矢量和与仿射相邻块的右下控制点相关的第四仿射种子矢量来导出当前块的第一仿射种子矢量和第二仿射种子矢量,

$$v_{0x} = \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(x_0-x_{n2}) - \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(y_0-y_{n2}) + nv_{2x}$$

$$v_{0y} = \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(x_0-x_{n2}) - \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(y_0-y_{n2}) + nv_{2y}$$

$$v_{1x} = \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(x_1-x_{n2}) - \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(y_1-y_{n2}) + nv_{2x}$$

$$v_{1y} = \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(x_1-x_{n2}) - \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(y_1-y_{n2}) + nv_{2y}$$

$(x_{n2}, y_{n2})$  表示仿射相邻块的左下控制点的坐标,并且  $(x_{n3}, y_{n3})$  表示仿射相邻块的右下控制点的坐标;

$(x_0, y_0)$  表示当前块的左上控制点的坐标,并且  $(x_1, y_1)$  表示当前块的右上控制点的坐标;

$(nv_{2x}, nv_{2y})$  表示仿射相邻块的左下控制点的第三仿射种子矢量,并且  $(nv_{3x}, nv_{3y})$  表示仿射相邻块的右下控制点的第四仿射种子矢量;

$(v_{0x}, v_{0y})$  表示当前块的左上控制点的第一仿射种子矢量,并且  $(v_{1x}, v_{1y})$  表示当前块的右上控制点的第二仿射种子矢量。

15. 一种视频解码装置,包括:

生成当前块的合并候选列表的模块;指定所述合并候选列表中包括的多个合并候选之一的模块;

基于所指定的合并候选的第一仿射种子矢量和第二仿射种子矢量,导出所述当前块的第一仿射种子矢量和第二仿射种子矢量的模块;

通过使用所述当前块的所述第一仿射种子矢量和所述第二仿射种子矢量导出所述当前块中的子块的仿射矢量的模块,其中,所述子块是大小小于所述当前块的大小的区域;以及

基于所述仿射矢量对所述子块执行运动补偿预测的模块,其中,

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于与所述当前

块相邻的相邻块的运动信息导出的；

当所述相邻块被包括在与所述当前块的编码树单元不同的编码树单元中时，所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的左下子块和右下子块的运动矢量导出的。

16. 根据权利要求15所述的视频解码装置，其中，

所述左下子块包括位于所述相邻块的左下角的左下参考样本，并且所述右下子块包括位于所述相邻块的右下角的右下参考样本。

17. 根据权利要求16所述的视频解码装置，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的，

并且所述比例因子是基于通过左下参考样本与右下参考样本之间的水平距离与偏移量相加获得的值导出的。

18. 根据权利要求16所述的视频解码装置，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的，并且所述比例因子是基于所述右下参考样本右侧相邻的相邻样本与所述左下参考样本之间的距离导出的。

19. 根据权利要求15所述的视频解码装置，其中，

所述合并候选列表包括第一合并候选和第二合并候选，所述第一合并候选是基于被确定为位于所述当前块上方的上相邻块中的第一可用块导出的，并且所述第二合并候选是基于被确定为位于所述当前块左侧的左相邻块中的第一可用块导出的。

20. 根据权利要求15所述的视频解码装置，其中，

当所述相邻块被包括在与所述当前块的编码树单元相同的编码树单元中时，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的第一仿射种子矢量和第二仿射种子矢量导出的。

21. 根据权利要求15所述的视频解码装置，其中，使用与仿射相邻块的左下控制点相关的第三仿射种子矢量和与仿射相邻块的右下控制点相关的第四仿射种子矢量来导出当前块的第一仿射种子矢量和第二仿射种子矢量，

$$v_{0x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x}$$

$$v_{0y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2y}$$

$$v_{1x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2x}$$

$$v_{1y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2y}$$

$(x_{n2}, y_{n2})$  表示仿射相邻块的左下控制点的坐标，并且  $(x_{n3}, y_{n3})$  表示仿射相邻块的右下控制点的坐标；

$(x_0, y_0)$  表示当前块的左上控制点的坐标，并且  $(x_1, y_1)$  表示当前块的右上控制点的坐标；

$(nv_{2x}, nv_{2y})$  表示仿射相邻块的左下控制点的第三仿射种子矢量，并且  $(nv_{3x}, nv_{3y})$  表示仿射相邻块的右下控制点的第四仿射种子矢量；

$(v_{0x}, v_{0y})$  表示当前块的左上控制点的第一仿射种子矢量，并且  $(v_{1x}, v_{1y})$  表示当前块的右上控制点的第二仿射种子矢量。

22. 一种视频编码装置，包括：

生成当前块的合并候选列表的模块；指定所述合并候选列表中包括的多个合并候选之一的模块；

基于所指定的合并候选的第一仿射种子矢量和第二仿射种子矢量，导出所述当前块的第一仿射种子矢量和第二仿射种子矢量的模块；

通过使用所述当前块的所述第一仿射种子矢量和所述第二仿射种子矢量导出所述当前块中的子块的仿射矢量的模块，其中，所述子块是大小小于所述当前块的大小的区域；以及

基于所述仿射矢量对所述子块执行运动补偿预测的模块，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于与所述当前块相邻的相邻块的运动信息导出的；

当所述相邻块被包括在与所述当前块的编码树单元不同的编码树单元中时，所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的左下子块和右下子块的运动矢量导出的。

23. 根据权利要求22所述的视频编码装置，其中，

所述左下子块包括位于所述相邻块的左下角的左下参考样本，并且所述右下子块包括位于所述相邻块的右下角的右下参考样本。

24. 根据权利要求23所述的视频编码装置，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的，

并且所述比例因子是基于通过左下参考样本与右下参考样本之间的水平距离与偏移量相加获得的值导出的。

25. 根据权利要求23所述的视频编码装置，其中，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是使用比例因子根据对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值导出的，

并且所述比例因子是基于所述右下参考样本右侧相邻的相邻样本与所述左下参考样本之间的距离导出的。

26. 根据权利要求22所述的视频编码装置，其中，

所述合并候选列表包括第一合并候选和第二合并候选，所述第一合并候选是基于被确定为位于所述当前块上方的上相邻块中的第一可用块导出的，并且所述第二合并候选是基于被确定为位于所述当前块左侧的左相邻块中的第一可用块导出的。

27. 根据权利要求22所述的视频编码装置，其中，

当所述相邻块被包括在与所述当前块的编码树单元相同的编码树单元中时，

所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量是基于所述相邻块的第一仿射种子矢量和第二仿射种子矢量导出的。

28. 根据权利要求22所述的视频编码装置,其中,使用与仿射相邻块的左下控制点相关的第三仿射种子矢量和与仿射相邻块的右下控制点相关的第四仿射种子矢量来导出当前块的第一仿射种子矢量和第二仿射种子矢量,

$$\begin{aligned} v_{0x} &= \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(x_0-x_{n2}) - \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(y_0-y_{n2}) + nv_{2x} \\ v_{0y} &= \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(x_0-x_{n2}) - \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(y_0-y_{n2}) + nv_{2y} \\ v_{1x} &= \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(x_1-x_{n2}) - \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(y_1-y_{n2}) + nv_{2x} \\ v_{1y} &= \frac{(nv_{3y}-nv_{2y})}{(x_{n3}-x_{n2})}(x_1-x_{n2}) - \frac{(nv_{3x}-nv_{2x})}{(x_{n3}-x_{n2})}(y_1-y_{n2}) + nv_{2y} \end{aligned}$$

$(x_{n2}, y_{n2})$  表示仿射相邻块的左下控制点的坐标,并且  $(x_{n3}, y_{n3})$  表示仿射相邻块的右下控制点的坐标;

$(x_0, y_0)$  表示当前块的左上控制点的坐标,并且  $(x_1, y_1)$  表示当前块的右上控制点的坐标;

$(nv_{2x}, nv_{2y})$  表示仿射相邻块的左下控制点的第三仿射种子矢量,并且  $(nv_{3x}, nv_{3y})$  表示仿射相邻块的右下控制点的第四仿射种子矢量;

$(v_{0x}, v_{0y})$  表示当前块的左上控制点的第一仿射种子矢量,并且  $(v_{1x}, v_{1y})$  表示当前块的右上控制点的第二仿射种子矢量。

29. 一种视频解码装置,包括存储器和处理器,所述存储器存储所述处理器可执行的指令,所述处理器执行所述指令时执行根据权利要求1至7中任一项所述的视频解码方法。

30. 一种视频编码装置,包括存储器和处理器,所述存储器存储所述处理器可执行的指令,所述处理器执行所述指令时执行根据权利要求8至14中任一项所述的视频编码方法。

31. 一种计算机可读记录介质,包括程序指令,所述程序指令在被计算机部件执行时执行根据权利要求1至7中任一项所述的视频解码方法。

32. 一种计算机可读记录介质,包括程序指令,所述程序指令在被计算机部件执行时执行根据权利要求8至14中任一项所述的视频编码方法。

## 视频编码/解码方法、视频编码/解码装置和存储介质

[0001] 本申请是申请日为2019年9月20日的PCT国际专利申请PCT/KR2019/012290进入中国国家阶段的中国专利申请号201980054695.2、发明名称为“视频信号编码/解码方法以及用于所述方法的设备”的分案申请。

[0002] 相关申请的交叉引用

[0003] 本申请基于申请号为10-2018-0114342、申请日为2018年9月21日的韩国专利申请提出，并要求该韩国专利申请的优先权，该韩国专利申请的全部内容在此引入本申请作为参考。

[0004] 本申请基于申请号为10-2018-0114343、申请日为2018年9月21日的韩国专利申请提出，并要求该韩国专利申请的优先权，该韩国专利申请的全部内容在此引入本申请作为参考。

[0005] 本申请基于申请号为10-2018-0114344、申请日为2018年9月21日的韩国专利申请提出，并要求该韩国专利申请的优先权，该韩国专利申请的全部内容在此引入本申请作为参考。

### 技术领域

[0006] 本发明涉及视频编码方法、视频解码方法、视频编码装置、视频解码装置以及存储介质。

### 背景技术

[0007] 随着显示面板变得越来越大，逐渐需要画质更高的视频服务。高清视频服务的最大问题在于数据量大幅增加，为了解决这种问题，正在积极进行用于提高视频压缩率的研究。作为代表性示例，在2009年，运动图像专家组 (Motion Picture Experts Group, MPEG) 和国际电信联盟-电信 (International Telecommunication Union-Telecommunication, ITU-T) 旗下的视频编码专家组 (Video Coding Experts Group, VCEG) 建立了视频编码联合协作团队JCT-VC (Joint Collaborative Team on Video Coding)。JCT-VC提出了视频压缩标准HEVC (高效视频编码, High Efficiency Video Coding) 并于2013年1月25日获得批准，其压缩性能约是H.264/AVC的2倍。随着高清视频服务的急剧发展，HEVC的性能也逐渐暴露出其局限性。

### 发明内容

[0008] 要解决的技术问题

[0009] 本发明的目的是提供一种在对视频信号进行编码/解码时使用仿射模型的帧间预测方法以及一种用于所述帧间预测方法的设备。

[0010] 本发明的目的是提供一种用于在对视频信号进行编码/解码时使用子块的平移运动矢量来导出仿射种子矢量的方法以及一种用于执行所述方法的设备。

[0011] 本发明的另一目的是提供一种用于在对视频信号进行编码/解码时通过将相邻块

与当前块之间的距离变换为2的幂级数来导出仿射种子矢量的方法以及一种用于执行所述方法的设备。

[0012] 本发明要实现的技术问题不限于以上提及的技术问题，并且本发明所属技术领域的普通技术人员将通过以下描述清楚理解未提到的其他技术问题。

### [0013] 技术方案

[0014] 根据本发明的视频信号解码/编码方法包括：生成当前块的合并候选列表；指定所述合并候选列表中包括的多个合并候选中的任一个；基于所指定的合并候选的第一仿射种子矢量和第二仿射种子矢量，导出所述当前块的第一仿射种子矢量和第二仿射种子矢量；通过使用所述当前块的所述第一仿射种子矢量和所述第二仿射种子矢量导出所述当前块中的子块的仿射矢量；以及基于所述仿射矢量对所述子块执行运动补偿预测。在这种情况下，所述子块是大小小于所述当前块的大小的区域。另外，可以基于与所述当前块相邻的相邻块的运动信息来导出所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量。

[0015] 在根据本发明的视频信号解码/编码方法中，当所述相邻块被包括在与所述当前块的编码树单元不同的编码树单元中时，可以基于所述相邻块的左下子块和右下子块的运动矢量来导出所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量。

[0016] 在根据本发明的视频信号解码/编码方法中，所述左下子块可以包括位于所述相邻块的左下角的左下参考样本，并且所述右下子块可以包括位于所述相邻块的右下角的右下参考样本。

[0017] 在根据本发明的视频信号解码/编码方法中，可以使用比例因子根据基于对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值来导出所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量，并且可以基于通过左下参考样本与右下参考样本之间的水平距离与偏移量相加获得的值来导出所述比例因子。

[0018] 在根据本发明的视频信号解码/编码方法中，可以使用比例因子根据基于对所述左下子块与所述右下子块之间的运动矢量的差值进行移位操作获得的值来导出所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量，并且可以基于所述右下参考样本右侧相邻的相邻样本与所述左下参考样本之间的距离来导出所述比例因子。

[0019] 在根据本发明的视频信号解码/编码方法中，所述合并候选列表包括第一合并候选和第二合并候选，所述第一合并候选是基于被确定为位于所述当前块上方的上相邻块中的第一可用块的上相邻块导出的，并且所述第二合并候选是基于被确定为位于所述当前块左侧的左相邻块中的第一可用块的左相邻块导出的。

[0020] 在根据本发明的视频信号解码/编码方法中，当所述相邻块被包括在与所述当前块的编码树单元相同的编码树单元中时，可以基于与所述相邻块的第一仿射种子矢量和第二仿射种子矢量来导出所述合并候选的所述第一仿射种子矢量和所述第二仿射种子矢量。

[0021] 以上对本发明简要概述的特征仅是稍后将描述的本发明的详细说明的示例性实施方式，并不限定本发明的范围。

### [0022] 发明效果

[0023] 根据本发明，通过使用仿射模型的帧间预测方法提高了预测效率。

[0024] 根据本发明，通过使用子块的平移运动矢量导出仿射种子矢量提高了编码效率。

[0025] 根据本发明，通过将相邻块与当前块之间的距离变换为2的幂级数以导出仿射种

子矢量，提高了编码效率。

[0026] 在本发明中可获得的效果不限于上述效果，并且本发明所属技术领域的普通技术人员将通过以下描述清楚理解未提到的其他效果。

## 附图说明

- [0027] 图1是本发明实施例的视频编码器(encoder)的框图。
- [0028] 图2是本发明实施例的视频解码器(decoder)的框图。
- [0029] 图3是示出本发明实施例的基本编码树单元的图。
- [0030] 图4是示出编码块的多种划分类型的图。
- [0031] 图5是示出编码树单元的划分示例的图。
- [0032] 图6是本发明实施例的帧间预测方法的流程图。
- [0033] 图7是示出对象的非线性运动的图。
- [0034] 图8是展示了根据本发明实施例的基于仿射运动的帧间预测方法的流程图。
- [0035] 图9是示出每个仿射运动模型的仿射种子矢量的示例的图。
- [0036] 图10是示出在4参数运动模型下的子块的仿射矢量的示例的图。
- [0037] 图11是示出可用于导出合并候选(여기 후보)的相邻块的图。
- [0038] 图12是示出基于仿射相邻块的仿射种子矢量来导出当前块的仿射种子矢量的图。
- [0039] 图13是示出将子块的运动矢量设置为仿射相邻块的仿射种子矢量的示例的图。
- [0040] 图14至图16是示出参考样本的位置的图。
- [0041] 图17是示出应用修改的仿射合并矢量导出方法的示例的图。
- [0042] 图18是示出基于非仿射相邻块来导出当前块的仿射种子矢量的示例的图。
- [0043] 图19是示出用另一个相邻块替换相邻块的示例的图。
- [0044] 图20和图21是示出使用多个合并候选的运动补偿预测方法的流程图。
- [0045] 图22是根据本发明实施例的帧内预测方法的流程图。
- [0046] 图23是示出每个参考样本线中包括的参考样本的图。
- [0047] 图24是示出帧内预测模式的图。
- [0048] 图25和图26是示出将参考样本排成一行的一维排列的示例的图。
- [0049] 图27是示出在角度帧内预测模式与平行于x轴的直线之间形成的角度的图。
- [0050] 图28是示出当当前块为非正方形时获得预测样本的示例的图。
- [0051] 图29是示出广角帧内预测模式的图。
- [0052] 图30是示出确定块强度的过程的流程图。
- [0053] 图31是示出了预定义的滤波器候选的图。
- [0054] 图32示出了将图像划分为多个图块的示例。
- [0055] 图33是示出根据灵活图块技术的图像划分模式的图。
- [0056] 图34是示出将图块ID分配给每个编码树单元的示例的图。
- [0057] 图35是示出选择性地确定是否对每个图块应用环路滤波器的示例的图。

## 具体实施方式

[0058] 以下，参照附图详细说明本发明的实施例。

[0059] 视频的编码和解码以块为单位来执行。例如,可以对编码块、变换块或预测块执行变换、量化、预测、环内滤波或重建等编码/解码处理。

[0060] 以下,将要编码/解码的块称为“当前块”。例如,根据当前编码/解码处理步骤,当前块可以表示编码块、变换块或预测块。

[0061] 另外,本说明书中使用的术语“单元”表示用于执行特定编码/解码过程的基本单元,并且“块”可以被理解为表示预定大小的样本阵列。除非另有说明,“块”和“单元”可互换使用。例如,在稍后描述的实施例中,编码块和编码单元可以被理解为具有相同的含义。

[0062] 图1是本发明实施例的视频编码器(encoder)的框图。

[0063] 参照图1,视频编码设备100可以包括图像划分部110、预测部120、125、变换部130、量化部135、重排部160、熵编码部165、逆量化部140、逆变换部145、滤波器部150和存储器155。

[0064] 图1所示的各部件被单独示出以表示视频编码设备中相互不同的特征性功能,并不表示各部件由分离的硬件或单一软件组件构成。即,为了便于说明,作为代表性结构要素列出并包括了各结构要素,并且至少两个结构要素可以合并成一个结构要素或者使得一个结构要素划分成多个结构要素并由此执行功能,只要不脱离本发明的本质,这种整合各结构要素的实施例和使得各结构要素分离的实施例也属于本发明的权利范围。

[0065] 另外,部分结构要素并不是在本发明中执行本质功能的必要结构要素,而是仅用于提高性能的可选结构要素。本发明可以通过仅包括实现本发明的本质所需的部件(不包括用于提高性能的结构要素)来实施,并且仅包括必要结构要素(不包括用于提高性能的结构要素)的结构也属于本发明的权利范围。

[0066] 图像划分部110可以将输入的图像划分为至少一个处理单元。在这种情况下,处理单元可以是预测单元(Prediction Unit:PU),也可以是变换单元(Transform Unit:TU),还可以是编码单元(Coding Unit:CU)。图像划分部110将一个图像划分为多个编码单元、预测单元和变换单元的组合,可以基于预定标准(例如,代价函数)选择编码单元、预测单元和变换单元的组合来对图像进行编码。

[0067] 例如,可以将一个图像划分为多个编码单元。为了将图像划分为编码单元,可以使用如四叉树结构(Quad Tree Structure)的递归树状结构,可以将一个视频或最大编码单元(largest coding unit)作为根而将编码单元划分为另外的编码单元,所述编码单元可以具有与所划分的编码单元数量一样多的子节点。根据某些限制不再被划分的编码单元将成为叶节点。即,当假设一个编码单元只能实现正方形划分时,最多可以将一个编码单元划分为4个另外的编码单元。

[0068] 以下,在本发明的实施例中,编码单元可以意指执行编码的单元,还可以意指执行解码的单元。

[0069] 可以将一个编码单元内的预测单元划分为至少一个相同大小的正方形或矩形,也可以将一个编码单元内的一个预测单元划分为具有与另一个预测单元不同的形状和/或大小。

[0070] 当基于编码单元执行帧内预测的预测单元并非是最小编码单元的情况下,可以执行帧内预测,而无需划分为多个预测单元N×N。

[0071] 预测部120、125可以包括执行帧间预测的帧间预测部120和执行帧内预测的帧内

预测部125。可以确定对于预测单元使用帧间预测还是执行帧内预测，并基于每种预测方法确定具体信息(例如，帧内预测模式、运动矢量、参考图像等)。在这种情况下，执行预测的处理单元有可能与确定预测方法和具体内容的处理单元不同。例如，可以由预测单元确定预测方法和预测模式等，并且可以由变换单元执行预测。可以将所生成的预测块与原始块之间的残差值(残差块)输入到变换部130。另外，可以将用于预测的预测模式信息、运动矢量信息等与残差值一同在熵编码部165中进行编码并传送至解码器。当使用特定的编码模式时，也可以直接对原始块进行编码并传送至解码器，而不通过预测部120、125生成预测块。

[0072] 帧间预测部120可以基于当前图像的上一个图像或下一个图像中的至少一个图像的信息来预测出预测单元，在一些情况下，还可以基于当前图像内已编码的一部分区域的信息来预测出预测单元。帧间预测部120可以包括参考图像插值部、运动预测部、运动补偿部。

[0073] 参考图像插值部从存储器155接收参考图像信息，并且可以从参考图像生成整数像素或分数像素的像素信息。对于亮度像素而言，为了以1/4像素为单位生成分数像素的像素信息，可以使用滤波器系数不同的基于DCT的8阶插值滤波器(DCT-based Interpolation Filter)。对于色度信号而言，为了以1/8像素为单位生成分数像素的像素信息，可以使用滤波器系数不同的基于DCT的4阶插值滤波器(DCT-based Interpolation Filter)。

[0074] 运动预测部可以基于通过参考图像插值部进行插值的参考图像来执行运动预测。用于计算运动矢量的方法可以使用全搜索块匹配算法(FBMA, Full search-based Block Matching Algorithm)、三步搜索法(TSS, Three Step Search)、新三步搜索算法(NTS, New Three-Step Search Algorithm)等多种方法。基于插值的像素，运动矢量可以具有以1/2像素或1/4像素为单位的运动矢量值。在运动预测部中，可以通过采取不同的运动预测方法来预测当前预测单元。运动预测方法可以使用跳过(Skip)方法、合并(Merge)方法、高级运动矢量预测(AMVP, Advanced Motion Vector Prediction)方法、帧内块复制(Intra Block Copy)方法等多种方法。

[0075] 帧内预测部125可以基于当前块周边的参考像素信息(所述参考像素信息是当前图像内的像素信息)来生成预测单元。在当前预测单元的相邻块是已执行帧间预测的块并且在参考像素是已执行帧间预测的像素的情况下，可以将已执行帧间预测的块中所包括的参考像素用作周边的已执行帧内预测的块的参考像素信息。即，在参考像素不可用的情况下，可以使用可用的参考像素中的至少一个参考像素来代替不可用的参考像素信息。

[0076] 在帧内预测中，预测模式可以具有根据预测方向使用参考像素信息的角度预测模式和在执行预测时不使用方向信息的非角度模式。用于预测亮度信息的模式和用于预测色度信息的模式可以不同，为了预测色度信息，可以使用用于预测亮度信息的帧内预测模式信息或所预测到的亮度信号信息。

[0077] 当执行帧内预测时，若预测单元的大小与变换单元的大小相同，则可以基于位于预测单元的左侧的像素、位于左上方的像素、位于上方的像素来对预测单元执行帧内预测。但是，当执行帧内预测时，若预测单元的大小与变换单元的大小不同，则可以基于变换单元的参考像素来执行帧内预测。另外，可以仅对最小编码单元应用使用N×N划分的帧内预测。

[0078] 在根据预测模式对参考像素应用自适应帧内平滑(AIS, Adaptive Intra Smoothing)滤波器之后，帧内预测方法可以生成预测块。应用于参考像素的自适应帧内平

滑滤波器的类型可能不同。为了执行帧内预测方法,可以根据位于当前预测单元周边的预测单元的帧内预测模式预测出当前预测单元的帧内预测模式。在使用从周边预测单元预测到的模式信息来预测当前预测单元的预测模式的情况下,若当前预测单元与周边预测单元的帧内预测模式相同,则可以使用预定的标志信息来传送表示当前预测单元与周边预测单元的预测模式相同的信息,若当前预测单元与周边预测单元的预测模式不同,则可以通过执行熵编码来对当前块的预测模式信息进行编码。

[0079] 另外,可以生成包括残差信息的残差块,所述残差信息是基于在预测部120、125中所生成的预测单元来执行预测的预测单元与预测单元的原始块之间的差值。可以向变换部130输入所生成的残差块。

[0080] 变换部130可以使用诸如离散余弦变换(DCT)、离散正弦变换(DST)或变换跳过等变换方法来对残差块进行变换,所述残差块包括原始块与通过预测部120和125生成的预测单元之间的残差信息。对于要应用DCT还是应用DST又或者应用KLT来对残差块进行变换,可以基于用于生成残差块的预测单元的帧内预测模式信息来确定。

[0081] 量化部135可以对在变换部130中变换为频域的值进行量化。量化系数可以根据块或图像的重要性而改变。在量化部135中计算的值可以被提供给逆量化部140和重排部160。

[0082] 重排部160可以对量化的残差值执行系数值的重新排列。

[0083] 重排部160可以通过系数扫描(Coefficient Scanning)方法来将2维的块形状系数更改为1维的矢量形式。例如,重排部160可以使用Z字形扫描(Zig-Zag Scan)方法来扫描DC系数乃至高频域的系数,并将其改变为1维矢量形式。根据变换单元的大小和帧内预测模式,代替Z字形扫描,还可以使用沿着列方向扫描2维块形状系数的垂直扫描和沿着行方向扫描2维块形状系数的水平扫描。即,可以根据变换单元的大小和帧内预测模式来确定使用Z字形扫描、垂直方向扫描和水平方向扫描中的哪一种扫描方法。

[0084] 熵编码部165可以基于通过重排部160计算出的值来执行熵编码。例如,熵编码可以使用指数哥伦布码(Exponential Golomb)、上下文自适应变长编码(CAVLC, Context-Adaptive Variable Length Coding)、上下文自适应二进制算术编码(CABAC, Context-Adaptive Binary Arithmetic Coding)等的多种编码方法。

[0085] 熵编码部165可以对源自重排部160和预测部120、125的编码单元的残差值系数信息和块类型信息、预测模式信息、划分单元信息、预测单元信息和传送单元信息、运动矢量信息、参考帧信息、块的插值信息、滤波信息等多种信息进行编码。

[0086] 熵编码部165可以对从重排部160输入的编码单元的系数值进行熵编码。

[0087] 逆量化部140和逆变换部145对由量化部135量化的多个值进行逆量化,并对由变换部130变换的值进行逆变换。可以通过将在逆量化部140和逆变换部145中生成的残差值与由预测部120、125中包括的运动预测部、运动补偿部和帧内预测部所预测的预测单元合并来生成重建块(Reconstructed Block)。

[0088] 滤波器部150可以包括去块滤波器、偏移校正部(offset correction unit)、自适应环路滤波器(ALF, Adaptive Loop Filter)中的至少一个。

[0089] 去块滤波器可以移除因块之间的边界而在重建的图像中生成的块失真。为了判断是否执行去块,可以基于块中所包括的几个列或行所包括的像素来判断是否对当前块应用去块滤波器。在对块应用去块滤波器的情况下,可以根据所需的去块滤波强度来应用强滤

器(Strong Filter)或弱滤器(Weak Filter)。另外,在使用去块滤波器的过程中,当执行垂直滤波和水平滤波时,可以同步处理水平方向滤波和垂直方向滤波。

[0090] 偏移校正部可以按像素单位来校正去块图像与原始图像之间的偏移。可以通过以下方式来对指定图像进行偏移校正:在将图像中所包括的像素划分为预定数量的区域之后,确定要执行偏移的区域,并对相应区域应用偏移,或者将各个像素的边缘信息考虑在内来应用偏移。

[0091] 可以基于对经过滤波的重建图像和原始图像进行比较的值来执行自适应环路滤波(ALF, Adaptive Loop Filtering)。将图像中包括的像素划分成预定的组之后,可以确定要用于相应组的一个滤波器,并且可以针对每个组有区别地执行滤波。可以针对每个编码单元(Coding Unit, CU)通过亮度信号传送与是否应用自适应环路滤波相关的信息,要应用的自适应环路滤波器的形状和滤波器系数可以根据各个块而不同。另外,还可以应用相同类型(固定类型)的ALF,而不管被应用的块的特性如何。

[0092] 存储器155可以存储通过滤波器部150计算的重建块或图像,可以在执行帧间预测时向预测部120、125提供所存储的重建块或图像。

[0093] 图2是本发明实施例的视频解码器(decoder)的框图。

[0094] 参照图2,视频解码器200可以包括熵解码部210、重排部215、逆量化部220、逆变换部225、预测部230、预测部235、滤波器部240、存储器245。

[0095] 在从视频编码器输入视频比特流时,可以按与视频编码器相反的步骤对所输入的比特流进行解码。

[0096] 熵解码部210可以按与视频编码器的熵编码部中执行熵编码的步骤相反的步骤进行熵解码。例如,与视频编码器中执行的方法相对应地,可以应用指数哥伦布码(Exponential Golomb)、上下文自适应变长编码(CAVLC, Context-Adaptive Variable Length Coding)、上下文自适应二进制算术编码(CABAC, Context-Adaptive Binary Arithmetic Coding)等多种方法。

[0097] 熵解码部210可以对与编码器执行的帧内预测和帧间预测相关的信息进行解码。

[0098] 重排部215可以基于在编码部中对由熵解码部210熵解码的比特流进行重新排列的方法来执行重新排列。可以将由1维矢量形式表示的多个系数重建成2维块形状的系数来进行重新排列。重排部215可以通过以下方式来执行重新排列:接收与由编码部执行的系数扫描相关的信息,并基于由相应编码部执行的扫描顺序来进行逆向扫描。

[0099] 逆量化部220可以基于编码器所提供的量化参数和重新排列的块的系数值来执行逆量化。

[0100] 针对由视频编码器执行的量化结果,逆变换部225可以针对变换部执行的变换即DCT、DST以及KLT来执行逆变换,即逆DCT、逆DST及逆KLT。逆变换可以基于在视频编码器中确定的传送单元来执行。在视频解码器的逆变换部225中,可以根据预测方法、当前块的大小及预测方向等多项信息来选择性地执行变换方法(例如,DCT、DST、KLT)。

[0101] 预测部230、235可以基于由熵解码部210提供的与预测块生成相关的信息和由存储器245提供的先前解码的块或图像信息来生成预测块。

[0102] 如上所述,当以与视频编码器中的操作相同的方式执行帧内预测时,若预测单元的大小与变换单元的大小相同,则基于位于预测单元的左侧的像素、位于左上方的像素、位

于上方的像素来对预测单元执行帧内预测,若执行帧内预测时的预测单元的大小与变换单元的大小不同,则可以使用基于变换单元的参考像素来执行帧内预测。另外,还可以仅对最小编码单元应用使用N×N划分的帧内预测。

[0103] 预测部230、235可以包括预测单元判定部、帧间预测部和帧内预测部。预测单元判定部接收从熵解码部210输入的预测单元信息、帧内预测方法的预测模式信息、帧间预测方法的运动预测相关信息等多种信息,根据当前编码单元对预测单元进行分类,并确定预测单元是在执行帧间预测还是在执行帧内预测。帧间预测部230可以使用由视频编码器提供的执行当前预测单元的帧间预测所需的信息,并基于当前预测单元所属的当前图像的上一个图像或下一个图像中的至少一个图像中所包括的信息来对当前预测单元执行帧间预测。可替代地,还可以基于在当前预测单元所属的当前图像中已重建的一部分区域的信息来执行帧间预测。

[0104] 为了执行帧间预测,可以基于编码单元来判断相应编码单元中所包括的预测单元的运动预测方法是跳过模式(Skip Mode)、合并模式(Merge Mode)、高级运动矢量预测模式(AMVP Mode)、帧内块复制模式中的哪种模式。

[0105] 帧内预测部235可以基于当前图像内的像素信息来生成预测块。当预测单元是已执行帧内预测的预测单元时,可以基于由视频编码器提供的预测单元的帧内预测模式信息来执行帧内预测。帧内预测部235可以包括自适应帧内平滑(AIS, Adaptive Intra Smoothing)滤波器、参考像素插值部、DC滤波器。自适应帧内平滑滤波器是对当前块的参考像素执行滤波的部分,并且可以根据当前预测单元的预测模式确定是否应用滤波器。可以使用由视频编码器提供的预测单元的预测模式和自适应帧内平滑滤波器信息来对当前块的参考像素执行自适应帧内平滑滤波。若当前块的预测模式是不执行自适应帧内平滑滤波的模式,则可以不应用自适应帧内平滑滤波器。

[0106] 若预测单元的预测模式是基于对参考像素进行插值的像素值来执行帧内预测的预测单元,则参考像素插值部可以通过对参考像素进行插值来生成整数值或分数值的像素单位的参考像素。若当前预测单元的预测模式是以不对参考像素进行插值的方式生成预测块的预测模式,则可以不对参考像素进行插值。若当前块的预测模式为DC模式,则DC滤波器可以通过滤波来生成预测块。

[0107] 可以向滤波器部240提供重建的块或图像。滤波器部240可以包括去块滤波器、偏移校正部、ALF。

[0108] 可以从视频编码器接收与是否对相应块或图像应用去块滤波器相关的信息以及与在应用去块滤波器时应用强滤器还是弱滤器相关的信息。视频解码器的去块滤波器可以接收由视频编码器提供的与去块滤波器相关的信息,并且视频解码器可以对相应块执行去块滤波。

[0109] 偏移校正部可以基于在编码时应用于图像的偏移校正的类型和偏移量信息等来对已重建的图像执行偏移校正。

[0110] 可以基于由编码器提供的与是否应用ALF相关的信息、ALF系数信息等来将ALF应用于编码单元。这种ALF信息可以通过被包括在特定的参数集中来提供。

[0111] 存储器245存储已重建的图像或块,使得所述图像或块可以用作参考图像或参考块,并且可以向输出部提供已重建的图像。

[0112] 图3是示出本发明实施例的基本编码树单元的图。

[0113] 可以将大小最大的编码块定义为编码树块。一个图像可以被划分为多个编码树单元 (Coding Tree Unit, CTU)。编码树单元是大小最大的编码单元,还可以被称为最大编码单元 (LCU, Largest Coding Unit)。图3示出了将一个图像划分为多个编码树单元的示例。

[0114] 编码树单元的大小可以以图像级或序列级定义。为此,可以通过图像参数集或序列参数集来用信号发送指示编码树单元的大小的信息。

[0115] 例如,可以将序列内整个图像的编码树单元的大小设置为 $128 \times 128$ 。可替代地,可以将图像级的 $128 \times 128$ 或 $256 \times 256$ 中的任一个确定为编码树单元的大小。例如,可以将第一图像中编码树单元的大小设置为 $128 \times 128$ ,并且可以将第二图像中编码树单元的大小设置为 $256 \times 256$ 。

[0116] 可以通过划分编码树单元来生成编码块。编码块表示用于进行编码/解码处理的基本单元。例如,可以按不同编码块执行预测或者变换,或者可以按不同编码块确定预测编码模式。其中,预测编码模式表示生成预测图像的方法。例如,预测编码模式可以包括画面内预测 (Intra Prediction, 帧内预测)、画面间预测 (Inter Prediction, 帧间预测)、当前图像参考 (Current Picture Referencing, CPR, 或帧内块复制 (Intra Block Copy, IBC)) 或组合预测 (Combined Prediction)。针对编码块,可以使用帧内预测、帧间预测、当前图像参考或组合预测中的至少一种预测编码模式来生成与编码块相关的预测块。

[0117] 可以通过比特流用信号发送表示当前块的预测编码模式的信息。例如,所述信息可以是指示预测编码模式为帧内模式还是帧间模式的1比特标志。可以仅在当前块的预测编码模式被确定为帧间模式的情况下使用当前图像参考或组合预测。

[0118] 当前图像参考用于将当前图像设置为参考图像,并从当前图像内的已编码/解码的区域获得当前块的预测块。其中,当前图像意味着包括当前块的图像。可以通过比特流用信号发送表示是否对当前块应用当前图像参考的信息。例如,所述信息可以是1比特标志。当所述标志为真时,可以将当前块的预测编码模式确定为当前图像参考,当所述标志为假时,可以将当前块的预测模式确定为帧间预测。

[0119] 可替代地,可以基于参考图像索引确定当前块的预测编码模式。例如,当参考图像索引指向当前图像时,可以将当前块的预测编码模式确定为当前图像参考。当参考图像索引指向其他图像而不是当前图像时,可以将当前块的预测编码模式确定为帧间预测。即,当前图像参考是使用当前图像中的已编码/解码的区域的信息的预测方法,并且帧间预测是使用已编码/解码的其他图像的信息的预测方法。

[0120] 组合预测表示由帧内预测、帧间预测和当前图像参考中的两种以上组合而成的编码模式。例如,在应用组合预测的情况下,可以基于帧内预测、帧间预测或当前图像参考中的一种来生成第一预测块,并且可以基于另一种来生成第二预测块。当生成第一预测块和第二预测块时,可以通过第一预测块和第二预测块的平均运算或加权和运算来生成最终预测块。可以通过比特流用信号发送表示是否应用组合预测的信息。所述信息可以是1比特标志。

[0121] 图4是示出编码块的多种划分类型的图。

[0122] 可以基于四叉树划分、二叉树划分或三叉树划分将编码块划分为多个编码块。还可以基于四叉树划分、二叉树划分或三叉树划分将所划分的编码块再次划分为多个编码

块。

[0123] 四叉树划分是指将当前块划分为4个块的划分技术。作为四叉树划分的结果,可以将当前块划分为4个正方形分区(参考图4的(a)部分中的‘SPLIT\_QT’)。

[0124] 二叉树划分是指将当前块划分为2个块的划分技术。可以将沿着垂直方向(即,使用横穿当前块的垂直线)将当前块划分为两个块的过程称为垂直方向二叉树划分,并且可以将沿着水平方向(即,使用横穿当前块的水平线)将当前块划分为两个块的过程称为水平方向二叉树划分。作为二叉树划分的结果,可以将当前块划分为2个非正方形分区。图4的(b)部分中的“SPLIT\_BT\_VER”表示垂直方向二叉树划分结果,并且图4的(c)部分中的“SPLIT\_BT\_HOR”表示水平方向二叉树划分结果。

[0125] 三叉树划分是指将当前块划分为3个块的划分技术。可以将沿着垂直方向(即,使用横穿当前块的两个垂直线)将当前块划分为三个块的过程称为垂直方向三叉树划分,并且可以将沿着水平方向(即,使用横穿当前块的两个水平线)将当前块划分为三个块的过程称为水平方向三叉树划分。作为三叉树划分的结果,可以将当前块划分为3个非正方形分区。在这种情况下,位于当前块的中心的分区的宽度/高度可以是其他分区的宽度/高度的2倍。图4的(d)部分中的“SPLIT\_TT\_VER”表示垂直方向三叉树划分结果,并且图4的(e)部分中的“SPLIT\_TT\_HOR”表示水平方向三叉树划分结果。

[0126] 可以将编码树单元的划分次数定义为划分深度(Partitioning Depth)。可以以序列或图像级确定编码树单元的最大划分深度。因此,编码树单元的最大划分深度可以根据不同序列或图像而变得不同。

[0127] 可替代地,可以针对多个划分技术的每一个单独确定最大划分深度。例如,允许四叉树划分的最大划分深度可以与允许二叉树划分和/或三叉树划分的最大划分深度不同。

[0128] 编码器可以通过比特流用信号发送表示当前块的划分类型或划分深度中的至少一个的信息。解码器可以基于从比特流中解析的信息来确定编码树单元的划分类型和划分深度。

[0129] 图5是示出编码树单元的划分示例的图。

[0130] 可以将使用四叉树划分、二叉树划分和/或三叉树划分等划分技术来划分编码块的过程称为多叉树划分(Multi Tree Partitioning)。

[0131] 可以将通过对编码块应用多叉树划分而生成的编码块称为下游编码块。当编码块的划分深度为k时,多个下游编码块的划分深度设置为k+1。

[0132] 另一方面,针对划分深度为k+1的编码块,可以将划分深度为k的编码块称为上游编码块。

[0133] 可以基于上游编码块的划分类型或相邻编码块的划分类型中的至少一个来确定当前编码块的划分类型。其中,相邻编码块与当前编码块相邻,其可以包括当前编码块的上相邻块、左相邻块或与左上角相邻的相邻块中的至少一个。其中,划分类型可以包括是否划分四叉树、是否划分二叉树、二叉树划分方向、是否划分三叉树或三叉树划分方向中的至少一个。

[0134] 为了确定编码块的划分类型,可以通过比特流用信号发送表示编码块是否被划分的信息。所述信息为1比特标志“split\_cu\_flag”,所述标志为真指示通过多叉树划分技术来划分编码块。

[0135] 当“split\_cu\_flag”为真时,可以通过比特流用信号发送表示编码块是否被四叉树划分的信息。所述信息为1比特标志“split\_qt\_flag”,当所述标志为真时,编码块可以被划分为4个块。

[0136] 例如,在图5所示的示例中图示了当编码树单元被四叉树划分时生成了划分深度为1的4个编码块。另外,图示了对作为四叉树划分的结果而生成的4个编码块中的第一个编码块和第四个编码块再次应用四叉树划分。最终,可以生成划分深度为2的4个编码块。

[0137] 另外,通过对划分深度为2的编码块再次应用四叉树划分来生成划分深度为3的编码块。

[0138] 在未对编码块应用四叉树划分时,可以通过考虑编码块的大小、编码块是否位于图像边界处、最大划分深度或相邻块的划分类型中的至少一个来确定对所述编码块执行二叉树划分还是三叉树划分。当确定对所述编码块执行二叉树划分或三叉树划分时,可以通过比特流用信号发送表示划分方向的信息。所述信息可以是1比特标志“mtt\_split\_cu\_vertical\_flag”。可以基于所述标志来确定划分方向是垂直方向还是水平方向。另外,可以通过比特流用信号发送表示对所述编码块应用二叉树划分或三叉树划分中的哪一个的信息。所述信息可以是1比特标志“mtt\_split\_cu\_binary\_flag”。可以基于所述标志来确定对所述编码块应用二叉树划分还是三叉树划分。

[0139] 例如,在图5所示的示例中图示了对划分深度为1的编码块应用垂直方向二叉树划分,对作为所述划分结果而生成的编码块中的左侧编码块应用垂直方向三叉树划分,而对右侧编码块应用垂直方向二叉树划分。

[0140] 帧间预测是指使用前一图像的信息来预测当前块的预测编码模式。例如,可以将前一图像内与当前块相同位置的块(以下称为同位块,Collocatedblock)设置为当前块的预测块。以下,将基于位置与当前块相同的块生成的预测块称为同位预测块(Collocated Prediction Block)。

[0141] 另一方面,若存在于前一图像中的对象已经移动至当前图像中的其他位置,则可以使用对象的运动来有效地预测当前块。例如,如果通过比较前一图像和当前图像,可以知道对象的移动方向和大小,则可以考虑对象的运动信息来生成当前块的预测块(或预测图像)。以下,可以将使用运动信息生成的预测块称为运动预测块。

[0142] 可以通过从当前块减去预测块来生成残差块(residualblock)。在这种情况下,当存在对象的运动时,通过使用运动预测块来代替同位预测块,可以减少残差块的能量,从而可以提高残差块的压缩性能。

[0143] 如上所述,可以将使用运动信息生成预测块的过程称为运动补偿预测。在大部分的帧间预测中,可以基于运动补偿预测来生成预测块。

[0144] 运动信息可以包括运动矢量、参考图像索引、预测方向或双向加权值索引中的至少一个。运动矢量表示对象的移动方向和大小。参考图像索引在参考图像列表中包括的多个参考图像中指定当前块的参考图像。预测方向是指单向L0预测、单向L1预测或双向预测(L0预测和L1预测)中的任一个。可以根据当前块的预测方向来使用L0方向的运动信息或L1方向的运动信息中的至少一个。双向加权值索引指定应用于L0预测块的加权值和应用于L1预测块的加权值。

[0145] 图6是本发明实施例的帧间预测方法的流程图。

[0146] 参考图6,帧间预测方法包括如下的步骤:确定当前块的帧间预测模式(S601);根据所确定的帧间预测模式来获得当前块的运动信息(S602);以及基于所获得的运动信息来对当前块执行运动补偿预测(S603)。

[0147] 其中,帧间预测模式表示用于确定当前块的运动信息的多种技术,并且可以包括使用平移(Transltion)运动信息的帧间预测模式和使用仿射(Affine)运动信息的帧间预测模式。例如,使用平移运动信息的帧间预测模式可以包括合并模式和高级运动矢量预测模式,并且使用仿射运动信息的帧间预测模式可以包括仿射合并模式和仿射运动矢量预测模式。根据帧间预测模式,可以基于与当前块相邻的相邻块或从比特流中解析的信息来确定当前块的运动信息。

[0148] 以下,详细说明使用仿射运动信息的帧间预测方法。

[0149] 图7是示出对象的非线性运动的图。

[0150] 视频内对象的运动有可能是非线性运动。例如,如图7所示的示例,可能发生摄像机放大(Zoom-in)、缩小(Zoom-out)、旋转(Rotation)或仿射变换等对象的非线性运动。当发生对象的非线性运动时,无法通过平移运动矢量来有效表示对象的运动。因此,在发生对象的非线性运动的部分,可以通过使用仿射运动来代替平移运动,由此提高编码效率。

[0151] 图8是展示了根据本发明实施例的基于仿射运动的帧间预测方法的流程图。

[0152] 可以基于从比特流解析的信息来确定是否对当前块应用基于仿射运动的帧间预测技术。具体地,可以基于指示是否对当前块应用仿射合并模式的标志或指示是否对当前块应用仿射运动矢量预测模式的标志中的至少一个来确定是否对当前块应用基于仿射运动的帧间预测技术。

[0153] 当将基于仿射运动的帧间预测技术应用于当前块时,可以确定当前块的仿射运动模型(S801)。仿射运动模型可以确定为6参数仿射运动模型或4参数仿射运动模型中的至少一个。6参数仿射运动模型使用6个参数来表示仿射运动,并且4参数仿射运动模型使用4个参数来表示仿射运动。

[0154] 等式1是使用6个参数表示仿射运动的情况。仿射运动表示与通过仿射种子矢量确定的预定区域的平移运动。

[0155] 【等式1】

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

[0158] 当使用6个参数表示仿射运动时,可以表示复杂的运动,但是对各个参数进行编码时所需的比特数增加,因此会降低编码效率。因此,还可以使用4个参数表示仿射运动。等式2为使用4个参数表示仿射运动的情况。

[0159] 【等式2】

$$v_x = ax - by + e$$

$$v_y = bx + ay + f$$

[0162] 可以对用于确定当前块的仿射运动模型的信息进行编码并通过比特流用信号发送。例如,所述信息可以是1比特标志“affine\_type\_flag”。所述标志的值为0表示应用4参数仿射运动模型,所述标志的值为1表示应用6参数仿射运动模型。所述标志可以以条带(slice)、图块或块(例如,编码块或编码树单元)为单位进行编码。当以条带级用信号发送

标志时,可以对所述条带中所属的所有块应用以所述条带级确定的仿射运动模型。

[0163] 可替代地,可以基于当前块的仿射帧间预测模式来确定当前块的仿射运动模型。例如,当应用仿射合并模式时,可以将当前块的仿射运动模型确定为4参数运动模型。另一方面,当应用仿射运动矢量预测模式时,可以对用于确定当前块的仿射运动模型的信息进行编码并通过比特流用信号发送。例如,在对当前块应用仿射运动矢量预测模式时,可以基于1比特标志“affine\_type\_flag”来确定当前块的仿射运动模型。

[0164] 接下来,可以导出当前块的仿射种子矢量(S802)。当选择4参数仿射运动模型时,可以导出当前块的两个控制点上的运动矢量。另一方面,当选择6参数仿射运动模型时,可以导出当前块的三个控制点上的运动矢量。可以将控制点上的运动矢量称为仿射种子矢量。控制点可以包括当前块的左上角、右上角或左下角中的至少一个。

[0165] 图9是示出每个仿射运动模型的仿射种子矢量的示例的图。

[0166] 在4参数仿射运动模型中,可以导出与左上角、右上角或左下角中的两个相关的仿射种子矢量。例如,如图9(a)部分所示的示例,当选择4参数仿射运动模型时,可以通过使用与当前块的左上角(例如,左上样本(x0,y0))相关的仿射种子矢量sv<sub>0</sub>和与当前块的右上角(例如,右上样本(x1,y1))相关的仿射种子矢量sv<sub>1</sub>来导出仿射矢量。还可以使用与左下角相关的仿射种子矢量来代替与左上角相关的仿射种子矢量,或者还可以使用与左下角相关的仿射种子矢量来代替与右上角相关的仿射种子矢量。

[0167] 在6参数仿射运动模型中,可以导出与左上角、右上角和左下角相关的仿射种子矢量。例如,如图9(b)部分所示的示例,当选择6参数仿射运动模型时,可以通过使用与当前块的左上角(例如,左上样本(x0,y0))相关的仿射种子矢量sv<sub>0</sub>、与当前块的右上角(例如,右上样本(x1,y1))相关的仿射种子矢量sv<sub>1</sub>和与当前块的左上角(例如,左上样本(x2,y2))相关的仿射种子矢量sv<sub>2</sub>来导出仿射矢量。

[0168] 在稍后所述的实施例中,在4参数仿射运动模型下,将左上控制点和右上控制点的仿射种子矢量分别称为第一仿射种子矢量和第二仿射种子矢量。在稍后所述的使用第一仿射种子矢量和第二仿射种子矢量的实施例中,第一仿射种子矢量和第二仿射种子矢量中的至少一个可以替换为左下控制点的仿射种子矢量(第三仿射种子矢量)或右下控制点的仿射种子矢量(第四仿射种子矢量)。

[0169] 另外,在6参数仿射运动模型下,将左上控制点、右上控制点和左下控制点的仿射种子矢量分别称为第一仿射种子矢量、第二仿射种子矢量和第三仿射种子矢量。在稍后所述的使用第一仿射种子矢量、第二仿射种子矢量和第三仿射种子矢量的实施例中,第一仿射种子矢量、第二仿射种子矢量和第三仿射种子矢量中的至少一个可以替换为右下控制点的仿射种子矢量(第四仿射种子矢量)。

[0170] 可以使用仿射种子矢量来针对不同子块导出仿射矢量(S803)。其中,仿射矢量表示基于仿射种子矢量导出的平移运动矢量。可以将子块的仿射矢量称为仿射子块运动矢量或子块运动矢量。

[0171] 图10是示出在4参数运动模型下的子块的仿射矢量的示例的图。

[0172] 可以基于控制点的位置、子块的位置和仿射种子矢量而导出子块的仿射矢量。例如,等式3表示导出仿射子块矢量的示例。

[0173] 【等式3】

$$[0174] v_x = \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)} (x - x_0) - \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)} (y - y_0) + sv_{0x}$$

$$[0175] v_y = \frac{(sv_{1y} - sv_{0y})}{(x_1 - x_0)} (x - x_0) - \frac{(sv_{1x} - sv_{0x})}{(x_1 - x_0)} (y - y_0) + sv_{0y}$$

[0176] 在等式3中,  $(x, y)$  表示子块的位置。其中, 子块的位置表示子块中所包括的参考样本的位置。参考样本可以是位于子块的左上角的样本, 或者是其x轴或y轴坐标中的至少一个处于中心位置的样本。 $(x_0, y_0)$  表示第一控制点的位置, 并且  $(sv_{0x}, sv_{0y})$  表示第一仿射种子矢量。另外,  $(x_1, y_1)$  表示第二控制点的位置, 并且  $(sv_{1x}, sv_{1y})$  表示第二仿射种子矢量。

[0177] 当第一控制点和第二控制点分别与当前块的左上角和右上角相对应时, 可以将  $x_1 - x_0$  设置为与当前块的宽度相同的值。

[0178] 之后, 可以使用各个子块的仿射矢量来对各个子块执行运动补偿预测(S804)。在执行运动补偿预测后, 可以生成与各个子块相关的预测块。可以将子块的预测块设置为当前块的预测块。

[0179] 可以基于与当前块相邻的相邻块的仿射种子矢量来导出当前块的仿射种子矢量。在当前块的帧间预测模式为仿射合并模式的情况下, 可以将合并候选列表所包括的合并候选的仿射种子矢量确定为当前块的仿射种子矢量。另外, 在当前块的帧间预测模式为仿射合并模式的情况下, 可以将包含当前块的参考图像索引、特定方向预测标志或双向加权值中的至少一个的运动信息设置为与合并候选相同。

[0180] 可以基于当前块的相邻块导出合并候选。所述相邻块可以包括与当前块空间相邻的空间相邻块和与当前图像不同的图像中所包括的时间相邻块中的至少一个。

[0181] 图11是示出可用于导出合并候选的相邻块的图。

[0182] 当前块的相邻块可以包括与当前块的左侧相邻的相邻块(A)、与当前块的上方相邻的相邻块(B)、与当前块的右上角相邻的相邻块(C)、与当前块的左下角相邻的相邻块(D)或与当前块的左上角相邻的相邻块中的至少一个。若当前块的左上样本的坐标为  $(x_0, y_0)$ , 则左相邻块A包括在  $(x_0-1, y_0+H-1)$  位置上的样本, 并且上相邻块B包括在  $(x_0+W-1, y_0-1)$  位置上的样本。其中, W和H分别表示当前块的宽度和高度。右上相邻块C包括在  $(x_0+W, y_0-1)$  位置上的样本, 并且左下相邻块D包括在  $(x_0-1, y_0+H)$  位置上的样本。左上相邻块E包括在  $(x_0-1, y_0-1)$  位置上的样本。

[0183] 当以仿射帧间预测模式对相邻块进行编码时, 可以基于相应相邻块的仿射种子矢量来导出合并候选的仿射种子矢量。在下文中, 以仿射帧间预测模式编码的相邻块将被称为仿射相邻块。

[0184] 通过以预定义的扫描顺序搜索相邻块, 可以生成当前块的合并候选。可以在编码器和解码器中预定义扫描顺序。例如, 可以按A、B、C、D、E顺序搜索相邻块。另外, 可以从搜索到的仿射相邻块顺序地导出合并候选。可替代地, 可以基于当前块的大小、形状或仿射运动模型中的至少一个来自适应地确定扫描顺序。即, 具有不同的大小、形状或仿射运动模型中的至少一个的块的扫描顺序会不同。

[0185] 可替代地, 顺序地搜索位于当前块上方的块并从首先找到的仿射相邻块导出合并候选, 并且顺序地搜索位于当前块左侧的块并从首先找到的仿射相邻块导出合并候选。其

中,位于当前块上方的多个相邻块包括相邻块E、相邻块B或相邻块C中的至少一个,位于当前块左侧的多个块可以包括块A或块D中的至少一个。在这种情况下,还可以将相邻块E分类为位于当前块左侧的块。

[0186] 尽管未示出,但是可以从当前块的时间相邻块导出合并候选。其中,时间相邻块可以包括位于与同位图像中的当前块相同位置的块或与其相邻的块。具体地,当以仿射帧间预测模式对当前块的时间相邻块进行编码时,可以基于时间合并候选的仿射种子矢量来导出合并候选。

[0187] 可以生成包括合并候选的合并候选列表,并且可以将在合并候选列表中包括的合并候选之一的仿射种子矢量确定为当前块的仿射种子矢量。为此,可以对标识多个合并候选中的任一个的索引信息进行编码并且通过比特流传送。

[0188] 作为另一示例,可以按照扫描顺序搜索多个相邻块,并且可以从首先找到的仿射相邻块的仿射种子矢量导出当前块的仿射种子矢量。

[0189] 如上所述,在仿射合并模式下,可以使用相邻块的仿射种子矢量来导出当前块的仿射种子矢量。

[0190] 在当前块的帧间预测模式为仿射运动矢量预测模式的情况下,可以将运动矢量预测候选列表所包括的运动矢量预测候选的仿射种子矢量确定为当前块的仿射种子矢量预测值。可以通过所述仿射种子矢量预测值加上仿射种子矢量差值来导出当前块的仿射种子矢量。

[0191] 可以基于当前块的相邻块来导出仿射种子矢量预测候选。具体地,按照预定扫描顺序来搜索位于当前块上方的多个相邻块,并且可以从首先找到的仿射相邻块导出第一仿射种子矢量预测候选。另外,按照预定扫描顺序来搜索位于当前块左侧的多个相邻块,并且可以从首先找到的仿射相邻块导出第二仿射种子矢量预测候选。

[0192] 可以对用于确定仿射种子矢量差值的信息进行编码并通过比特流传送。所述信息可以包括表示仿射种子矢量差值的大小的大小信息和表示仿射种子矢量差值的符号的符号信号。可以将与各个控制点相关的仿射种子矢量差值设置为相同。可替代地,可以针对每个控制点不同地设置仿射种子矢量差值。

[0193] 如上所述,可以从仿射相邻块的仿射种子矢量导出合并候选或仿射种子矢量预测候选的仿射种子矢量,并且可以使用导出的合并候选或仿射种子矢量预测候选的仿射种子矢量来导出当前块的仿射种子矢量。可替代地,还可以在按预定的扫描顺序搜索多个仿射相邻块之后,从首先找到的仿射相邻块的仿射种子矢量导出当前块的仿射种子矢量。

[0194] 在下文中,将详细描述从仿射相邻块的仿射种子矢量导出当前块的仿射种子矢量、合并候选或仿射种子矢量预测候选的方法。在稍后描述的实施例中,导出当前块的仿射种子矢量可以理解为导出合并候选的仿射种子矢量或导出仿射种子矢量预测候选的仿射种子矢量。

[0195] 图12是示出基于仿射相邻块的仿射种子矢量来导出当前块的仿射种子矢量的图。

[0196] 若仿射相邻块存储有与左上控制点相关的第一仿射种子矢量 $nv_0$ 和与右上控制点相关的第二仿射种子矢量 $nv_1$ ,则可以基于所述第一仿射种子矢量和第二仿射种子矢量来导出与仿射相邻块的左下控制点相关的第三仿射种子矢量 $nv_2$ 。等式4表示导出第三仿射种子矢量的示例。

[0197] 【等式4】

$$[0198] nv_{2x} = \frac{(nv_{1x}-nv_{0x})}{(x_{n1}-x_{n0})}(x_{n2}-x_{n0}) - \frac{(nv_{1y}-nv_{0y})}{(x_{n1}-x_{n0})}(y_{n2}-y_{n0}) + nv_{0x}$$

$$[0199] nv_{2y} = \frac{(nv_{1y}-nv_{0y})}{(x_{n1}-x_{n0})}(x_{n2}-x_{n0}) - \frac{(nv_{1x}-nv_{0x})}{(x_{n1}-x_{n0})}(y_{n2}-y_{n0}) + nv_{0y}$$

[0200] 在等式4中,  $(nv_{0x}, nv_{0y})$  表示第一仿射种子矢量  $nv_0$ ,  $(nv_{1x}, nv_{1y})$  表示第二仿射种子矢量  $nv_1$ , 并且  $(nv_{2x}, nv_{2y})$  表示第三仿射种子矢量  $nv_2$ 。另外,  $(x_{n0}, y_{n0})$  表示第一控制点的位置,  $(x_{n1}, y_{n1})$  表示第二控制点的位置, 并且  $(x_{n2}, y_{n2})$  表示第三控制点的位置。

[0201] 之后, 可以使用第一仿射种子矢量、第二仿射种子矢量和第三仿射种子矢量来导出当前块的仿射种子矢量。等式5示出了导出当前块的第一仿射种子矢量  $v_0$  的示例, 并且等式6示出了导出当前块的第二仿射种子矢量  $v_1$  的示例。

[0202] 【等式5】

$$[0203] v_{0x} = \frac{(nv_{1x}-nv_{0x})}{(x_{n1}-x_{n0})}(x_0-x_{n0}) - \frac{(nv_{2x}-nv_{0x})}{(y_{n2}-x_{n0})}(y_0-y_{n0}) + nv_{0x}$$

$$[0204] v_{0y} = \frac{(nv_{1y}-nv_{0y})}{(x_{n1}-x_{n0})}(x_0-x_{n0}) - \frac{(nv_{2y}-nv_{0y})}{(y_{n2}-x_{n0})}(y_0-y_{n0}) + nv_{0y}$$

[0205] 【等式6】

$$[0206] v_{1x} = \frac{(nv_{1x}-nv_{0x})}{(x_{n1}-x_{n0})}(x_1-x_{n0}) - \frac{(nv_{2x}-nv_{0x})}{(y_{n2}-x_{n0})}(y_1-y_{n0}) + nv_{0x}$$

$$[0207] v_{1y} = \frac{(nv_{1y}-nv_{0y})}{(x_{n1}-x_{n0})}(x_1-x_{n0}) - \frac{(nv_{2y}-nv_{0y})}{(y_{n2}-x_{n0})}(y_1-y_{n0}) + nv_{0y}$$

[0208] 在等式5和等式6中,  $(v_{0x}, v_{0y})$  表示当前块的第一仿射种子矢量  $sv_0$ , 并且  $(v_{1x}, v_{1y})$  表示当前块的第二仿射种子矢量  $sv_1$ 。另外,  $(x_0, y_0)$  表示第一控制点的位置, 并且  $(x_1, y_1)$  表示第二控制点的位置。例如, 第一控制点表示当前块的左上角, 并且第二控制点表示当前块的右上角。

[0209] 在所述示例中说明了使用与仿射相邻块相关的3个仿射种子矢量来导出当前块的多个仿射种子矢量。作为另一示例, 还可以仅使用仿射相邻块的多个仿射种子矢量中的2个来导出当前块的仿射种子矢量。

[0210] 可替代地, 可以使用与右下角相关的第四仿射种子矢量来导出当前块的多个仿射种子矢量, 而不是使用与仿射相邻块相关的左上角的第一仿射种子矢量、右上角的第二仿射种子矢量或左下角的第三仿射种子矢量。

[0211] 尤其是, 在当前块的上边界与编码树单元的上边界接触并且要使用与当前块的上方相邻的仿射相邻块(以下称为上方仿射相邻块)的上方控制点(例如, 左上角或右上角)的仿射种子矢量的情况下, 需要将它们预先存储在存储器中, 这将有可能引起行缓冲区的数量增加的问题。因此, 当当前块的上边界与编码树单元的上边界接触时, 可以设置为对于上方仿射相邻块使用下方控制点(例如, 左下角或右下角)的仿射种子矢量, 而不是使用上方控制点的仿射种子矢量。例如, 可以使用与上方仿射相邻块的左下角相关的第三仿射种子

矢量和与右下角相关的第四仿射种子矢量来导出当前块的多个仿射种子矢量。在这种情况下,与下角相关的仿射种子矢量可以通过复制与上角相关的仿射种子矢量导出,或者可以从与多个上角相关的仿射种子矢量导出。例如,可以将第一仿射种子矢量、第二仿射种子矢量或第三仿射种子矢量变换/替换为与右下角相关的第四仿射种子矢量。

[0212] 等式7和等式8表示使用与相邻仿射矢量的左下控制点相关的第三仿射种子矢量和与右下控制点相关的第四仿射种子矢量来导出当前块的第一仿射种子矢量和第二仿射种子矢量的示例。

### [0213] 【等式7】

$$[0214] v_{0x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2x}$$

$$[0215] v_{0y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_0 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_0 - y_{n2}) + nv_{2y}$$

### [0216] 【等式8】

$$[0217] v_{1x} = \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2x}$$

$$[0218] v_{1y} = \frac{(nv_{3y} - nv_{2y})}{(x_{n3} - x_{n2})} (x_1 - x_{n2}) - \frac{(nv_{3x} - nv_{2x})}{(x_{n3} - x_{n2})} (y_1 - y_{n2}) + nv_{2y}$$

[0219] 在等式7和8中,  $(x_{n2}, y_{n2})$  表示仿射相邻块的左下控制点的坐标, 并且  $(x_{n3}, y_{n3})$  表示仿射相邻块的右下控制点的坐标。 $(x_0, y_0)$  表示当前块的左上控制点的坐标, 并且  $(x_1, y_1)$  表示当前块的右上控制点的坐标。 $(nv_{2x}, nv_{2y})$  表示仿射相邻块的左下控制点的仿射种子矢量(即, 第三仿射种子矢量), 并且  $(nv_{3x}, nv_{3y})$  表示仿射相邻块的右下控制点的仿射种子矢量(即, 第四仿射种子矢量)。 $(v_{0x}, v_{0y})$  表示当前块的左上控制点的仿射种子矢量(即, 第一仿射种子矢量), 并且  $(v_{1x}, v_{1y})$  表示当前块的右上控制点的仿射种子矢量(即, 第二仿射种子矢量)。

[0220] 还可以将等式7和等式8中所包括的除法运算更改为移位操作。移位操作可以基于由左下控制点与右下控制点之间的宽度(即,  $(x_{n3} - x_{n2})$ ) 导出的值来执行。

[0221] 在上述示例中, 可以根据已编码/解码的仿射相邻块的多个仿射种子矢量导出当前块的多个仿射种子矢量。为此, 需要将已编码/解码的仿射相邻块的多个仿射种子矢量存储在存储器中。但是, 由于除了仿射相邻块所包括的多个子块的多个平移运动矢量(即, 多个仿射矢量)之外, 还将仿射相邻块的多个仿射种子矢量存储在存储器中, 因而存在存储器使用量增加的问题。为了消除这种问题, 可以使用与仿射相邻块的控制点相邻的子块的运动矢量来导出当前块的仿射种子矢量, 由此代替仿射相邻块的仿射种子矢量。即, 可以将与仿射相邻块的控制点相邻的子块的运动矢量设置为仿射相邻块的仿射种子矢量。其中, 子块是具有已在编码器和解码器中预定义的大小/形状的块, 并且可以是具有存储运动矢量的基本大小/形状的块。例如, 子块可以是大小为 $4 \times 4$ 的正方形块。可替代地, 可以将指定样本位置的运动矢量设置为仿射相邻块的仿射种子矢量。

[0222] 图13是示出将子块的运动矢量设置为仿射相邻块的仿射种子矢量的示例的图。

[0223] 可以将与控制点相邻的子块的运动矢量设置为相应控制点的仿射种子矢量。例

如,在图13所示的示例中,可以将与仿射相邻块的左下角相邻的子块(左下子块)的运动矢量( $nv_{4x}, nv_{4y}$ )设置为左下控制点的仿射种子矢量( $nv_{2x}, nv_{2y}$ ),并且可以将与右下角相邻的子块(右下子块)的运动矢量( $nv_{5x}, nv_{5y}$ )设置为右下角的控制点的仿射种子矢量( $nv_{3x}, nv_{3y}$ )。其中,左下子块是指包括与相邻仿射块的左下控制点( $x_{n2}, y_{n2}$ )相邻的样本(例如,( $x_{n2}, y_{n2}-1$ )位置的样本)的子块,并且右下子块是指包括与相邻仿射块的右下控制点( $x_{n3}, y_{n3}$ )相邻的样本(例如,( $x_{n3}-1, y_{n3}-1$ )位置的样本)的块。当基于等式7和等式8导出当前块的仿射种子矢量时,仿射相邻块的第三仿射种子矢量可以由左下子块的运动矢量代替,并且第四仿射种子矢量可以由右下子块的运动矢量代替。

[0224] 以下,在稍后所述的实施例中,将用作仿射相邻块的仿射种子矢量的子块称为仿射子块。

[0225] 根据本发明的一实施例,可以基于特定位置处的样本来确定仿射子块。例如,还可以将包括特定位置处的样本的子块设置为仿射子块。以下,将特定位置的样本称为仿射参考样本。另外,将用于确定左下控制点的仿射子块的参考样本称为左下参考样本,并且将用于确定右下控制点的仿射子块的参考样本称为右下参考样本。

[0226] 左下参考样本和右下参考样本可以选自仿射相邻块所包括的多个样本。例如,可以将左下子块的左上样本、左下样本、右上样本或左下样本中的至少一个设置为左下参考样本,并且可以将右下子块的左上样本、左下样本、右上样本或左下样本中的至少一个设置为右下参考样本。因此,可以分别将包括左下参考样本的左下子块和包括右下参考样本的右下子块的运动矢量设置为与左下控制点相关的仿射种子矢量和与右下控制点相关的仿射种子矢量。

[0227] 作为另一示例,可以将左下参考样本或右下参考样本中的至少一个设置为位于仿射相邻块之外的样本。对此,将参考图14至图16进行详细说明。

[0228] 图14至图16是示出参考样本的位置的图。

[0229] 如图14(a)部分所示的示例,针对左下控制点,可以将左下子块的左上样本设置为参考样本( $x_{n4}, y_{n4}$ )。因此,可以将包括参考样本( $x_{n4}, y_{n4}$ )的左下子块设置为与左下控制点相关的仿射子块。

[0230] 针对右下控制点,可以将位于右下子块的右上样本的右侧的样本设置为参考样本( $x_{n5}, y_{n5}$ )。因此,可以将与包括参考样本( $x_{n5}, y_{n5}$ )的右下子块的右侧相邻的子块设置为与右下控制点相关的仿射子块。

[0231] 可替代地,如图14(b)部分所示的示例,针对左下控制点,可以将位于左下子块的左上样本的左侧的样本设置为参考样本( $x_{n4}, y_{n4}$ )。因此,可以将与包括参考样本( $x_{n4}, y_{n4}$ )的左下子块的左侧相邻的子块设置为与左下控制点相关的仿射子块。

[0232] 针对右下控制点,可以将右下子块的右上样本设置为参考样本( $x_{n5}, y_{n5}$ )。因此,可以将包括参考样本( $x_{n5}, y_{n5}$ )的右下子块设置为与右下控制点相关的仿射子块。

[0233] 可替代地,如图15(a)部分所示的示例,针对左下控制点,可以将左下子块的左下样本设置为参考样本( $x_{n4}, y_{n4}$ )。因此,可以将包括参考样本( $x_{n4}, y_{n4}$ )的左下子块设置为与左下控制点相关的仿射子块。

[0234] 针对右下控制点,可以将位于右下子块的右下样本的右侧的样本设置为参考样本( $x_{n5}, y_{n5}$ )。因此,可以将与包括参考样本( $x_{n5}, y_{n5}$ )的右下子块的右侧相邻的子块设置为与右

下控制点相关的仿射子块。

[0235] 可替代地,如图15(b)部分所示的示例,针对左下控制点,可以将位于左下子块的左下样本的左侧的样本设置为参考样本 $(x_{n4}, y_{n4})$ 。因此,可以将与包括参考样本 $(x_{n4}, y_{n4})$ 的左下子块的左侧相邻的子块设置为与左下控制点相关的仿射子块。

[0236] 针对右下控制点,可以将右下子块的右下样本设置为参考样本 $(x_{n5}, y_{n5})$ 。因此,可以将包括参考样本 $(x_{n5}, y_{n5})$ 的右下子块设置为与右下控制点相关的仿射子块。

[0237] 可替代地,如图16(a)部分所示的示例,针对左下控制点,可以将位于左下子块的左上样本与左下样本之间的样本(例如,左侧中间样本)设置为参考样本 $(x_{n4}, y_{n4})$ 。因此,可以将包括参考样本 $(x_{n4}, y_{n4})$ 的左下子块设置为与左下控制点相关的仿射子块。

[0238] 针对右下控制点,可以将位于右下子块的右上样本与右下样本之间的样本(例如,右侧中间样本)的右侧的样本设置为参考样本 $(x_{n5}, y_{n5})$ 。因此,可以将与包括参考样本 $(x_{n5}, y_{n5})$ 的右下子块的右侧相邻的子块设置为与右下控制点相关的仿射子块。

[0239] 可替代地,如图16(b)部分所示的示例,针对左下控制点,可以将位于左下子块的左上样本与左下样本之间的样本的左侧的样本设置为参考样本 $(x_{n4}, y_{n4})$ 。因此,可以将与包括参考样本 $(x_{n4}, y_{n4})$ 的左下子块的左侧相邻的子块设置为与左下控制点相关的仿射子块。

[0240] 针对右下控制点,可以将位于右下子块的右上样本与右下样本之间的样本设置为参考样本 $(x_{n5}, y_{n5})$ 。因此,可以将包括参考样本 $(x_{n5}, y_{n5})$ 的右下子块设置为与右下控制点相关的仿射子块。

[0241] 在基于等式7和等式8导出当前块的多个仿射种子矢量的情况下,仿射相邻块的第三仿射种子矢量可以由与左下控制点相关的仿射子块的运动矢量代替,并且第四仿射种子矢量可以由与右下控制点相关的仿射子块的运动矢量代替。另外,左下控制点的位置可以由左下参考样本的位置代替,并且右下控制点的位置可以由右下参考样本的位置代替。

[0242] 与图14至图16所述的内容不同,还可以将包括与参考样本相邻的样本的子块设置为仿射子块。具体地,可以将位于仿射相邻子块之外的样本设置为参考样本,并且可以将仿射相邻块所包括的子块设置为仿射子块。例如,在图14(a)部分所示的示例中,可以将位于右下子块的右上样本的右侧的样本设置为参考样本 $(x_{n5}, y_{n5})$ ,并且可以将右下子块设置为与右下角相关的仿射子块。可替代地,在图14(b)部分所示的示例中,可以将位于左下子块的左上样本的左侧的样本设置为参考样本 $(x_{n4}, y_{n4})$ ,并且可以将左下子块设置为与左下角相关的仿射子块。

[0243] 可以等同地应用图15和图16中所描述的实施例。即,在图15(a)部分或图16(a)部分所示的示例中,可以将右下子块的右下样本或位于右侧中间样本的右侧的样本设置为参考样本 $(x_{n5}, y_{n5})$ ,并且可以将右下子块设置为与右下角相关的仿射子块。可替代地,在图15(b)部分或图16(b)部分所示的示例中,可以将左下子块的左下样本或位于左侧中间样本的左侧的样本设置为参考样本 $(x_{n4}, y_{n4})$ ,并且可以将左下子块设置为与左下角相关的仿射子块。

[0244] 在上述示例中,可以通过使用仿射子块的运动矢量来导出仿射相邻块的仿射种子矢量。为此,对于已编码/解码的块而言,可以以子块为单位来存储运动矢量。

[0245] 作为另一示例,在仿射相邻块存储了最小数量的仿射种子矢量之后,可以使用所

存储的多个仿射种子矢量来导出仿射子块的运动矢量。

[0246] 等式9和等式10表示使用仿射相邻块的仿射种子矢量来导出仿射子块的运动矢量的示例。

**[0247] 【等式9】**

$$[0248] nv_{4x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n4} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n4} - y_{n0}) + nv_{0x}$$

$$[0249] nv_{4y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n4} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n4} - y_{n0}) + nv_{0y}$$

**[0250] 【等式10】**

$$[0251] nv_{5x} = \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (x_{n5} - x_{n0}) - \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (y_{n5} - y_{n0}) + nv_{0x}$$

$$[0252] nv_{5y} = \frac{(nv_{1y} - nv_{0y})}{(x_{n1} - x_{n0})} (x_{n5} - x_{n0}) - \frac{(nv_{1x} - nv_{0x})}{(x_{n1} - x_{n0})} (y_{n5} - y_{n0}) + nv_{0y}$$

[0253] 在等式9和等式10中,  $(nv_{4x}, nv_{4y})$  表示与左下控制点相关的仿射子块的运动矢量, 并且  $(nv_{5x}, nv_{5y})$  表示与右下控制点相关的仿射子块的运动矢量。由于仿射子块的运动矢量与控制点的仿射种子矢量被设置为相同, 因此  $(nv_{4x}, nv_{4y})$  可以由与左下控制点相关的仿射种子矢量  $(nv_{2x}, nv_{2y})$  代替, 或者  $(nv_{5x}, nv_{5y})$  可以由与右下控制点相关的仿射种子矢量  $(nv_{3x}, nv_{3y})$  代替。

[0254]  $(x_{n4}, y_{n4})$  表示左下子块的参考样本的位置。可替代地, 代替该位置, 还可以使用左下子块的中心位置或左下控制点的位置。 $(x_{n5}, y_{n5})$  表示右下子块的参考样本的位置。可替代地, 代替该位置, 还可以使用右下子块的中心位置或右下控制点的位置。

[0255] 可以在当前块不与编码树单元的边界相接触的情况下应用等式9和等式10。在当前块与编码树单元的上边界相接触的情况下, 代替使用等式9和等式10, 可以将基于左下参考样本确定的仿射子块的平移运动矢量设置为第三仿射种子矢量, 并且可以将基于右下参考样本确定的仿射子块的平移运动矢量设置为第四仿射种子矢量。

[0256] 在等式7和等式8中,  $(x_{n3} - x_{n2})$  表示左下控制点与右下控制点之间的宽度。如上所述,  $x_{n3}$  可以由右下参考样本的位置  $x_{n5}$  代替, 并且  $x_{n2}$  可以由左下参考样本的位置  $x_{n4}$  代替。以下, 将  $(x_{n3} - x_{n2})$  或通过由参考样本的位置代替前述等式的位置来获得的值(例如,  $(x_{n5} - x_{n4})$ ) 定义为变量  $W_{seed}$ , 所述变量称为子种子矢量宽度。

[0257] 根据参考样本的位置, 会发生子种子矢量宽度不是2的幂级数(例如,  $2^n$ )的情况。例如, 在将左下子块的左下样本设置为左下参考样本并将右下子块的右下样本设置为右下参考样本时, 子种子矢量的宽度并不是2的倍数。如上所述, 当子种子矢量宽度不是2的幂级数时, 可以将子种子矢量宽度变换为2的幂级数。所述变换可以包括将子种子矢量宽度加/减偏移量, 或者使用与所述参考样本相邻的样本的位置来代替参考样本的位置。例如, 可以通过左下参考样本与右下参考样本之间的宽度加1来导出所变换的子种子矢量宽度。可替代地, 可以将与右下参考样本的右侧相邻的相邻参考样本与左下参考样本之间的宽度设置为所变换的子种子矢量宽度。之后, 可以通过将变换的子种子矢量宽度代入等式7和等式8来导出当前块的仿射种子矢量。

[0258] 还可以将等式7和等式8中所包括的除法运算更改为移位操作。移位操作可以基于通过经变换的子种子矢量宽度(即,表示为2的幂级数的值)导出的值来执行。

[0259] 当用于确定仿射子块的参考样本不属于仿射相邻块时,可以基于仿射相邻块所包括的多个样本中与参考样本相邻的样本来导出仿射相邻块的仿射种子矢量。具体地,可以将包括与仿射相邻块中的参考样本相邻的样本(以下称为相邻参考样本)的子块的平移运动矢量设置为仿射相邻块的仿射种子矢量。如上所述,可以将使用相邻参考样本来导出仿射种子矢量的方法定义为修改的仿射合并矢量导出方法。

[0260] 图17是示出应用修改的仿射合并矢量导出方法的示例的图。

[0261] 当仿射相邻块E的右下参考样本( $x_{n5}, y_{n5}$ )不属于仿射相邻块时,可以基于仿射相邻块所包括的样本中的与右下参考样本的左侧相邻的样本( $x_{n5}-1, y_{n5}$ )来导出仿射种子矢量。具体地,可以将包括相邻参考样本( $x_{n5}-1, y_{n5}$ )的子块的平移运动矢量设置为右下控制点的仿射种子矢量。

[0262] 在图17所示的示例中,将与右下子块的右上样本的右侧相邻的样本示出为右下参考样本。在与右下子块的右下样本的右侧相邻的样本或与右下子块的右侧中间样本的右侧相邻的样本设置为右下参考样本的情况下,可以基于与相邻参考样本的左侧相邻的样本来导出仿射种子矢量。

[0263] 另外,当左下参考样本不属于仿射相邻块时,也可以按照所描述的实施例来基于与左下参考样本右侧相邻的样本导出仿射种子矢量。

[0264] 通过以不同的方式设置参考样本的位置和用于导出仿射种子矢量的子块,由此可以将子种子矢量宽度设置为2的幂级数。

[0265] 可以通过使用当前块周围未以仿射帧间模式编码的相邻块来导出当前块的合并候选、仿射种子矢量预测候选或仿射种子矢量。具体地,可以将未以仿射帧间模式编码的块进行组合,并且可以将其组合设置为合并候选或仿射种子矢量预测候选。例如,可以将与当前块的左上角相邻的相邻块中任一相邻块的运动矢量、与当前块的右上角相邻的相邻块中任一相邻块的运动矢量、以及与当前块的左下角相邻的相邻块中任一相邻块的运动矢量中的至少一者的组合设置为合并候选或仿射种子矢量预测候选。在这种情况下,与左上角相邻的相邻块的运动矢量、与右上角相邻的相邻块的运动矢量和与左下角相邻的相邻块的运动矢量可以分别设置为左上控制点的第一仿射种子矢量、右上控制点的第二仿射种子矢量和左下控制点的第三仿射种子矢量。

[0266] 可替代地,在上述修改的仿射合并矢量导出方法中,可以使用未以仿射帧间模式编码的相邻块来导出当前块的合并候选、仿射种子矢量预测候选或仿射种子矢量。在下文中,未以仿射帧间模式编码的相邻块将被称为非仿射相邻块。

[0267] 图18是示出基于非仿射相邻块来导出当前块的仿射种子矢量的示例的图。

[0268] 在图18所示的示例中,假设与当前块相邻的所有相邻块都是非仿射相邻块。

[0269] 当要从与当前块相邻的相邻块中的非仿射相邻块A导出当前块的仿射种子矢量时,可以设置A的左下参考样本和右下参考样本。例如,可以将与块A的左下样本左侧相邻的样本设置为左下参考样本,并且可以将块A的右下样本设置为右下参考样本。由于左下参考样本在块A之外,因此可以将包括与左下参考样本右侧相邻的样本的子块的运动矢量设置为块A的第三仿射种子矢量。另外,可以将包括右下参考样本的子块的运动矢量设置为块A

的第四仿射种子矢量。此后,可以基于等式9和10从块A导出当前块的第一仿射种子矢量和第二仿射种子矢量。

[0270] 仅当以子块为单位执行针对非仿射相邻块的运动补偿预测时,才可以使用从非仿射相邻块导出仿射种子矢量的方法。在此,用于以子块为单位执行运动补偿预测的预测技术可以包括STMVP、ATMVP、双向光流(BI0)、重叠块运动补偿(0BMC)和解码器侧运动矢量修正(DMVR)中的至少一种。

[0271] 在上述实施例中,已经描述了在当前块的上边界与编码树单元的边界接触时,使用位于当前块上方的仿射相邻块的左下控制点的第三仿射种子矢量和右下控制点的第四仿射种子矢量来导出当前块的合并候选、仿射种子矢量预测候选或仿射种子矢量。

[0272] 作为另一示例,如果当前块的上边界与编码树单元的边界接触,并且位于当前块上方的相邻块属于与当前块的编码树单元不同的编码树单元,则不是使用所述相邻块,而是使用当前块所属的编码树单元中包括的这些块中的与所述相邻块最接近的相邻块来导出当前块的合并候选、仿射种子矢量预测候选或仿射种子向量。

[0273] 在图19所示的示例中,示出了当前块与编码树单元的上边界接触,并且位于当前块上方的块B、C和E属于与当前块的编码树单元不同的编码树单元。因此,代替使用块E,可以使用当前块所属的编码树单元中包括的这些块中的与块E相邻的块F来导出当前块的仿射种子矢量。

[0274] 为了当前块的运动补偿预测,可以使用多个块的仿射种子矢量。例如,可以从合并候选列表中选择多个合并候选,并且可以基于所选合并候选的仿射种子矢量来导出当前块的仿射种子矢量或子块矢量。使用多个块的仿射种子矢量对当前块进行编码/解码可以被称为多仿射合并编码方法。

[0275] 可以对指示是否对当前块应用了多仿射合并编码方法的信息进行编码并通过比特流传输。可替代地,可以基于与当前块相邻的相邻块中的仿射相邻块的数量、合并候选列表中包括的合并候选的数量、以及当前块的仿射运动模型中的至少一者来确定是否对当前块应用了多仿射合并编码方法。

[0276] 图20和图21是示出使用多个合并候选的运动补偿预测方法的流程图。

[0277] 图20示出了通过使用多个合并候选的仿射种子矢量来导出当前块的仿射种子矢量的示例的图。图21是示出通过使用多个合并候选的仿射种子矢量来导出每个子块的运动矢量的示例的图。

[0278] 可以基于两个合并候选的仿射种子矢量的和、差、平均值、或加权和运算来生成当前块的仿射种子矢量。

[0279] 以下等式11和12示出了通过将合并候选的仿射种子矢量相加来导出当前块的仿射种子矢量的示例。

[0280] 【等式11】

$$(sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + (sv_{2x}, sv_{2y})$$

[0282] 【等式12】

$$(sv_{5x}, sv_{5y}) = (sv_{1x}, sv_{1y}) + (sv_{3x}, sv_{3y})$$

[0284] 在等式11和12中,sv<sub>4</sub>表示当前块的第一仿射种子矢量,sv<sub>0</sub>表示第一合并候选的第一仿射种子矢量,并且sv<sub>2</sub>表示第二合并候选的第一仿射种子矢量。另外,sv<sub>5</sub>表示当前块的

第二仿射种子矢量,  $sv_1$  表示第一合并候选的第二仿射种子矢量,  $sv_3$  表示第二合并候选的第二仿射种子矢量。

[0285] 另外, 以下等式13和14示出了通过合并候选的仿射种子矢量的加权和运算来导出当前块的仿射种子矢量的示例。

[0286] **【等式13】**

$$[0287] (sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + ((sv_{2x}, sv_{2y}) >> 1)$$

[0288] **【等式14】**

$$[0289] (sv_{5x}, sv_{5y}) = (sv_{1x}, sv_{1y}) + ((sv_{3x}, sv_{3y}) >> 1)$$

[0290] 作为另一示例, 可以基于第一合并候选的仿射种子矢量和第二合并候选的仿射种子矢量, 针对当前块中的每个子块生成第一子块运动矢量和第二子块运动矢量。此后, 可以基于第一子块运动矢量和第二子块运动矢量的和、差、平均值或加权和运算来生成最终子块运动矢量。

[0291] 以下等式15示出了通过将第一子块运动矢量和第二子块运动矢量相加而得出最终子块运动矢量的示例。

[0292] **【等式15】**

$$[0293] (V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + (V_{1x}, V_{1y})$$

[0294] 在等式15中,  $V_0$  表示第一子块运动矢量,  $V_1$  表示第二子块运动矢量, 并且  $V_2$  表示最终子块运动矢量。

[0295] 另外, 以下等式16示出了通过第一子块运动矢量和第二子块运动矢量的加权和运算来导出最终子块运动矢量的示例。

[0296] **【等式16】**

$$[0297] (V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + ((V_{1x}, V_{1y}) >> 1)$$

[0298] 帧内预测是使用当前块周边已编码/解码的重建样本来预测当前块。在这种情况下, 当前块的帧内预测可以使用应用环内滤波器之前的重建样本。

[0299] 帧内预测技术包括基于矩阵 (Matrix) 的帧内预测和考虑与周边重建样本的方向性的一般帧内预测。可以通过比特流用信号发送指示当前块的帧内预测技术的信息。所述信息可以是1比特标志。可替代地, 可以基于当前块的位置、大小、形状或相邻块的帧内预测技术中的至少一个来确定当前块的帧内预测技术。例如, 当当前块跨过图像边界存在时, 可以将当前块设置为不应用基于矩阵的帧内预测。

[0300] 基于矩阵的帧内预测是基于编码器和解码器中已存储的矩阵与当前块周边的重建样本之间的矩阵乘法来获得当前块的预测块的方法。可以通过比特流用信号发送用于指定已存储的多个矩阵中的任一个的信息。解码器可以基于所述信息和当前块的大小来确定用于当前块的帧内预测的矩阵。

[0301] 一般帧内预测是基于非角度帧内预测模式或角度帧内预测模式来获得当前块的预测块的方法。以下, 参考附图更详细地描述基于一般帧内预测执行帧内预测的过程。

[0302] 图22是根据本发明实施例的帧内预测方法的流程图。

[0303] 可以确定当前块的参考样本线 (S2201)。参考样本线是指从当前块的上方和/或左侧偏离第K个线中包括的参考样本的集合。可以从当前块周边已编码/解码的重建样本导出参考样本。

[0304] 可以通过比特流用信号发送标识多个参考样本线中的当前块的参考样本线的索引信息。多个参考样本线可以包括在当前块上方和/或左侧第一线、第二线、第三线或第四线中的至少一个。表1示出了分配给每个参考样本线的索引。在表1中，假设第一线、第二线和第四线被用作参考样本线候选。

[0305] 【表1】

| 索引 | 参考样本线   |
|----|---------|
| 0  | 第一参考样本线 |
| 1  | 第二参考样本线 |
| 2  | 第四参考样本线 |

[0307] 还可以基于当前块的位置、大小、形状或相邻块的预测编码模式中的至少一个确定当前块的参考样本线。例如，在当前块与图像、图块、条带或编码树单元的边界相接时，可以将第一参考样本线确定为当前块的参考样本线。

[0308] 参考样本线可以包括位于当前块上方的上方参考样本和位于当前块左侧的左侧参考样本。可以从当前块周边的重建样本导出上方参考样本和左侧参考样本。所述重建样本可以处于应用环内滤波器之前的状态。

[0309] 图23是示出每个参考样本线中包括的参考样本的图。

[0310] 根据当前块的帧内预测模式，可以利用属于参考样本线的参考样本中的至少一个来获取预测样本。

[0311] 接下来，可以确定当前块的帧内预测模式(S2202)。对于当前块的帧内预测模式而言，可以将非角度帧内预测模式或角度帧内预测模式中的至少一个确定为当前块的帧内预测模式。非角度帧内预测模式包括平面(Planer)和DC，并且角度帧内预测模式包括从左下对角线方向至右上对角线方向的33个或65个模式。

[0312] 图24是示出帧内预测模式的图。

[0313] 图24(a)示出35个帧内预测模式，并且图24(b)示出67个帧内预测模式。

[0314] 还可以定义相比于图24所示的更多或更少数量的帧内预测模式。

[0315] 可以基于与当前块相邻的相邻块的帧内预测模式来设置最可能模式(MPM, Most Probable Mode)。其中，相邻块可以包括与当前块的左侧相邻的左相邻块和与当前块的上方相邻的上相邻块。在当前块的左上样本的坐标为(0,0)时，左相邻块可以包括(-1,0)、(-1,H-1)或(-1,(H-1)/2)位置的样本。其中，H表示当前块的高度。上相邻块可以包括(0,-1)、(W-1,-1)或((W-1)/2,-1)位置的样本。其中，W表示当前块的宽度。

[0316] 当以一般帧内预测对相邻块进行编码时，可以基于相邻块的帧内预测模式导出MPM。具体地，可以将左相邻块的帧内预测模式设置为变量candIntraPredModeA，将上方相邻块的帧内预测模式设置为变量candIntraPredModeB。

[0317] 此时，当相邻块不可用时(例如，当相邻块尚未编码/解码或者相邻块的位置偏离图像边界时)，在相邻块以基于矩阵的帧内预测进行编码的情况下、相邻块以帧间预测进行编码的情况下或相邻块包括在与当前块不同的编码树单元中的情况下，可以将基于相邻块的帧内预测模式导出的变量candIntraPredModeX(其中，X为A或B)设置为默认模式。其中，默认模式可以包括平面模式、DC模式、垂直方向模式或水平方向模式中的至少一种。

[0318] 可替代地，当以基于矩阵的帧内预测对相邻块进行编码时，可以将与用于指定矩

阵中的任一个的索引值对应的帧内预测模式设置为candIntraPredModeX。为此,可以将指示用于指定矩阵的索引值和帧内预测模式的映射关系的查找表预先存储在编码器及解码器中。

[0319] 可以基于变量candIntraPredModeA及变量candIntraPredModeB来导出MPM。可以在编码器和解码器中预定义MPM列表所包括的MPM的数量。例如,MPM的数量可以是3个、4个、5个或6个。可替代地,可以通过比特流用信号发送表示MPM的数量的信息。可替代地,可以基于相邻块的预测编码模式、当前块的大小或形状中的至少一个来确定MPM的数量。

[0320] 在稍后描述的实施例中,假设MPM的数量为3个,这3个MPM将被称为MPM[0]、MPM[1]及MPM[2]。当MPM的数量多于3个时,MPM可以包括稍后描述的实施例中说明的3个MPM。

[0321] 当candIntraPredA与candIntraPredB相同且candIntraPredA为平面模式或DC模式时,可以分别将MPM[0]及MPM[1]设置为平面模式及DC模式。可以将MPM[2]设置为垂直帧内预测模式、水平帧内预测模式或对角线帧内预测模式。对角线帧内预测模式可以是左下对角线帧内预测模式、左上帧内预测模式或右上帧内预测模式。

[0322] 当candIntraPredA与candIntraPredB相同且candIntraPredA为帧内预测模式时,可以将MPM[0]与candIntraPredA设置为相同。可以将MPM[1]及MPM[2]设置为与candIntraPredA类似的帧内预测模式。与candIntraPredA类似的帧内预测模式可以是与candIntraPredA的索引差值为±1或±2的帧内预测模式。可以利用模运算(%)和偏移量来导出与candIntraPredA类似的帧内预测模式。

[0323] 当candIntraPredA与candIntraPredB不同时,可以将MPM[0]与candIntraPredA设置为相同,MPM[1]与candIntraPredB为相同。此时,当candIntraPredA和candIntraPredB均为非角度帧内预测模式时,可以将MPM[2]设置为垂直帧内预测模式、水平帧内预测模式或对角线帧内预测模式。可替代地,当candIntraPredA和candIntraPredB中的至少一个为角度帧内预测模式时,可以将MPM[2]设置为对具有平面、DC、或candIntraPredA或candIntraPredB中的更大值加上或减去偏移值来导出的帧内预测模式。其中,偏移值可以是1或2。

[0324] 可以生成包括多个MPM的MPM列表,并且可以通过比特流用信号发送表示与当前块的帧内预测模式相同的MPM是否包括在MPM列表中的信息。所述信息为1比特标志,可以称为MPM标志。当MPM标志指示与当前块相同的MPM包括在MPM列表中时,可以通过比特流用信号发送标识MPM之一的索引信息。可以将由所述索引信息指定的MPM设置为当前块的帧内预测模式。当MPM标志指示与当前块相同的MPM不包括在MPM列表中时,可以通过比特流用信号发送指示除了MPM以外的残差帧内预测模式中的任一个的残差模式信息。残差模式信息表示当向除了MPM以外的残差帧内预测模式重新分配索引时与当前块的帧内预测模式相对应的索引值。解码器可以按升序排列MPM,并且通过将残差模式信息与MPM进行比较来确定当前块的帧内预测模式。例如,当残差模式信息与MPM相同或者更小时,可以通过将残差模式信息加上1来导出当前块的帧内预测模式。

[0325] 代替将默认模式设置为MPM的操作,可以通过比特流用信号发送表示当前块的帧内预测模式是否为默认模式的信息。所述信息为1比特标志,并且所述标志可以称为默认模式标志。可以仅当MPM标志指示与当前块相同的MPM包括在MPM列表中时才用信号发送默认模式标志。如上所述,默认模式可以包括平面、DC、垂直方向模式或水平方向模式中的至少

一个。例如,当平面设置为默认模式时,默认模式标志可以指示当前块的帧内预测模式是否为平面。当默认模式标志指示当前块的帧内预测模式并不是默认模式时,可以将由索引信息指示的MPM之一设置为当前块的帧内预测模式。

[0326] 当将多个帧内预测模式设置为默认模式时,可以进一步用信号发送指示默认模式中的任一个的索引信息。可以将当前块的帧内预测模式设置为所述索引信息所指的默认模式。

[0327] 当当前块的参考样本线的索引不是0时,设置为不使用默认模式。因此,当参考样本线的索引不是0时,不用信号发送默认模式标志,并且可以将所述默认模式标志的值设置为预定义的值(即,假)。

[0328] 若确定当前块的帧内预测模式,则可以基于确定的帧内预测模式来获得与当前块相关的预测样本(S2203)。

[0329] 当选择DC模式时,可以基于参考样本的平均值生成与当前块相关的预测样本。具体地,可以基于参考样本的平均值生成预测块的所有样本的值。可以使用位于当前块上方的上方参考样本和位于当前块左侧的左侧参考样本中的至少一个来导出平均值。

[0330] 用于导出平均值的参考样本的数量或范围可能取决于当前块的形状而不同。例如,当当前块为宽度大于高度的非正方形块时,可以仅使用上方参考样本来计算平均值。另一方面,当当前块为宽度小于高度的非正方形块时,可以仅使用左侧参考样本来计算平均值。即,当前块的宽度和高度不同时,可以仅使用与较长长度相邻的参考样本来计算平均值。可替代地,可以基于当前块的宽度和高度比率来确定是否仅使用上方参考样本来计算平均值或者仅使用左侧参考样本来计算平均值。

[0331] 当选择平面模式时,可以使用水平方向预测样本和垂直方向预测样本来获得预测样本。其中,基于位于与预测样本相同的水平线上的左侧参考样本和右侧参考样本获得水平方向预测样本,并且基于位于与预测样本相同的垂直线上的上方参考样本和下方参考样本获得垂直方向预测样本。其中,可以通过复制与当前块的右上角相邻的参考样本而生成右侧参考样本,可以通过复制与当前块的左下角相邻的参考样本而生成下方参考样本。可以基于左侧参考样本和右侧参考样本的加权和运算获得水平方向预测样本,并且可以基于上方参考样本和下方参考样本的加权和运算获得垂直方向预测样本。在这种情况下,可以根据预测样本的位置来确定赋予各参考样本的加权值。可以基于水平方向预测样本和垂直方向预测样本的平均运算或加权和运算获得预测样本。当执行加权和运算时,可以基于预测样本的位置来确定赋予水平方向预测样本和垂直方向预测样本的加权值。

[0332] 当选择角度预测模式时,可以确定表示所选的角度预测模式的预测方向(或预测角度)的参数。下表2示出了每种帧内预测模式的帧内预测参数intraPredAng。

[0333] 【表2】

|        |                               |       |       |       |       |       |       |       |
|--------|-------------------------------|-------|-------|-------|-------|-------|-------|-------|
| [0334] | PredModeIntra<br>IntraPredAng | 1-    | 232   | 326   | 421   | 517   | 613   | 79    |
|        | PredModeIntraInt<br>raPredAng | 85    | 92    | 100   | 11-2  | 12-5  | 13-9  | 14-13 |
|        | PredModeIntraInt<br>raPredAng | 15-17 | 16-21 | 17-26 | 18-32 | 19-26 | 20-21 | 21-17 |
|        | PredModeIntraInt<br>raPredAng | 22-13 | 23-9  | 24-5  | 25-2  | 260   | 272   | 285   |
|        | PredModeIntraInt<br>raPredAng | 299   | 3013  | 3117  | 3221  | 3326  | 3432  |       |
|        |                               |       |       |       |       |       |       |       |

[0335] 表2示出了当定义了35个帧内预测模式时具有2至34中任一个的索引的每个帧内预测模式的帧内方向参数。当定义了超过33种角度帧内预测模式时,表2进一步细分为设置每个角度帧内预测模式的帧内方向参数。

[0336] 在将当前块的上方参考样本和左侧参考样本排成一行之后,可以基于帧内方向参数的值获得预测样本。在这种情况下,当帧内方向参数的值为负数时,可以将左侧参考样本和上方参考样本排成一行。

[0337] 图25和图26是示出将参考样本排成一行的一维排列的示例的图。

[0338] 图25示出了在垂直方向上排列参考样本的垂直方向一维阵列的示例,并且图26示出了在水平方向上排列参考样本的水平方向一维阵列的示例。将在定义35个帧内预测模式的假设下描述图25和26的实施例。

[0339] 当帧内预测模式索引为11至18中的任一个时,可以应用以逆时针方向旋转上方参考样本的水平方向一维排列,并且当帧内预测模式索引为19至25中的任一个时,可以应用以顺时针方向旋转左侧参考样本的垂直方向一维排列。将参考样本排成一行时,可以考虑帧内预测模式角度。

[0340] 基于帧内方向参数可以确定参考样本确定参数。参考样本确定参数可以包括用于指定参考样本的参考样本索引和用于确定应用于参考样本的加权值的加权值参数。

[0341] 参考样本索引*iIdx*和加权值参数*ifact*可以分别通过以下等式17和等式18获得。

[0342] 【等式17】

$$i\text{Idx} = (y+1) * P_{\text{ang}} / 32$$

[0344] 【等式18】

$$i_{\text{fact}} = [(y+1) * P_{\text{ang}}] \& 31$$

[0346] 在等式17和18中,  $P_{\text{ang}}$  表示帧内方向参数。根据参考样本索引*iIdx*指定的参考样本相当于整数像素(Integerpel)。

[0347] 为了导出预测样本,可以指定一个以上的参考样本。具体地,可以考虑预测模式的斜率来指定用于导出预测样本的参考样本的位置。例如,可以使用参考样本索引*iIdx*来指定用于导出预测样本的参考样本。

[0348] 在这种情况下,当帧内预测模式的斜率不由一个参考样本表示时,可以通过对多个参考样本进行插值来生成预测样本。例如,当帧内预测模式的斜率为预测样本与第一参考样本之间的斜率和预测样本与第二参考样本之间的斜率之间的值时,可以对第一参考样本和第二参考样本进行插值来获得预测样本。即,当跟随帧内预测角度的角线(Angular Line)不经过位于整数像素的参考样本时,可以对与所述角线经过的位置的左右或上下相邻的参考样本进行插值来获得预测样本。

[0349] 以下等式19表示基于参考样本来获得预测样本的示例。

[0350] 【等式19】

$$[0351] P(x,y) = ((32 - i_{fact}) / 32) * Ref\_1D(x+iIdx+1) + (i_{fact} / 32) * Ref\_1D(x+iIdx+2)$$

[0352] 在等式19中, P表示预测样本, 并且Ref\_1D表示一维排列的参考样本中的任何一个。在这种情况下, 可以根据预测样本的位置(x,y)和参考样本索引iIdx确定参考样本的位置。

[0353] 当帧内预测模式的斜率可以表示为一个参考样本时, 加权值参数ifact可以设置为0。因此, 等式19可以简化为如以下等式20所示。

[0354] 【等式20】

$$[0355] P(x,y) = Ref\_1D(x+iIdx+1)$$

[0356] 还可以基于多个帧内预测模式对当前块执行帧内预测。例如, 可以针对不同预测样本导出帧内预测模式, 并且可以基于分配到各个预测样本的帧内预测模式来导出预测样本。

[0357] 可替代地, 可以针对不同区域导出帧内预测模式, 并且可以基于分配到各个区域的帧内预测模式对各区域执行帧内预测。其中, 所述区域可以包括至少一个样本。可以基于当前块的大小、形状或帧内预测模式中的至少一个自适应地确定所述区域的大小或形状中的至少一个。可替代地, 可以在编码器和解码器中预定义区域的大小或形状中的至少一个, 而与当前块的大小或形状无关。

[0358] 可替代地, 可以分别基于多个帧内预测执行帧内预测, 并且可以基于通过多次的帧内预测获得的多个预测样本的平均运算或加权和运算导出最终预测样本。例如, 可以基于第一帧内预测模式执行帧内预测来获得第一预测样本, 并且可以基于第二帧内预测模式执行帧内预测来获得第二预测样本。之后, 可以基于第一预测样本和第二预测样本之间的平均运算或加权和运算来获得最终预测样本。在这种情况下, 可以考虑第一帧内预测模式是否为非角度/角度预测模式、第二帧内预测模式是否为非角度/角度预测模式或相邻块的帧内预测模式中的至少一个来确定分别分配于第一预测样本和第二预测样本的加权值。

[0359] 多个帧内预测模式可以是非角度帧内预测模式和角度预测模式的组合、角度预测模式的组合或非角度预测模式的组合。

[0360] 图27是示出在角度帧内预测模式与平行于x轴的直线之间形成的角度的图。

[0361] 如图27所示的示例中, 角度预测模式可以存在于左下对角线方向与右上对角线方向之间。当被描述为x轴与角度预测模式形成的角度时, 角度预测模式可以存在于45度(左下对角方向)至-135度(右侧上方对角方向)之间。

[0362] 如果当前块是非正方形形状, 则根据当前块的帧内预测模式, 使用位于跟随帧内预测角度的角线上的参考样本中更远离预测样本的参考样本而不是更接近预测样本的参考样本来导出预测样本。

[0363] 图28是示出当当前块为非正方形时获得预测样本的示例的图。

[0364] 例如, 如在图28(a)所示的示例中, 假设当前块是宽度大于高度的非正方形形状, 并且当前块的帧内预测模式是角度在0度至45度之间的角度帧内预测模式。在这种情况下, 在导出当前块的右侧列附近的预测样本A时, 会发生使用位于所述角度的角度模式上的参考样本中远离所述预测样本的左侧参考样本L来代替接近所述预测样本的上方参考样本T

的情况。

[0365] 作为另一示例,如在图28 (b) 所示的示例中,假设当前块是高度大于宽度的非正方形形状,并且当前块的帧内预测模式是角度在-90度至-135度之间的角度帧内预测模式。在上述情况下,在导出当前块的下方行附近的预测样本A时,会发生使用位于所述角度的角度模式上的参考样本中远离所述预测样本的上方参考样本T来代替接近所述预测样本的左侧参考样本L的情况。

[0366] 为了解除如上问题,当当前块为非正方形时,可以将当前块的帧内预测模式替换为反方向的帧内预测模式。因此,对于非正方形形状的块,可以使用具有比图24所示的角度预测模式的角度更大或更小的角度的角度预测模式。这种角度帧内预测模式可以被定义为广角帧内预测模式。广角帧内预测模式表示不落入45度至-135度范围内的角度帧内预测模式。

[0367] 图29是示出广角帧内预测模式的图。

[0368] 在图29所示的示例中,索引为-1至-14的帧内预测模式和索引为67至80的帧内预测模式表示广角帧内预测模式。

[0369] 尽管在图29中示出了具有大于45度的角度的14个广角帧内预测模式 (-1至-14) 和具有小于-135度的角度的14个广角帧内预测模式 (67至80),但可以定义更多或更少数量的广角帧内预测模式。

[0370] 当使用广角帧内预测模式时,上方参考样本的长度设置为 $2W+1$ ,并且左侧参考样本的长度设置为 $2H+1$ 。

[0371] 当使用广角帧内预测模式时,可以使用参考样本T预测图28 (a) 所示的样本A,并且可以使用参考样本L预测图28 (b) 所示的样本A。

[0372] 通过将现有帧内预测模式和N个广角帧内预测模式相加,可以使用总共 $67+N$ 个帧内预测模式。例如,表3示出了当定义了20个广角帧内预测模式时帧内预测模式的帧内方向参数。

[0373] 【表3】

|        |                |     |     |     |     |     |     |     |     |     |
|--------|----------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| [0374] | PredModeIntra  | -10 | -9  | -8  | -7  | -6  | -5  | -4  | -3  | -2  |
|        | intraPredAngle | 114 | 93  | 79  | 68  | 60  | 54  | 49  | 45  | 39  |
|        | PredModeIntra  | -1  | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   |
|        | intraPredAngle | 35  | 32  | 29  | 26  | 23  | 21  | 19  | 17  | 15  |
|        | PredModeIntra  | 10  | 11  | 12  | 13  | 14  | 15  | 16  | 17  | 18  |
|        | intraPredAngle | 13  | 11  | 9   | 7   | 5   | 3   | 2   | 1   | 0   |
|        | PredModeIntra  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  |
|        | intraPredAngle | -1  | -2  | -3  | -5  | -7  | -9  | -11 | -13 | -15 |
|        | PredModeIntra  | 28  | 29  | 30  | 31  | 32  | 33  | 34  | 35  | 36  |
|        | intraPredAngle | -17 | -19 | -21 | -23 | -26 | -29 | -32 | -29 | -26 |
|        | PredModeIntra  | 37  | 38  | 39  | 40  | 41  | 42  | 43  | 44  | 45  |
|        | intraPredAngle | -23 | -21 | -19 | -17 | -15 | -13 | -11 | -9  | -7  |
|        | PredModeIntra  | 46  | 47  | 48  | 49  | 50  | 51  | 52  | 53  | 54  |
|        | intraPredAngle | -5  | -3  | -2  | -1  | 0   | 1   | 2   | 3   | 5   |

|                |    |    |    |     |    |    |    |    |    |
|----------------|----|----|----|-----|----|----|----|----|----|
| PredModeIntra  | 55 | 56 | 57 | 58  | 59 | 60 | 61 | 62 | 63 |
| intraPredAngle | 7  | 9  | 11 | 13  | 15 | 17 | 19 | 21 | 23 |
| PredModeIntra  | 64 | 65 | 66 | 67  | 68 | 69 | 70 | 71 | 72 |
| intraPredAngle | 26 | 29 | 32 | 35  | 39 | 45 | 49 | 54 | 60 |
| PredModeIntra  | 73 | 74 | 75 | 76  |    |    |    |    |    |
| intraPredAngle | 68 | 79 | 93 | 114 |    |    |    |    |    |

[0375] 当当前块是非正方形形状的并且在步骤S2202中获取的当前块的帧内预测模式落入变换范围内时,可以将当前块的帧内预测模式变换为广角帧内预测模式。可以基于当前块的大小、形状或比率中的至少一个来确定所述变换范围。其中,所述比率可以表示当前块的宽度和高度之间的比率。

[0376] 当当前块为宽度大于高度的非正方形时,可以将变换范围设置为右侧上方对角方向的帧内预测模式索引(例如,66)至(右侧上方对角方向的帧内预测模式的索引-N)。其中,N可以基于当前块的比率来确定。当当前块的帧内预测模式落入变换范围内时,可以将所述帧内预测模式变换为广角帧内预测模式。可以从所述帧内预测模式减去预定义的值来执行所述变换,所述预定义的值可以是除了广角帧内预测模式之外的帧内预测模式的总数量(例如,67)。

[0377] 根据所述实施例,可以将第66个至第53个之间的帧内预测模式分别变换为第-1个至第-14个之间的广角帧内预测模式。

[0378] 当当前块为高度大于宽度的非正方形时,可以将变换范围设置为从左下对角方向的帧内预测模式索引(例如,2)至(左下对角方向的帧内预测模式的索引+M)。其中,M可以基于当前块的比率来确定。当当前块的帧内预测模式落入变换范围内时,可以将所述帧内预测模式变换为广角帧内预测模式。可以在所述帧内预测模式加上预定义的值来执行所述变换,预定义的值可以是除了广角帧内预测模式之外的角度帧内预测模式的总数量(例如,65)。

[0379] 根据所述实施例,可以将第2个至第15个之间的帧内预测模式分别变换为第67个至第80个之间的广角帧内预测模式。

[0380] 以下,将落入变换范围内的帧内预测模式称为广角帧内替换预测模式。

[0381] 变换范围可以基于当前块的比率来确定。例如,表4和表5各自示出当定义了除广角帧内预测模式之外的35个帧内预测模式和67个帧内预测模式时的变换范围。

[0382] 【表4】

| 条件      | 替换帧内预测模式         |
|---------|------------------|
| W/H=2   | 模式2,3,4          |
| W/H>2   | 模式2,3,4,5,6      |
| W/H=1   | 无                |
| H/W=1/2 | 模式32,33,34       |
| H/W<1/2 | 模式30,31,32,33,34 |

[0384] 【表5】

| 条件    | 替换帧内预测模式      |
|-------|---------------|
| W/H=2 | 模式2,3,4,5,6,7 |

|         |                                 |
|---------|---------------------------------|
| W/H>2   | 模式2,3,4,5,6,7,8,9,10,11         |
| W/H=1   | 无                               |
| H/W=1/2 | 模式61,62,63,64,65,66             |
| H/W<1/2 | 模式57,58,59,60,61,62,63,64,65,66 |

[0386] 如表4和表5中所示的示例,落入变换范围中的广角帧内替换预测模式的数量可以根据当前块的比率而不同。

[0387] 随着除现有帧内预测模式之外还使用广角帧内预测模式,用于对广角帧内预测模式进行编码所需的资源会增加,因此可能降低编码效率。因此,代替直接对广角帧内预测模式进行编码,而是对与广角帧内预测模式相关的替换帧内预测模式进行编码,从而提高编码效率。

[0388] 例如,当使用第67个广角帧内预测模式对当前块进行编码时,可以将作为第67个广角替换帧内预测模式的编号2编码为当前块的帧内预测模式。另外,当使用第-1个广角帧内预测模式对当前块进行编码时,可以将作为第-1个广角替换帧内预测模式的编号66编码为当前块的帧内预测模式。

[0389] 解码器可以对当前块的帧内预测模式进行解码,并确定解码的帧内预测模式是否包括在变换范围内。解码的帧内预测模式为广角替换帧内预测模式时,可以将帧内预测模式变换为广角帧内预测模式。

[0390] 可替代地,当以广角帧内预测模式对当前块进行编码时,还可以直接对广角帧内预测模式进行编码。

[0391] 帧内预测模式的编码可以基于所述MPM列表实现。在下文中,将详细描述配置MPM列表的方法。在以下描述的实施例中,假设定义了角度大于45度的10个广角帧内预测模式(-1至-10)和角度小于-135度的10个广角帧内预测模式(67至76)。

[0392] 当以广角帧内预测模式对相邻块进行编码时,可以基于与广角帧内预测模式相对应的广角替换帧内预测模式来设置MPM。例如,当相邻块以广角帧内预测模式进行编码时,可以将变量candIntraPredX(X为A或B)设置为广角代替帧内预测模式。

[0393] 可替代地,可以根据当前块的形状来确定导出MPM的方法。例如,在当前块是宽度和高度相同的正方形形状时,可以将candIntraPredX设置为广角替换帧内预测模式。另一方面,在当前块是非正方形形状时,可以将candIntraPredX设置为广角帧内预测模式。

[0394] 可替代地,可以基于相邻块的广角帧内预测模式是否适用于当前块来确定是否将candIntraPredX设置为广角帧内预测模式。例如,如果当前块是宽度大于高度的非正方形形状,则将索引大于右上对角线方向的帧内预测模式的索引的广角帧内预测模式原样设置为candIntraPredX,但对于索引小于左下对角线方向的帧内预测模式的索引的广角帧内预测模式,将与所述广角帧内预测模式相对应的广角替换帧内预测模式设置为candIntraPredX。另一方面,如果当前块是高度大于宽度的非正方形形状,则将索引小于左下对角线方向的帧内预测模式的索引的广角帧内预测模式原样设置为candIntraPredX,但对于索引大于右上对角线方向的帧内预测模式的索引的广角帧内预测模式,将与所述广角帧内预测模式相对应的广角替换帧内预测模式设置为candIntraPredX。

[0395] 即,根据在广角帧内预测模式下编码的相邻块的形状是否与当前块的形状相同或相似,可以确定是否使用广角帧内预测模式来导出MPM或是否使用广角替换帧内预测模式

来导出MPM。

[0396] 可替代地,可以将相邻块的广角帧内预测模式设置为candIntraPredX,而不管当前块的形状如何。

[0397] 总之,可以将candIntraPredX设置为相邻块的广角帧内预测模式或广角替换帧内预测模式。

[0398] 可以基于candIntraPredA和candIntraPredB导出MPM。在这种情况下,可以以类似于candIntraPredA或candIntraPredB的帧内预测模式导出MPM。可以基于模运算和偏移量来导出类似于candIntraPredA或candIntraPredB的帧内预测模式。在这种情况下,可以根据当前块的形状不同地确定用于模运算的常数和偏移量。

[0399] 表6示出了根据当前块的形状导出MPM的示例。

[0400] 【表6】

[0401]

```
i) 如果 cuWidth 等于 cuHeight
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)
ii) 否则 (即 cuWidth 不等于 cuHeight)
candModeList[0] = candIntraPredModeA
candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)
candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)
```

[0402] 假设candIntraPredA和candIntraPredB相同,并且candIntraPredA是角度帧内预测模式。在当前块是正方形形状时,可以根据基于从除广角帧内预测模式的角度帧内预测模式的总数中减去1而获得的值的模运算来获得类似于candIntraPredA的帧内预测模式。例如,当除广角帧内预测模式之外的角度帧内预测模式的数量是65时,可以根据基于candIntraPredA和64的模运算而得出的值来导出MPM。另一方面,在当前块是非正方形形状时,可以根据基于从包括广角帧内预测模式的角度帧内预测模式的总数中减去1而获得的值的模运算来获得类似于candIntraPredA的帧内预测模式。例如,当广角帧内预测模式的数量是20时,可以根据基于candIntrapredA和84的模运算得出的值来导出MPM。

[0403] 由于用于模运算的常数根据当前块的形状而不同地设置,因此可以确定是否可以将广角帧内预测模式设置为类似于candIntraPredA的角度帧内预测模式。例如,在使用64的模运算中,可能不能将广角帧内预测模式设置为类似于candIntraPredA的角度帧内预测模式,而在使用84的模运算中,可以将广角帧内预测模式设置为与candIntraPredA类似的角度帧内预测模式。

[0404] 可替代地,当candIntraPredA和candIntraPredB相同时,可以考虑当前块的形状以及candIntraPredA是否是广角帧内预测模式来导出MPM。

[0405] 表7示出了根据当前块的形状导出MPM的示例。

[0406] 【表7】

- 如果 candIntraPredModeB 等于 candIntraPredModeA, 则适用以下条件:  
 - 如果 candIntraPredModeA 小于 2 (即等于 INTRA\_PLANAR 或 INTRA\_DC), 则 x = 0 ... 2 的 candModeList[x] 导出如下:  
 candModeList[0] = INTRA\_PLANAR  
 candModeList[1] = INTRA\_DC  
 candModeList[2] = INTRA\_ANGULAR50  
 - 否则, x = 0...2 的 candModeList[x] 导出如下:  
 i) 如果 cuWidth 等于 cuHeight  
 1. 如果 candIntrapredA 小于 0 或 candIntraPredModeA 大于 66, 则  
 candModeList[0] = INTRA\_PLANAR  
 candModeList[1] = INTRA\_DC  
 candModeList[2] = INTRA\_ANGULAR50  
 2. 否则, 如果 candIntraPredA >= 0 && candIntraPredModeA < 67, 则  
 candModeList[0] = candIntraPredModeA  
 candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)  
 candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)  
 ii) 否则, 如果 cuWidth 大于 cuHeight  
 candModeList[0] = candIntraPredModeA  
 candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)  
 candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)  
 iii) 否则 (即, cuWidth 大于 cuHeight)  
 candModeList[0] = candIntraPredModeA  
 candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)  
 candModeList[2] = -8 + ((candIntraPredModeA - 1) % 84)  
 - 否则 (即, candIntraPredModeB 不等于 candIntraPredModeA), 适用以下条件:  
 - candModeList[0] 和 candModeList[1] 导出如下:  
 candModeList[0] = candIntraPredModeA  
 candModeList[1] = candIntraPredModeB  
 - 如果 candModeList[0] 和 candModeList[1] 都不等于 INTRA\_PLANAR, 则将 candModeList[2] 设置

为 INTRA\_PLANAR

- 否则, 如果 candModeList[0] 和 candModeList[1] 都不等于 INTRA\_DC, 则将 candModeList[2] 设置为等于 INTRA\_DC,  
 - 否则, 将 candModeList[2] 设置为 INTRA\_ANGULAR 50.

[0409] 假设 candIntraPredA 和 candIntraPredB 相同。

[0410] 在当前块是正方形形状并且 candIntraPredA 是广角帧内预测模式时, 可以将 MPM 设置为默认模式。例如, 可以将 MPM[0]、MPM[1] 和 MPM[2] 分别设置为平面模式、DC 模式和垂直帧内预测模式。

[0411] 在当前块是正方形形状并且 candIntraPredA 是除广角帧内预测模式之外的角度帧内预测模式时, 可以将 MPM 设置为类似于 candIntraPredA 的角度帧内预测模式。例如, 可以将 MPM[0] 设置为 candIntraPredA, 并且可以将 MPM[1] 和 MPM[2] 设置为类似于 candIntraPredA 的角度帧内预测模式。

[0412] 在当前块是非正方形形状的并且 candIntraPredA 是角度帧内预测模式时, 可以将 MPM 设置为类似于 candIntraPredA 的角度帧内预测模式。例如, 可以将 MPM[0] 设置为 candIntraPredA, 并且可以将 MPM[1] 和 MPM[2] 设置为类似于 candIntrapredA 的角度帧内预测模式。

[0413] 可以使用模运算和偏移量来导出类似于 candIntraPredA 的角度帧内预测模式。在这种情况下, 用于模运算的常数可能根据当前块的形状而有所不同。另外, 可以根据当前块的形状来不同地设置用于导出类似于 candIntraPredA 的角度帧内预测模式的偏移量。例如, 在当前块是宽度大于高度的非正方形形状时, 可以通过使用偏移量 2 来导出类似于

candIntraPredA的角度帧内预测模式。另一方面,在当前块是高度大于宽度的非正方形形状时,可以使用偏移量2和-8来导出类似于candIntraPredA的角度帧内预测模式。

[0414] 可替代地,可以通过考虑candIntraPredX是具有最大索引还是最小索引的广角帧内预测模式来导出MPM。

[0415] 表8示出了通过考虑广角帧内预测模式索引来导出MPM的示例。

[0416] 【表8】

|        |   |
|--------|---|
| [0417] | <ul style="list-style-type: none"> <li>- 如果 candIntraPredModeB 等于 candIntraPredModeA, 则适用以下条件:</li> <li>- 如果 candIntraPredModeA 小于 2 (即等于 INTRA_PLANAR 或 INTRA_DC), 则 x = 0 ... 2 的 candModeList[x] 导出如下:           <pre>candModeList[0] = INTRA_PLANAR candModeList[1] = INTRA_DC candModeList[2] = INTRA_ANGULAR50</pre> </li> <li>- 否则, x = 0...2 的 candModeList[x] 导出如下:           <ol style="list-style-type: none"> <li>如果 (candIntraPredModeA &lt; 0 &amp;&amp; candIntraPredModeA != -10), 则               <pre>candModeList[0] = candIntraPredModeA candModeList[1] = 2 + ((candIntraPredModeA - 3) % 84) candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)</pre> </li> <li>否则, 如果 (candIntraPredModeA = -10), 则               <pre>candModeList[0] = candIntraPredModeA candModeList[1] = -1 candModeList[2] = -9</pre> </li> <li>否则, 如果 (candIntraPredModeA &gt; 67 &amp;&amp; candIntraPredModeA &lt; 76), 则               <pre>candModeList[0] = candIntraPredModeA candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84) candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)</pre> </li> <li>否则, candIntraPredModeA 等于 67               <pre>candModeList[0] = candIntraPredModeA candModeList[1] = 66 candModeList[2] = 68</pre> </li> <li>否则, candIntraPredModeA 等于 77               <pre>candModeList[0] = candIntraPredModeA candModeList[1] = 76 candModeList[2] = 67</pre> </li> <li>否则, 如果 (candIntraPredModeA &gt;= 0 &amp;&amp; candIntraPredModeA &lt; 67), 则               <pre>candModeList[0] = candIntraPredModeA candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64) candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)</pre> </li> </ol> </li> <li>- 否则 (candIntraPredModeB 不等于 candIntraPredModeA), 适用以下条件:           <ul style="list-style-type: none"> <li>- candModeList[0] 和 candModeList[1] 导出如下:</li> </ul> </li> </ul> |
|--------|---|

|        |   |
|--------|---|
| [0418] | <pre>candModeList[0] = candIntraPredModeA candModeList[1] = candIntraPredModeB</pre> <ul style="list-style-type: none"> <li>- 如果 candModeList[0] 和 candModeList[1] 都不等于 INTRA_PLANAR, 则将 candModeList[2] 设置为 INTRA_PLANAR</li> <li>- 否则, 如果 candModeList[0] 和 candModeList[1] 都不等于 INTRA_DC, 则将 candModeList[2] 设置为等于 INTRA_DC,</li> <li>- 否则, 将 candModeList[2] 设置为 INTRA_ANGULAR 50.</li> </ul> |
|--------|---|

[0419] 假设candIntraPredA和candIntraPredB相同。为了便于说明,将索引值小于左下对角线方向的帧内预测模式的索引值的广角帧内预测模式称为向下广角帧内预测模式,并且将索引值大于右上对角线方向的帧内预测模式的索引值的广角帧内预测模式称为向右广角帧内预测模式。

[0420] 当candIntraPredA是向下广角帧内预测模式时,可以将MPM设置为类似于candIntraPredA的角度帧内预测模式。在这种情况下,当candIntraPredA是具有最小值的

向下广角帧内预测模式时,可以将MPM设置为具有预定义索引值的向下广角帧内预测模式。在此,预定义索引可以是在向下广角帧内预测模式的索引中具有最大值的索引。例如,当candIntraPredA为-10时,可以将MPM[0]、MPM[1]和MPM[2]分别设置为-10、-1和-9。

[0421] 当candIntraPredA是向右广角帧内预测模式时,可以将MPM设置为类似于candIntraPredA的角度帧内预测模式。在这种情况下,当candIntraPredA是具有最大值的向右广角帧内预测模式时,可以将MPM设置为具有预定义索引值的向右广角帧内预测模式。在此,预定义索引可以是在向右广角帧内预测模式的索引中具有最小值的索引。例如,当candIntraPredA为77时,可以将MPM[0]、MPM[1]和MPM[2]分别设置为77、76和67。

[0422] 可替代地,当通过对candIntraPredA的索引减1而得到的索引小于帧内预测模式的索引中的最小值或加1而得到的索引大于最大值时,可以将MPM设置为默认模式。在此,默认模式可以包括平面模式、DC模式、垂直帧内预测模式、水平帧内预测模式和对角帧内预测模式中的至少一种。

[0423] 可替代地,当通过将candIntraPredA的索引减1而得到的索引小于帧内预测模式的索引中的最小值或加1而得到的索引大于最大值时,可以将MPM设置为与candIntraPredA相反的帧内预测模式或设置为类似于与candIntraPredA相反的帧内预测模式的帧内预测模式。

[0424] 可替代地,可以考虑当前块的形状和相邻块的形状来导出MPM候选。例如,在当前块和相邻块都是非正方形形状的情况下导出MPM的方法可能与在当前块是正方形形状但相邻块是非正方形形状的情况下导出MPM的方法不同。

[0425] 可以考虑当前块的大小、当前块的形状、相邻块的大小和相邻块的形状中的至少一项来重新排列(或重新排序)MPM列表中的MPM。在此,重新排列表示重新分配被分配给每个MPM的索引。例如,可以将较小的索引分配给与具有与当前块的大小或形状相同的大小或形状的相邻块的帧内预测模式相同的MPM。

[0426] 假设将MPM[0]和MPM[1]分别设置为左相邻块的帧内预测模式candIntraPredA和上相邻块的帧内预测模式candIntraPredB。

[0427] 在当前块和上相邻块是宽度大于高度的非正方形形状时,可以将MPM重新排列为使得上相邻块的帧内预测模式candIntraPredB具有较小的索引。即,可以将candIntraPredB重新排列为MPM[0],并且可以将candIntraPredA重新排列为MPM[1]。

[0428] 可替代地,在当前块和上相邻块是高度大于宽度的非正方形形状时,可以将MPM重新排列为使得上相邻块的帧内预测模式candIntraPredB具有较小的索引。即,可以将candIntraPredB重新排列为MPM[0],并且可以将candIntraPredA重新排列为MPM[1]。

[0429] 可替代地,在当前块和上相邻块为正方形形状时,可以将MPM重新排列为使得上相邻块的帧内预测模式candIntraPredB具有较小的索引。即,可以将candIntraPredB重新排列为MPM[0],并且可以将candIntraPredA重新排列为MPM[1]。

[0430] 代替重新排列MPM,当最初将candIntraPredX分配给MPM时,可以考虑当前块的大小、当前块的形状、相邻块的大小和相邻块的形状中的至少一个。

[0431] 可以根据当前块的大小或形状重新排列MPM。例如,如果当前块是宽度大于高度的非正方形形状,则可以按降序顺序重新排列MPM。另一方面,如果当前块是高度大于宽度的非正方形形状,则可以按升序顺序重新排列MPM。

[0432] 可以通过将原始图像与预测图像相减来导出所导出的残差图像。在这种情况下，将残差图像变更为频域时，即使移除频率分量中的高频率分量，也不会大幅降低视频的主观画质。因此，如果将高频率分量的值转换为较小的值或者将高频率分量的值设置为0，则具有在不引起明显视觉失真的情况下提高压缩效率的效果。反映以上特性，可以对当前块进行变换以将残差图像分解为2维频率分量。可以使用离散余弦变换(DCT, Discrete Cosine Transform)或离散正弦变换(DST, Discrete Sine Transform)等变换技术来执行所述变换。

[0433] DCT使用余弦变换将残差图像分解(或变换)为2维频率分量，而DST使用正弦变换将残差图像分解(或变换)为2维频率分量。作为残差图像的变换结果，频率分量可以表示为基本图像。例如，当对具有 $N \times N$ 大小的块执行DCT变换时，可以获得 $N^2$ 个基本图案分量。可以通过变换来获得大小为 $N \times N$ 的块中包括的各个基本图案分量的大小。根据使用的变换技术，可以将基本图案分量的大小称为DCT系数或DST系数。

[0434] 变换技术DCT主要用于对非0的低频率分量分布较多的图像进行变换。变换技术DST主要用于高频率分量分布较多的图像。

[0435] 还可以使用DCT或DST以外的变换技术来变换残差图像。

[0436] 以下，将残差图像变换为2维频率分量的过程称为2维图像变换。另外，将通过变换而获得的基本图案分量的大小称为变换系数。例如，变换系数可指DCT系数或DST系数。当同时应用稍后要描述的主变换和二次变换时，变换系数可以表示通过二次变换的结果生成的基本图案分量的大小。

[0437] 可以以块为单位确定变换技术。可以基于当前块的预测编码模式、当前块的大小或当前块的形状中的至少一个来确定变换技术。例如，当以帧内预测模式对当前块进行编码并且当前块的大小小于 $N \times N$ 时，可以使用变换技术DST执行变换。另一方面，在不能满足所述条件时，可以使用变换技术DCT执行变换。

[0438] 在残差图像中，还可以对一部分块不执行2维图像变换。不执行2维图像变换可称为变换跳过(Transform Skip)。当应用变换跳过时，可以将量化应用于未执行变换的残差值。

[0439] 在使用DCT或DST对当前块进行变换之后，可以再次对变换的当前块进行变换。在这种情况下，可以将基于DCT或DST的变换定义为主变换，并且将对应用主变换的块再次进行变换的过程称为二次变换。

[0440] 主变换可以使用多个变换核候选中的任一个来执行。例如，可以使用DCT2、DCT8或DCT7中的任一个来执行主变换。

[0441] 针对水平方向和垂直方向还可以使用不同的变换核。还可以通过比特流用信号发送表示水平方向的变换核和垂直方向的变换核的组合的信息。

[0442] 主变换和二次变换的执行单元会不同。例如，可以对 $8 \times 8$ 块执行主变换，并且可以对变换的 $8 \times 8$ 块中的大小为 $4 \times 4$ 的子块执行二次变换。在这种情况下，还可以将不执行二次变换的剩余区域的变换系数设置为0。

[0443] 可替代地，可以对 $4 \times 4$ 块执行主变换，并且可以对包括变换的 $4 \times 4$ 块的大小为 $8 \times 8$ 的区域可执行二次变换。

[0444] 可以通过比特流用信号发送表示是否执行二次变换的信息。

[0445] 可以在解码器中执行二次变换的逆变换(第二逆变换),并且可以对其结果执行主变换的逆变换(第一逆变换)。作为第二逆变换和第一逆变换的执行结果,可以获得当前块的残差信号。

[0446] 量化用于减少块的能量,并且量化过程包括将变换系数除以特定常数的过程。所述常数可以由量化参数来导出,并且量化参数可以定义为1至63之间的值。

[0447] 若在编码器中执行变换和量化,则解码器可以通过逆量化和逆变换来获得残差块。解码器可以通过将预测块和残差块相加来获得当前块的重建块。

[0448] 若获得当前块的重建块,则可以通过环内滤波(In-loop filtering)来减少在量化和编码过程中产生的信息丢失。环内滤波器可以包括去块滤波器(Deblocking filter)、样本自适应偏移滤波器(Sample Adaptive Offset filter, SAO)或自适应环路滤波器(Adaptive Loop Filter, ALF)中的至少一个。以下,将应用环内滤波器之前的重建块称为第一重建块,并且将应用环内滤波器之后的重建块称为第二重建块。

[0449] 可以对第一重建块应用去块滤波器、SAO或ALF中的至少一个来获得第二重建块。在这种情况下,可以在应用去块滤波器之后应用SAO或ALF。

[0450] 去块滤波器用于缓解通过以块为单位执行量化时在块的边界中产生的画质劣化(块效应, Blocking Artifact)。为了应用去块滤波器,可以确定第一重建块与相邻重建块之间的块强度(Blocking Strength, BS)。

[0451] 图30是示出确定块强度的过程的流程图。

[0452] 如图30所示的示例中,P表示第一重建块,并且Q表示相邻重建块。其中,相邻重建块可以与当前块的左侧或上方相邻。

[0453] 在图30所示的示例中,示出了考虑P和Q的预测编码模式、是否包括非零变换系数、是否使用相同的参考图像执行帧间预测、以及运动矢量的差值是否大于或等于阈值来确定块强度。

[0454] 基于块强度,可以确定是否应用了去块滤波器。例如,当块强度为0时,可能不执行滤波。

[0455] SAO用于缓解在频域中执行量化时产生的振铃效应(Ringing Artifact)。可以通过加上或减去考虑第一重建图像的图案而确定的偏移量来执行SAO。偏移量的确定方法包括边偏移(Edge Offset, EO)或边带偏移(Band Offset)。EO表示根据周边像素的图案来确定当前样本的偏移的方法。BO表示对区域内具有类似的亮度值的像素的集合应用共同的偏移的方法。具体地,可以将像素亮度区分为32个均等的区间,并且可以将具有类似的亮度值的像素设置为一个集合。例如,可以将32个带中相邻的4个带设置为一组,并且可以对属于4个带的样本应用相同的偏移量。

[0456] ALF是对第一重建图像或应用去块滤波器的重建图像应用预定义的大小/形状的滤波器来生成第二重建图像的方法。以下等式21示出了ALF的应用示例。

[0457] 【等式21】

$$[0458] R'(i, j) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} \sum_{l=-\frac{N}{2}}^{\frac{N}{2}} f(k, l) \cdot R(i + k, j + l)$$

[0459] 可以以图像、编码树单元、编码块、预测块或变换块为单位来选择预定义的滤波器候选中的任一个。每个滤波器候选的大小或形状中的任一个可以不同。

[0460] 图31示出了预定义的滤波器候选的图。

[0461] 如在图31所示的示例中,可以选择 $5 \times 5$ 、 $7 \times 7$ 和 $9 \times 9$ 菱形中的至少一种。

[0462] 针对色度分量只能使用大小为 $5 \times 5$ 的菱形。

[0463] 为了对诸如全景视频、360度视频或4K/8K UHD(超高清)等高分辨率视频进行实时或低延迟编码,可以考虑将一张图像划分为多个区域并且并行地对所述多个区域进行编码/解码。为此,可以将图像划分为图块(即,并行编码/解码的基本单元),并且可以并行处理这些图块。

[0464] 可以将图块约束为具有矩形形状。在对图块进行编码/解码时,不使用其他图块的数据。可以以图块为单位初始化上下文自适应二进制算术编码(CABAC)上下文的概率表,并且可以将其设置为不在图块的边界处应用环路滤波器。

[0465] 图32示出了将图像划分为多个图块的示例。

[0466] 图块包括至少一个编码树单元,并且图块的边界与编码树单元的边界重合。

[0467] 如在图32中所示的示例中,图像可以被划分为多个图块集。可以通过比特流用信号发送用于将图像划分为多个图块集的信息。

[0468] 根据图像的划分类型,图块可以在除了图像边界之外的所有区域中具有相同的小。

[0469] 可替代地,可以将图像划分为在水平方向上彼此相邻的图块具有相同的高度,或者可以将图像划分为在垂直方向上彼此相邻的图块具有相同的宽度。

[0470] 当通过使用与图像相交的垂直线或水平线中的至少一条线来划分图像时,每个图块都属于不同的列和/或行。在以下描述的示例性实施例中,图块所属的列被称为图块列,并且图块所属的行被称为图块行。

[0471] 可以通过比特流用信号发送用于确定将图像划分为图块的形状的信息。可以通过图像参数集或序列参数集对信息进行编码并用信号发送。该信息用于确定图像中的图块的数量,并且可以包括指示图块行的数量的信息和指示图块列的数量的信息。例如,语法元素num\_tile\_columns\_minus1指示通过从图块列的数量减去1而获得的值,并且语法元素num\_tile\_rows\_minus1指示通过从图块行的数量减去1而获得的值。

[0472] 在图32所示的示例中,由于图块列的数量是4并且图块行的数量是3,所以num\_tile\_columns\_minus1可以为3,并且num\_tile\_rows\_minus1可以为2。

[0473] 当将图像划分为多个图块时,可以通过比特流用信号发送指示图块大小的信息。例如,当将图像划分为多个图块列时,通过比特流用信号发送指示每个图块列的宽度的信息,并且当将图像划分为多个图块行时,通过比特流用信号发送指示每个图块行的高度的信息。例如,对于每个图块列,可以对指示图块列的宽度的语法元素column\_width\_minus1进行编码并用信号发送,并且对于每个图块行,可以对指示图块行的高度的语法元素row\_height\_minus1进行编码并用信号发送。

[0474] column\_width\_minus1可以指示通过从图块列的宽度减去1而获得的值,并且row\_height\_minus1可以指示通过从图块行的高度减去1而获得的值。

[0475] 对于最后一个图块列,可以省略对column\_width\_minus1的编码,并且对于最后一

个图块行,可以省略对row\_height\_minus1的编码。可以考虑图像大小来导出最后一个图块列的宽度和最后一行的高度。

[0476] 解码器可以基于column\_width\_minus1和row\_height\_minus1确定图块的大小。

[0477] 表9示出了用于将图像划分为图块的语法表。

[0478] 【表9】

|        |   |       |
|--------|---|-------|
| [0479] | ...   |       |
|        | if( tile_enabled_flag ) {                     |       |
|        | num_tile_columns_minus1                       | ue(v) |
|        | num_tile_rows_minus1                          | ue(v) |
|        | uniform_spacing_flag                          | u(1)  |
|        | if( !uniform_spacing_flag ) {                 |       |
|        | for ( i=0; i < num_tile_columns_minus1; i++ ) |       |
|        | column_width_minus1[i]                        | ue(v) |
|        | for ( i=0; i < num_tile_rows_minus1; i++ )    |       |
|        | row_height_minus1[i]                          | ue(v) |
|        | }   |       |
|        | loop_filter_across_tiles_enabled_flag         | u(1)  |
|        | }   |       |
|        | ...   |       |

[0481] 参照表9,可以用信号发送指示图块列的数量的语法元素num\_tile\_columns\_minus1和指示图块行的数量的语法元素num\_tile\_rows\_minus1。

[0482] 接下来,可以用信号发送指示图像是否被划分为相等大小的图块的语法元素uniform\_spacing\_flag。当uniform\_spacing\_flag为真时,可以将除图像边界之外的其余区域中的图块划分为相等的大小。

[0483] 当uniform\_spacing\_flag为假时,可以用信号发送指示每个图块列的宽度的语法元素column\_width\_minus1和指示每个图块行的高度的语法元素row\_height\_minus1。

[0484] 语法元素loop\_filter\_across\_tiles\_enabled\_flag指示是否允许在图块边界处使用环路滤波器。

[0485] 在图块列中具有最小宽度的图块列可以被称为最小宽度图块,并且在图块行中具有最小高度的图块行可以被称为最小高度图块。可以通过比特流用信号发送指示最小宽度图块的宽度的信息和指示最小高度图块的高度的信息。例如,语法元素min\_column\_width\_minus1指示从最小宽度图块的宽度减去1所获得的值,并且语法元素min\_row\_height\_minus1指示从最小高度图块的高度减去1所获得的值。

[0486] 对于每个图块列,可以用信号发送指示与最小图块宽度的差值的信息。例如,语法元素diff\_column\_width指示当前图块列与最小图块列之间的宽度差值。宽度差值可以表示为编码树单元列的数量的差值。解码器可以通过将基于min\_column\_width\_minus1导出的最小宽度图块的宽度与基于diff\_column\_width导出的宽度差值相加来导出当前图块的宽度。

[0487] 另外,对于每个图块行,可以用信号发送指示与最小图块高度的差值的信息。例如,语法元素diff\_row\_height指示当前图块行与最小图块行之间的高度差值。高度差值可以表示为编码树单元行的数量的差值。解码器可以通过将基于min\_row\_height\_minus1导出的最小高度图块的高度与基于diff\_row\_height导出的高度差值相加来导出当前图块的

高度。

[0488] 表10示出了包括关于大小差异的信息的语法表。

[0489] 【表10】

|        |  |       |
|--------|--|-------|
|        | ...  |       |
|        | if( tile_enabled_flag ) {                    |       |
|        | num_tile_columns_minus1                      | ue(v) |
|        | num_tile_rows_minus1                         | ue(v) |
|        | uniform_spacing_flag                         | u(1)  |
|        | if( !uniform_spacing_flag ) {                |       |
|        | min_column_width_minus1                      |       |
|        | min_row_height_minus1                        |       |
| [0490] | for( i=0; i < num_tile_columns_minus1; i++ ) |       |
|        | diff_column_width[i]                         | ue(v) |
|        | for( i=0; i < num_tile_rows_minus1; i++ )    |       |
|        | diff_row_height[i]                           | ue(v) |
|        | }  |       |
|        | loop_filter_across_tiles_enabled_flag        | u(1)  |
|        | }  |       |
|        | ...  |       |

[0491] 可以将图像划分为使得与水平相邻的图块具有不同的高度,或者可以将图像划分为使得与垂直相邻的图块具有不同的宽度。以上图像划分方法可以被称为灵活图块(Flexible Tile)划分方法,并且通过灵活图块划分方法划分的图块可以被称为灵活图块。

[0492] 图33是示出根据灵活图块技术的图像划分模式的图。

[0493] 通过划分图像而生成的图块的搜索顺序可以遵循预定的扫描顺序。另外,可以根据预定的扫描顺序将索引分配给每个图块。

[0494] 图块的扫描顺序可以是光栅扫描、对角线扫描、垂直扫描或水平扫描中的任何一种。图33(a)至图33(d)示出了分别根据光栅扫描、对角线扫描、垂直扫描和水平扫描将索引分配给每个图块的示例。

[0495] 可以根据当前图块的大小或位置确定下一个扫描顺序。例如,当当前图块的高度和与当前图块的右侧相邻的图块的高度不同时(例如,当右相邻图块的高度大于当前图块的高度时),在与同当前图块的下方相邻的图块的垂直线相同的垂直线上的图块中位于最左侧的图块可以被确定为当前图块之后的扫描目标。

[0496] 可以以图像或序列为单位确定图块的扫描顺序。

[0497] 可替代地,可以考虑图像中第一图块的大小来确定图块的扫描顺序。例如,当第一图块的宽度大于高度时,可以将图块的扫描顺序设置为水平扫描。当第一图块的高度大于宽度时,可以将图块的扫描顺序设置为垂直扫描。当第一图块的宽度与高度相同时,可以将图块的扫描顺序设置为光栅扫描或对角线扫描。

[0498] 可以通过比特流用信号发送指示图块的总数的信息。例如,当应用灵活图块技术时,可以用信号发送通过从图像中的图块总数减去2而导出的语法元素number\_of\_tiles\_in\_picture\_minus2。解码器可以基于number\_of\_tiles\_in\_picture\_minus2来识别当前图像中包括的图块数量。

[0499] 表11示出了包括关于图块数量的信息的语法表。

[0500] 【表11】

[0501]

|  |  | 描述符   |
|--|--|-------|
|  | pic_parameter_set_rbsp( ) {                                  |       |
|  | ...  |       |
|  | <b>tiles_enabled_flag</b>                                    | u(1)  |
|  | ...  |       |
|  | if( tiles_enabled_flag ) {                                   |       |
|  | <b>number_of_tiles_in_picture_minus2</b>                     | ue(v) |
|  | <b>subtile_width_minus1</b>                                  | ue(v) |
|  | <b>subtile_height_minus1</b>                                 | ue(v) |
|  | for ( i=0; i < number_of_tiles_in_picture_minus2+2 ; i++ ) { |       |
|  | if ( i > 0 )   |       |
|  | <b>use_previous_tile_size_flag</b>                           | u(1)  |
|  | if ( use_previous_tile_size_flag == 0 ) {                    |       |
|  | <b>tile_width_minus1[ i ]</b>                                | ue(v) |
|  | <b>tile_height_minus1[ i ]</b>                               | ue(v) |
|  | }  |       |
|  | }  |       |
|  | <b>loop_filter_across_tiles_enabled_flag</b>                 | u(1)  |
|  | }  |       |
|  | ...  |       |

[0502] 为了减少对图块的大小进行编码所需的比特数,可以对指示子图块的大小的信息进行编码并用信号发送。子图块是构成图块的基本单元,并且每个图块可以被配置为包括至少一个子图块。子图块可以包括一个或多个编码树单元。

[0503] 例如,语法元素subtile\_width\_minus1指示通过从子图块的宽度减去1而获得的值。语法元素subtile\_height\_minus1指示通过从子图块的高度减去1而获得的值。

[0504] 可以对指示除第一图块之外的其余图块是否具有与前一个图块相同的大小的信息进行编码并用信号发送。例如,语法元素use\_previous\_tile\_size\_flag指示当前图块的大小是否与前一个图块的大小相同。当use\_previous\_tile\_size\_flag为真时,指示当前图块的大小与前一个图块的大小相同。当use\_previous\_tile\_size\_flag为假时,可以对指示当前图块的大小的信息进行编码并用信号发送。对于第一个图块,可以省略对use\_previous\_tile\_size\_flag的编码,并且可以将标志的值设置为假。

[0505] 指示图块大小的信息可以包括指示第i个图块的宽度的语法元素tile\_width\_minus1[i]和指示第i个图块的高度的语法元素tile\_height\_minus1[i]。

[0506] 指示图块大小的信息可以指示与子图块的大小的差值。当使用子图块的大小信息时,可以通过减少对每个图块的大小进行编码所需的比特数来提高编码/解码效率。例如,可以基于以下等式22导出第i个图块的宽度tileWidth,并且可以基于以下等式23导出第i个图块的高度tileHeight。

[0507] 【等式22】

[0508]  $tileWidth = (subtile_width_minus1+1) * (tile_width_minus1[i]+1)$

[0509] 【等式23】

[0510]  $tileHeight = (subtile_height_minus1+1) * (tile_height_minus1[i]+1)$

[0511] 可替代地,可以省略对子图块的大小信息的编码,并且可以将第i个图块的大小按原样编码为图块大小信息。可以可选地对子图块的大小信息进行编码。可以通过视频参数集、序列参数集或图像参数集来用信号发送指示子图块的大小信息是否被编码的信息。

[0512] 与图块大小有关的信息可以被编码为指示编码树单元的数量并用信号发送。例

如, column\_width\_minus1、min\_column\_width\_minus1、subtile\_width\_minus1、tile\_width\_minus1等可以指示图块中包括的编码树单元列的数量。另外,diff\_column\_width可以指示最小宽度图块中包括的编码树单元列的数量与当前图块中包括的编码树单元列的数量之间的差值。

[0513] 另外, row\_height\_minus1、min\_row\_height\_minus1、subtile\_height\_minus1、tile\_height\_minus1等可以指示图块中包括的编码树单元行的数量。此外,diff\_row\_height可以指示最小高度图块中包括的编码树单元行的数量与当前图块中包括的编码树单元行的数量之间的差值。

[0514] 解码器可以根据基于语法元素导出的编码树单元列的数量和/或编码树单元行的数量、以及编码树单元的大小来确定图块的大小。例如,第*i*个图块的宽度可以被设置为 $(tile\_width\_minus1[i]+1) * (编码树单元的宽度)$ ,并且第*i*个图块的高度可以被设置为 $(tile\_height\_minus1[i]+1) * (编码树单元的高度)$ 。

[0515] 同时,可以通过序列参数集或图像参数集来用信号发送指示编码树单元的大小的信息。

[0516] 在表11中,描述了使用语法元素use\_previous\_tile\_size\_flag来指示当前图块的大小是否与前一个图块的大小相同。作为另一示例,可以对指示当前图块的宽度是否与前一个图块的宽度相同的信息或指示当前图块的高度是否与前一个图块的高度相同的信息进行编码并用信号发送。

[0517] 表12示出了包括指示当前图块的宽度是否与前一各图块的宽度相同的信息的语法表。

[0518] 【表12】

|  |       | 描述符 |
|--|-------|-----|
| pic_parameter_set_rbsp( ) {                          |       |     |
| ...  |       |     |
| <b>tiles_enabled_flag</b>                            | u(1)  |     |
| ...  |       |     |
| CoveredWidthByTile = 0;                              |       |     |
| CoveredHeightByTile = 0 ;                            |       |     |
| PrevWidthByTile = 0;                                 |       |     |
| PrevHeightByTile = 0 ;                               |       |     |
| if( tiles_enabled_flag ) {                           |       |     |
| <b>number_of_tiles_in_picture_minus2</b>             | ue(v) |     |
| for ( i=0; i < number_of_tiles_in_picture_minus2+2 ; |       |     |
| i++ ) {  |       |     |
| if( i > 0 )  |       |     |
| <b>use_previous_tile_size_flag</b>                   | u(1)  |     |
| if( use_previous_tile_size_flag == 0 ) {             |       |     |
| <b>use_previous_tile_width_flag</b>                  | ue(v) |     |
| if( use_previous_tile_width_flag == 0 ) {            | ue(v) |     |
| <b>tile_width_minus1[ i ]</b>                        | ue(v) |     |
| <b>tile_height_minus1[ i ]</b>                       | ue(v) |     |
| } else {   |       |     |
| <b>tile_height_minus1[ i ]</b>                       | ue(v) |     |
| }  |       |     |
| }  |       |     |
| <b>loop_filter_across_tiles_enabled_flag</b>         | u(1)  |     |
| }  |       |     |
| ...  |       |     |

[0519] [0520] 语法元素use\_previous\_tile\_width\_flag指示当前图块的宽度是否与前一个图块的宽度相同。当use\_previous\_tile\_width\_flag为真时,当前图块的宽度可以设置为等于前一个图块的宽度。在这种情况下,可以省略对指示当前图块的宽度的信息的编码,并且可以从一个图块的宽度导出当前图块的宽度。

[0521] 当use\_previous\_tile\_width\_flag为假时,可以用信号发送指示当前图块的宽度的信息。例如, tile\_width\_minus1[i]可以指示通过从第i个图块的宽度减去1而获得的值。

[0522] 仅当确定当前图块的大小不同于前一个图块的大小时(例如,当use\_previous\_tile\_size\_flag的值是0时),才可能对语法元素use\_previous\_tile\_width\_flag进行编码并用信号发送。

[0523] tile\_width\_minus1[i]可以具有通过从第i个图块中包括的编码树单元列的数量减去1而获得的值。解码器可以通过将tile\_width\_minus1[i]加1来导出属于第i个图块的编码树单元的列数,并将导出的值乘以编码树单元的宽度来计算图块宽度。

[0524] 表13示出了进一步包括指示当前图块的高度是否与前一个图块的高度相同的信息的语法表。

[0525] 【表13】

|  | 描述符   |
|--|-------|
| pic_parameter_set_rbsp( ) {                          |       |
| ...  |       |
| <b>tiles_enabled_flag</b>                            | u(1)  |
| ...  |       |
| CoveredWidthByTile = 0;                              |       |
| CoveredHeightByTile = 0 ;                            |       |
| PrevWidthByTile = 0;                                 |       |
| PrevHeightByTile = 0 ;                               |       |
| if( tiles_enabled_flag ) {                           |       |
| <b>number_of_tiles_in_picture_minus2</b>             | ue(v) |
| for ( i=0; i < number_of_tiles_in_picture_minus2+2 ; |       |
| i++ ) {  |       |
| if( i > 0 )  |       |
| <b>use_previous_tile_size_flag</b>                   | u(1)  |
| if( use_previous_tile_size_flag == 0 ) {             |       |
| <b>use_previous_tile_width_flag</b>                  | ue(v) |
| if( use_previous_tile_width_flag == 0 ) {            | ue(v) |
| <b>tile_width_minus1[ i ]</b>                        | ue(v) |
| tile_width_minus1[i] +=1;                            |       |
| <b>use_previous_tile_height_flag</b>                 |       |
| if( use_previous_tile_height_flag ==0){              |       |
| <b>tile_height_minus1[ i ]</b>                       | ue(v) |
| tile_height_minus1[i] +=1;                           |       |
| }  |       |
| }  |       |
| }  |       |
| <b>use_previous_tile_height_flag</b>                 |       |
| if( use_previous_tile_height_flag ==0){              |       |
| <b>tile_height_minus1[ i ]</b>                       | ue(v) |
| tile_height +=1;                                     |       |
| }  |       |
| }  |       |
| }  |       |
| <b>loop_filter_across_tiles_enabled_flag</b>         | u(1)  |
| }  |       |
| ...  |       |

[0526]

[0527] 语法规元素use\_previous\_tile\_height\_flag指示当前图块的高度是否与前一个图块的高度相同。当use\_previous\_tile\_height\_flag为真时,当前图块的高度可以设置为等于前一个图块的高度。在这种情况下,可以省略对指示当前图块的高度的信息的编码,并且可以从前一个图块的高度导出当前图块的高度。

[0528] 当use\_previous\_tile\_height\_flag为假时,可以用信号发送指示当前图块的高度的信息。例如, tile\_height\_minus1[i] 可以指示通过从第i个图块的高度减去1而获得的值。

[0529] 仅当确定当前图块的大小不同于前一个图块的大小时(例如,当use\_previous\_tile\_size\_flag的值为0时),才可能对语法规元素use\_previous\_tile\_height\_flag进行编码并用信号发送。另外,仅在use\_previous\_tile\_width\_flag为假时才用信号发送语法规元素use\_previous\_tile\_height\_flag。

[0530] 表12示出了使用use\_previous\_tile\_width\_flag的示例,并且表13示出了使用use\_previous\_tile\_width\_flag和use\_previous\_tile\_height\_flag的示例。尽管在上表中未示出,但可以省略对use\_previous\_tile\_width\_flag的编码,而可以仅使用use\_

previous\_tile\_height\_flag。

[0531] 可以基于图块扫描顺序、第一图块的宽度和高度、以及前一个图块的宽度和高度中的至少一者来确定使用use\_previous\_tile\_height\_flag和use\_previous\_tile\_size\_flag中的哪一个。例如，当图块扫描顺序为垂直方向时，可以使用use\_previous\_tile\_height\_flag，而当图块扫描顺序为水平方向时，可以使用use\_previous\_tile\_width\_flag。可替代地，当第一图块或前一个图块是宽度大于高度的非正方形形状时，可以使用use\_previous\_tile\_width\_flag，而当第一图块或前一个图块是高度大于宽度的非正方形形状时，可以使用use\_previous\_tile\_height\_flag。

[0532] 在用信号发送图像中包括的图块数量时，可以针对最后一个图块省略对与图块大小有关的信息的编码。

[0533] 表14示出了针对最后一个图块省略对图块大小信息的编码的示例。

[0534] 【表14】

|        | pic_parameter_set_rbsp() {                           | 描述符   |
|--------|--|-------|
|        | ...  |       |
|        | <b>tiles_enabled_flag</b>                            | u(1)  |
|        | ...  |       |
|        | if( tiles_enabled_flag ) {                           |       |
|        | <b>number_of_tiles_in_picture_minus2</b>             | ue(v) |
|        | <b>subtile_width_minus1</b>                          | ue(v) |
|        | <b>subtile_height_minus1</b>                         | ue(v) |
|        | for ( i=0; i < number_of_tiles_in_picture_minus2+1 ; |       |
| [0535] | i++ ) {  |       |
|        | if( i > 0 )  |       |
|        | <b>use_previous_tile_size_flag</b>                   | u(1)  |
|        | if( use_previous_tile_size_flag == 0 ) {             |       |
|        | <b>tile_width_minus1[ i ]</b>                        | ue(v) |
|        | <b>tile_height_minus1[ i ]</b>                       | ue(v) |
|        | }  |       |
|        | }  |       |
|        | <b>loop_filter_across_tiles_enabled_flag</b>         | u(1)  |
|        | }  |       |
|        | ...  |       |

[0536] 当指定了除最后一个图块之外的图块大小时，可以将图像中的剩余区域设置为最后一个图块。

[0537] 对于每个编码树单元，可以分配用于识别编码树单元所属的图块的标识符（在下文中，称为图块ID，TileID）。

[0538] 图34是示出将图块ID分配给每个编码树单元的示例的图。

[0539] 可以将相同的图块ID分配给属于同一个图块的编码树单元。具体地，可以将第N个TileID分配给属于图块N的编码树单元。

[0540] 为了确定分配给每个编码树单元的图块ID，可以确定指示图像中的编码树单元的位置的变量x和y。在此，x表示通过将编码树单元的左上样本的位置(x0,y0)的x轴坐标除以编码树单元的宽度获得的值，并且y表示通过将编码树单元的左上样本的位置(x0,y0)的y轴坐标除以编码树单元的高度获得的值。具体地，可以通过以下等式24和25来导出x和y。

[0541] 【等式24】

[0542]  $x = (x0 / (\text{CTU宽度}))$

[0543] 【等式25】

[0544]  $Y = (y0 / (\text{CTU高度}))$

[0545] 可以通过以下描述的过程来执行将图块ID分配给每个编码树单元的操作。

[0546] i) 初始化图块ID

[0547] 每个编码树单元的图块ID可以被初始化为通过从图像中的图块数量减去1而获得的值。

[0548] 【表15】

如果 tile\_enabled\_flag 等于 1，则如下指定变量 tile\_id 的值和二维阵列 TileId 的值：

```
tile_id = 0
for (y=0; y<PicHeightInCtbsY; y++)
- for (x=0; x<PicWidthInCtbsY; x++)
-- TileId[x][y] = number_of_tiles_in_picture_minus2+1
```

```
tile_height_minus1[number_of_tiles_in_picture_minus2+1] = 0
tile_width_minus1[number_of_tiles_in_picture_minus2+1] = 0
```

[0550] ii) 导出图块ID

[0551] 【表16】

如下导出变量 tile\_id 的值和二维阵列 CTU\_tile 分配的值：

```
for(ctu_y=0; ctu_y < (tile_height_minus1[i]+1)*(subtile_height_minus1+1) && (tile_coordinate_y+ctu_y) <
PicHeightInCtbsY; ctu_y++)
- for(ctu_x=0; ctu_x < (tile_width_minus1[i]+1)*(subtile_width_minus1+1) && (tile_coordinate_x+ctu_x) <
PicWidthInCtbsY; ctu_x++)
```

```
-- TileId[tile_coordinate_x + ctu_x] [tile_coordinate_y + ctu_y] = tile_id
tile_id++
```

[0554] 在上述实施例中，已经描述了通过图像参数集用信号发送指示是否允许在图块的边界处应用环路滤波器的标志。然而，如果设置为在所有图块边界处都不使用环路滤波器，则可能出现主观图像质量下降并且编码效率下降的问题。

[0555] 因此，可以对指示每个图块是否允许应用环路滤波器的信息进行编码并用信号发送。

[0556] 图35是示出选择性地确定是否对每个图块应用环路滤波器的示例的图。

[0557] 如在图35中所示的示例中，可以针对每个图块确定在水平或垂直边界处是否允许应用环路滤波器（例如，去块滤波器、SAO和/或ALF）。

[0558] 表17示出了针对每个图块对指示是否允许应用环路滤波器的信息进行编码的示例。

[0559] 【表17】

|        | pic_parameter_set_rbsp( ) {                          | 描述符   |
|--------|--|-------|
| [0560] | ...  |       |
|        | <b>tiles_enabled_flag</b>                            | u(1)  |
|        | ...  |       |
|        | if( tiles_enabled_flag ) {                           |       |
|        | <b>number_of_tiles_in_picture_minus2</b>             | ue(v) |
|        | <b>subtile_width_minus1</b>                          | ue(v) |
|        | <b>subtile_height_minus1</b>                         | ue(v) |
|        | for ( i=0; i < number_of_tiles_in_picture_minus2+1 ; |       |
|        | i++ ) {  |       |
|        | if ( i > 0 )   |       |
|        | <b>use_previous_tile_size_flag</b>                   | u(1)  |
|        | if ( use_previous_tile_size_flag == 0 ) {            |       |
|        | <b>tile_width_minus1[ i ]</b>                        | ue(v) |
|        | <b>tile_height_minus1[ i ]</b>                       | ue(v) |
|        | }  |       |
|        | <b>loop_filter_across_tiles_flag[i]</b>              | u(1)  |
|        | }  |       |
|        | }  |       |
|        | ...  |       |

[0561] 在表17的示例中,语法元素loop\_filter\_across\_tiles\_flag[i]指示是否允许将环路滤波器应用于第i个图块。loop\_filter\_across\_tile\_flag[i]的值为1指示可以在图块ID为i的图块的水平边界和垂直边界处使用环路滤波器。loop\_filter\_across\_tile\_flag[i]的值为0指示不在图块ID为i的图块的水平边界和垂直边界处使用环路滤波器。

[0562] 可以对指示在水平方向和垂直方向的每个方向上是否允许应用环路滤波器的信息进行编码。

[0563] 表18示出了分别在水平方向和垂直方向上对指示是否允许应用环路滤波器的信息进行编码的示例。

[0564] 【表18】

|        | pic_parameter_set_rbsp( ) {                          | 描述符   |
|--------|--|-------|
| [0565] | ...  |       |
|        | <b>tiles_enabled_flag</b>                            | u(1)  |
|        | ...  |       |
|        | if( tiles_enabled_flag ) {                           |       |
|        | <b>number_of_tiles_in_picture_minus2</b>             | ue(v) |
|        | <b>subtile_width_minus1</b>                          | ue(v) |
|        | <b>subtile_height_minus1</b>                         | ue(v) |
|        | for ( i=0; i < number_of_tiles_in_picture_minus2+1 ; |       |
|        | i++ ) {  |       |
|        | if ( i > 0 )   |       |
|        | <b>use_previous_tile_size_flag</b>                   | u(1)  |
|        | if ( use_previous_tile_size_flag == 0 ) {            |       |
|        | <b>tile_width_minus1[ i ]</b>                        | ue(v) |
|        | <b>tile_height_minus1[ i ]</b>                       | ue(v) |
|        | }  |       |
|        | <b>loop_filter_hor_across_tiles_flag[i]</b>          | u(1)  |
|        | <b>loop_filter_ver_across_tiles_flag[i]</b>          | u(1)  |
|        | }  |       |
|        | }  |       |
|        | ...  |       |

|        |     |  |
|--------|-----|--|
| [0566] | {   |  |
|        | }   |  |
|        | ... |  |

[0567] 在表18的示例中,语法元素loop\_filter\_hor\_across\_tiles\_flag[i]指示是否允许在水平方向上与第i个图块交叉的位置处应用环路滤波器。语法元素loop\_filter\_ver\_across\_tiles\_flag[i]指示是否允许在垂直方向上与第i个图块交叉的位置处应用环路滤波器。

[0568] loop\_filter\_hor\_across\_tile\_flag[i]的值为1指示可以在图块ID为i的图块的水平边界处使用环路滤波器。loop\_filter\_hor\_across\_tile\_flag[i]的值为0指示不在图块ID为i的图块的垂直边界处使用环路滤波器。

[0569] loop\_filter\_ver\_across\_tile\_flag[i]的值为1指示可以在图块ID为i的图块的垂直边界处使用环路滤波器,并且loop\_filter\_ver\_across\_tile\_flag[i]的值为0指示不在图块ID为i的图块的垂直边界处使用环路滤波器。

[0570] 可替代地,可以对指示包括多个图块的图块组是否允许应用环路滤波器的信息进行编码并用信号发送。可以基于该信息来确定图块组中包括的多个图块是否允许应用环路滤波器。

[0571] 为了确定图块组,可以通过比特流用信号发送属于所述图块组的图块的数量、图块组的大小和图像的划分信息中的至少一者。可替代地,可以将编码器和解码器中的预定义大小的区域设置为图块组。

[0572] 可以省略对指示是否允许应用环路滤波器的信息的编码,并且可以基于图块中包括的编码树单元的数量、图块的宽度和图块的高度中的至少一者来确定是否允许应用环路滤波器。例如,当图块宽度小于参考值时,可以允许在水平方向上应用环路滤波器,而当图块高度小于参考值时,可以在垂直方向上应用环路滤波器。

[0573] 当在图块边界处使用环路滤波器时,可以基于图块中包括的数据来生成图块外部的重建数据。在这种情况下,通过对图块中包括的数据进行填充或插值来获得图块外部的重建视频。此后,可以通过使用图块外部的重建数据来应用环路滤波器。

[0574] 将集中于解码过程或编码过程描述的实施例用于编码过程或解码过程也包括在本发明的范围内。以与所描述的不同的顺序来改变按预定顺序描述的多个实施例也包括在本发明的范围内。

[0575] 已经基于一系列的步骤或流程图来描述实施例,但这并不限定发明的时间序列顺序,并且根据需要可同时执行或者按另外顺序执行。另外,在上述实施例中,构成框图的结构要素(例如,单元、模块等)还可以分别实现为硬件设备或软件,并且多个结构要素可以组合实施为单个硬件设备或软件。所述实施例可以通过程序指令的形式实施,所述程序指令可以由各种计算机部件执行并记录在计算机可读记录介质中。所述计算机可读记录介质可以单独地或组合地包括程序指令、数据文件、数据结构等。计算机可读记录介质的示例可以包括硬盘、软盘和磁带等磁性介质、CD-ROM、DVD等光学记录介质、如软式光盘(floptical disk)的磁-光介质(magneto-optical media)和ROM、RAM、闪速存储器等等以存储程序指令并执行该指令的方式专门配置的硬件设备。所述硬件设备可以被配置为作为一个或多个软件模块进行操作以执行根据本发明的处理,反之亦然。

[0576] 工业实用性

[0577] 本发明可以应用于对视频进行编码/解码的电子设备。

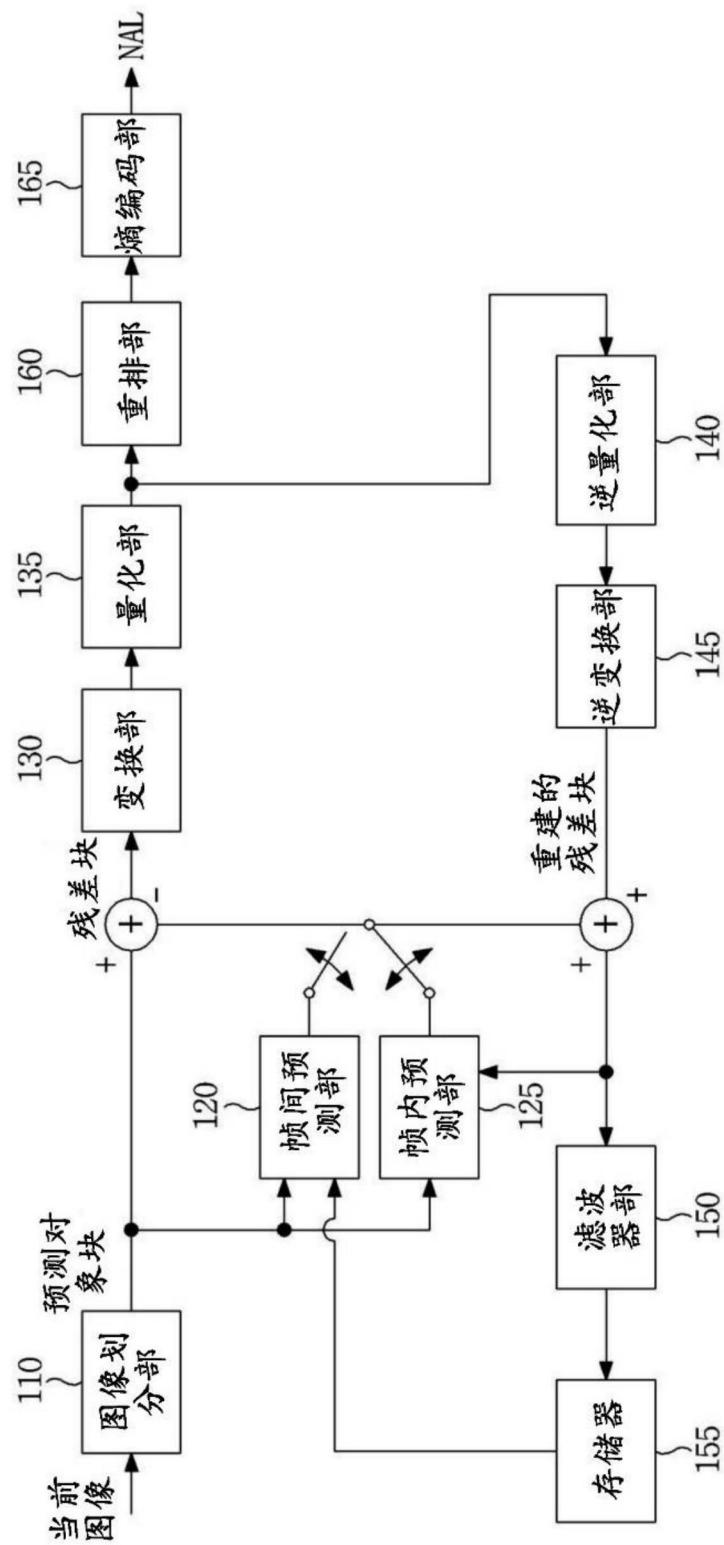


图1

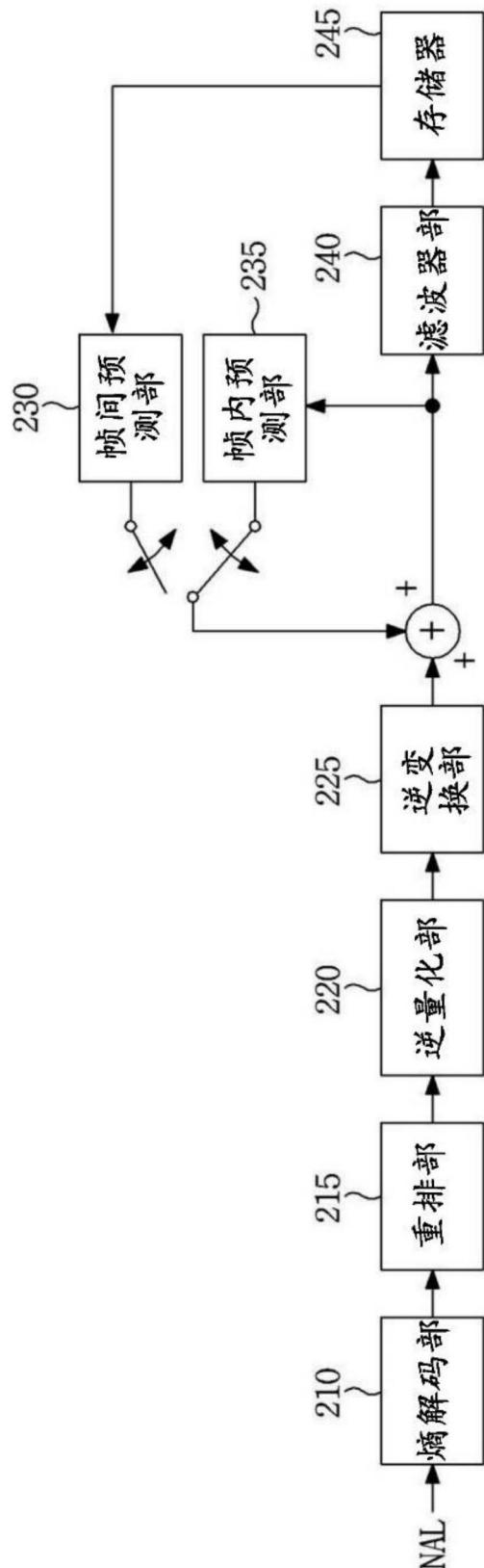


图2

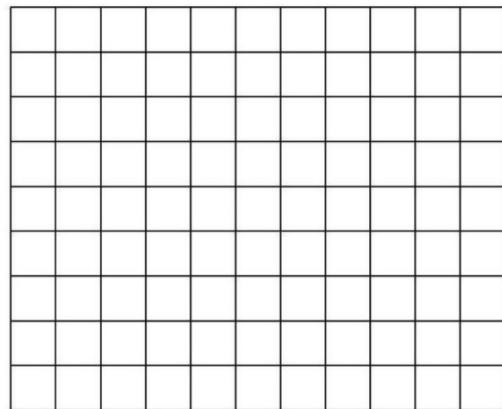


图3

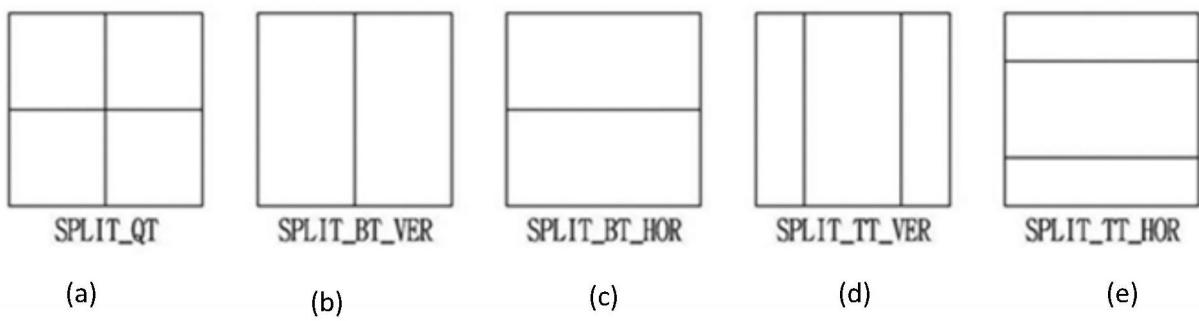


图4

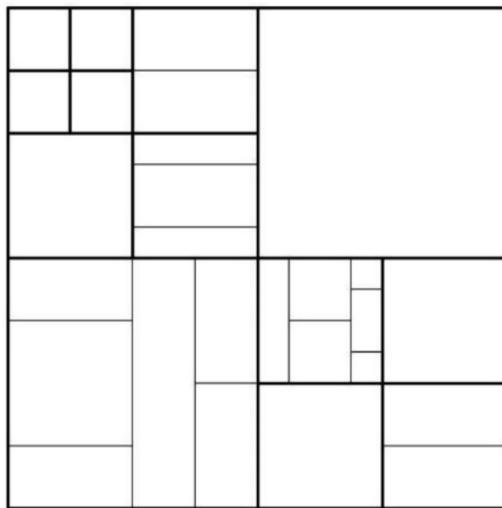


图5

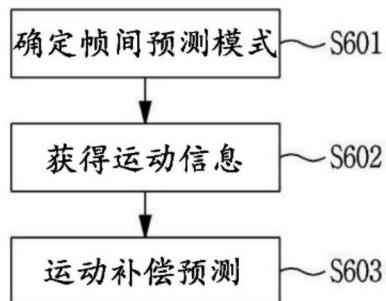


图6

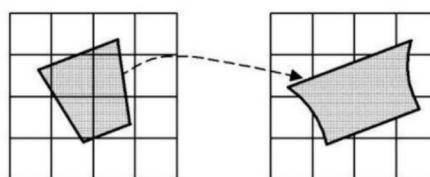
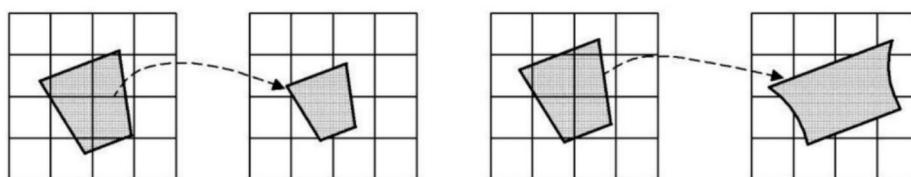
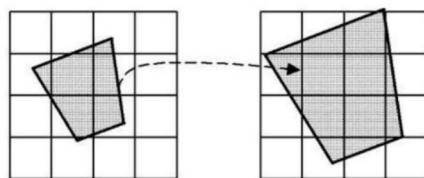
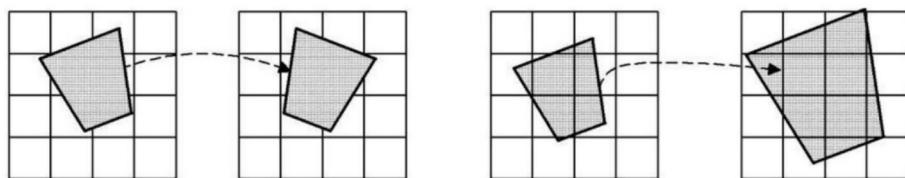


图7

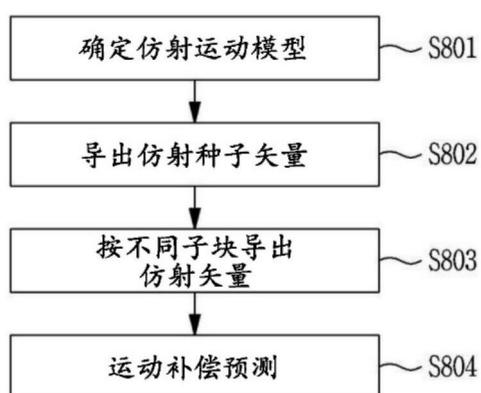


图8

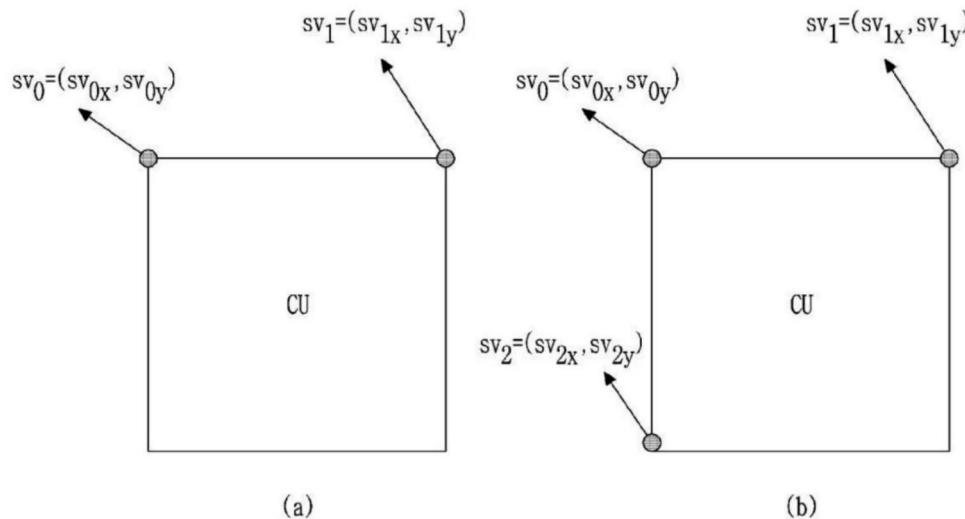


图9

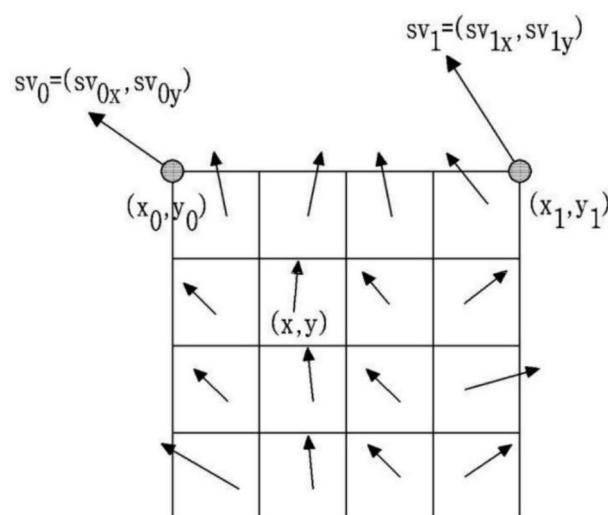


图10

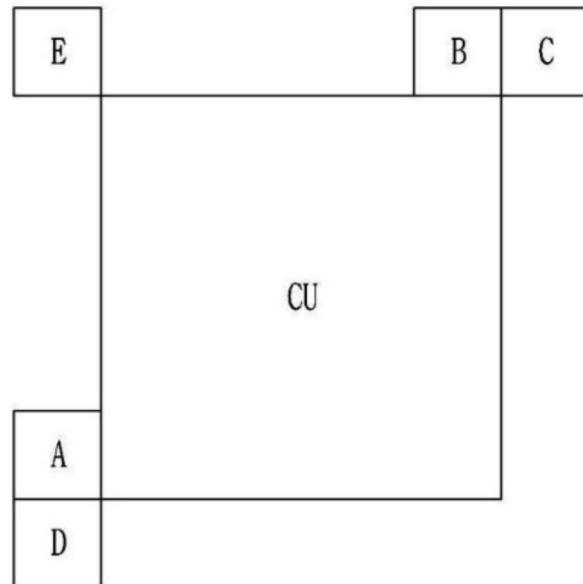


图11

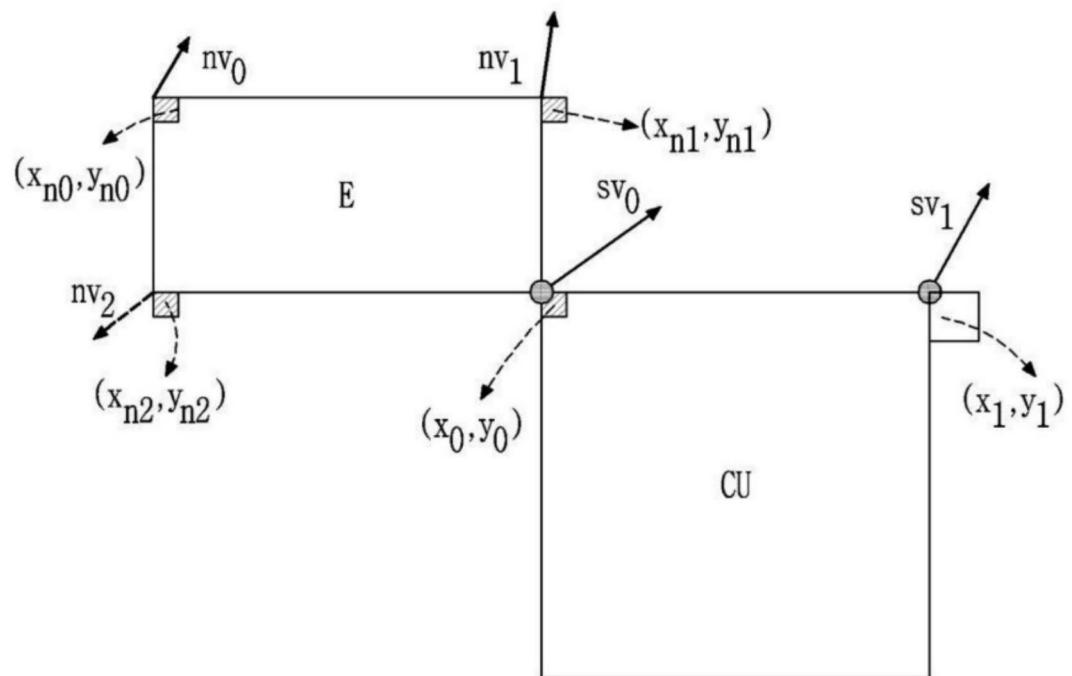


图12

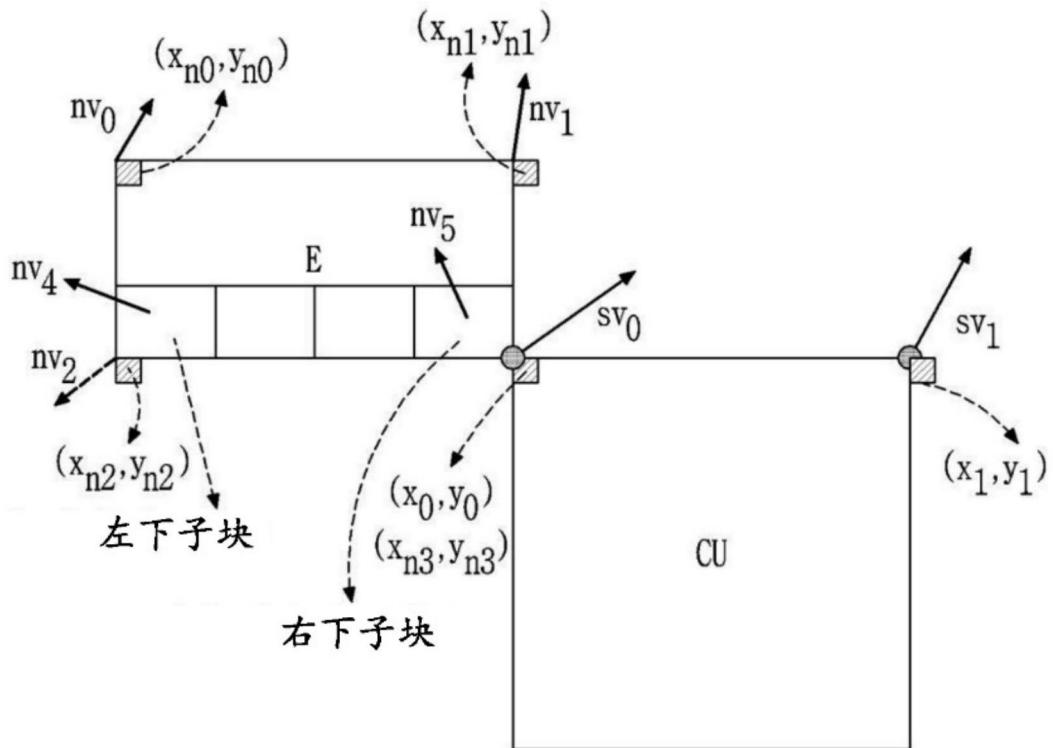


图13

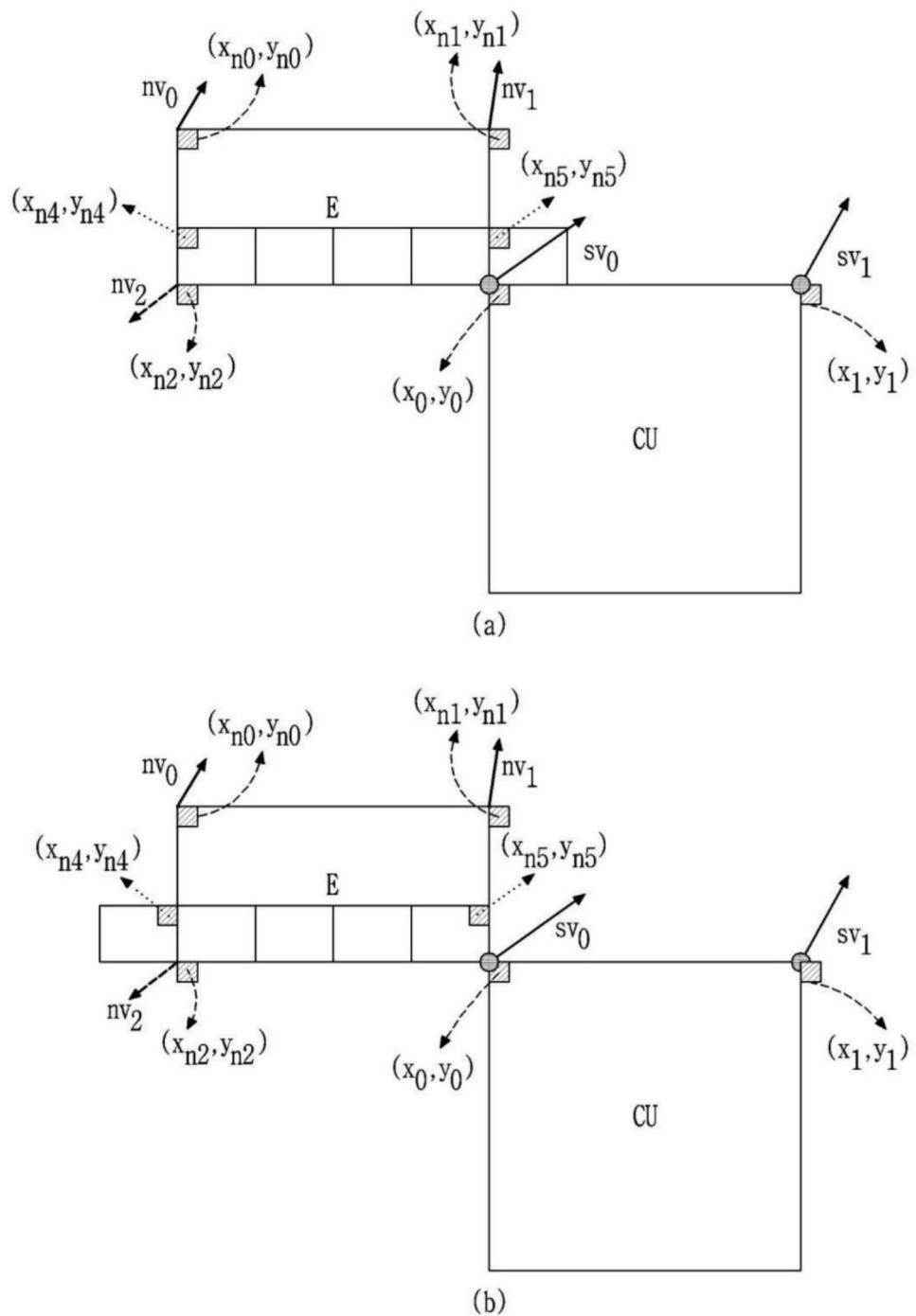


图14

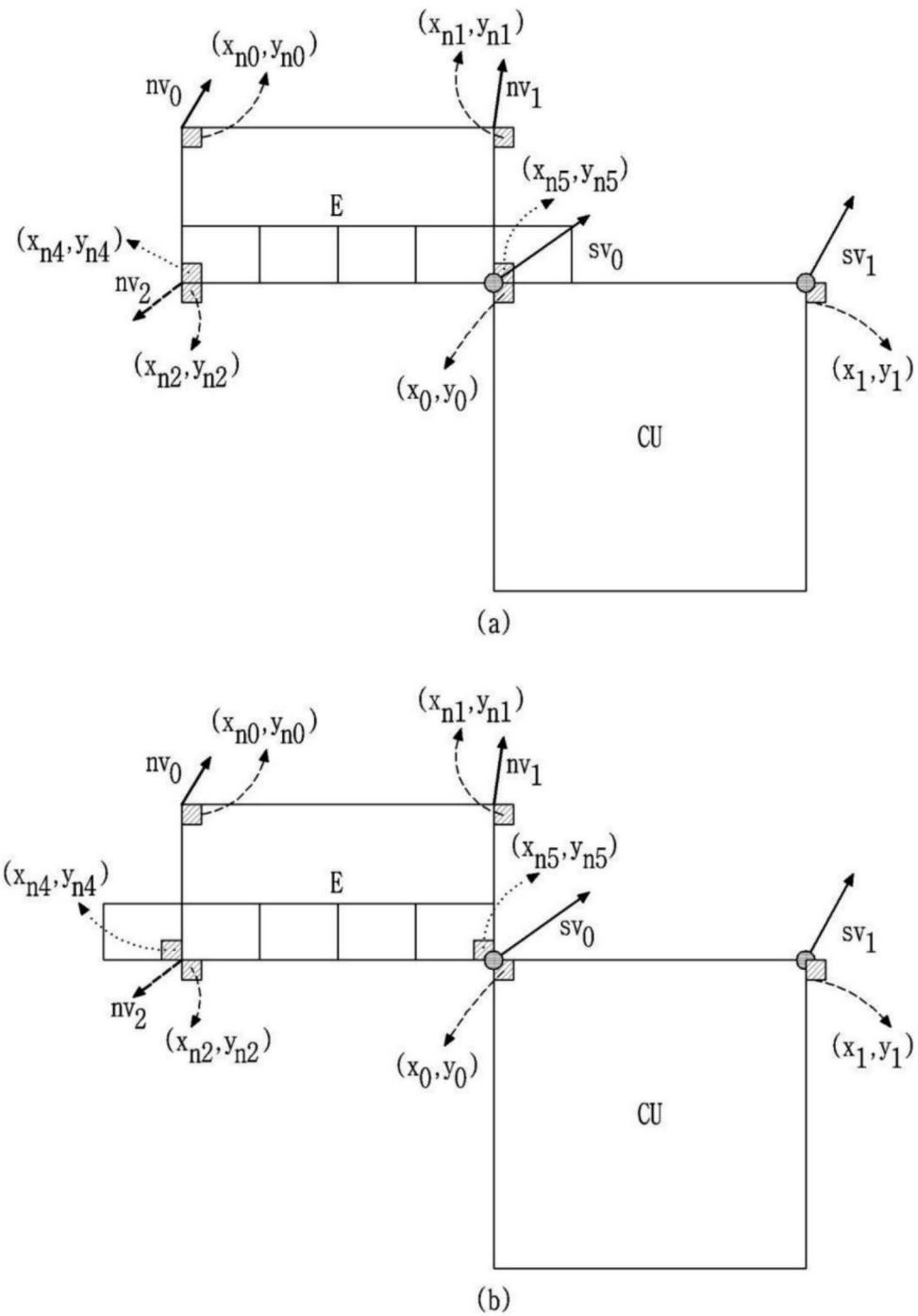


图15

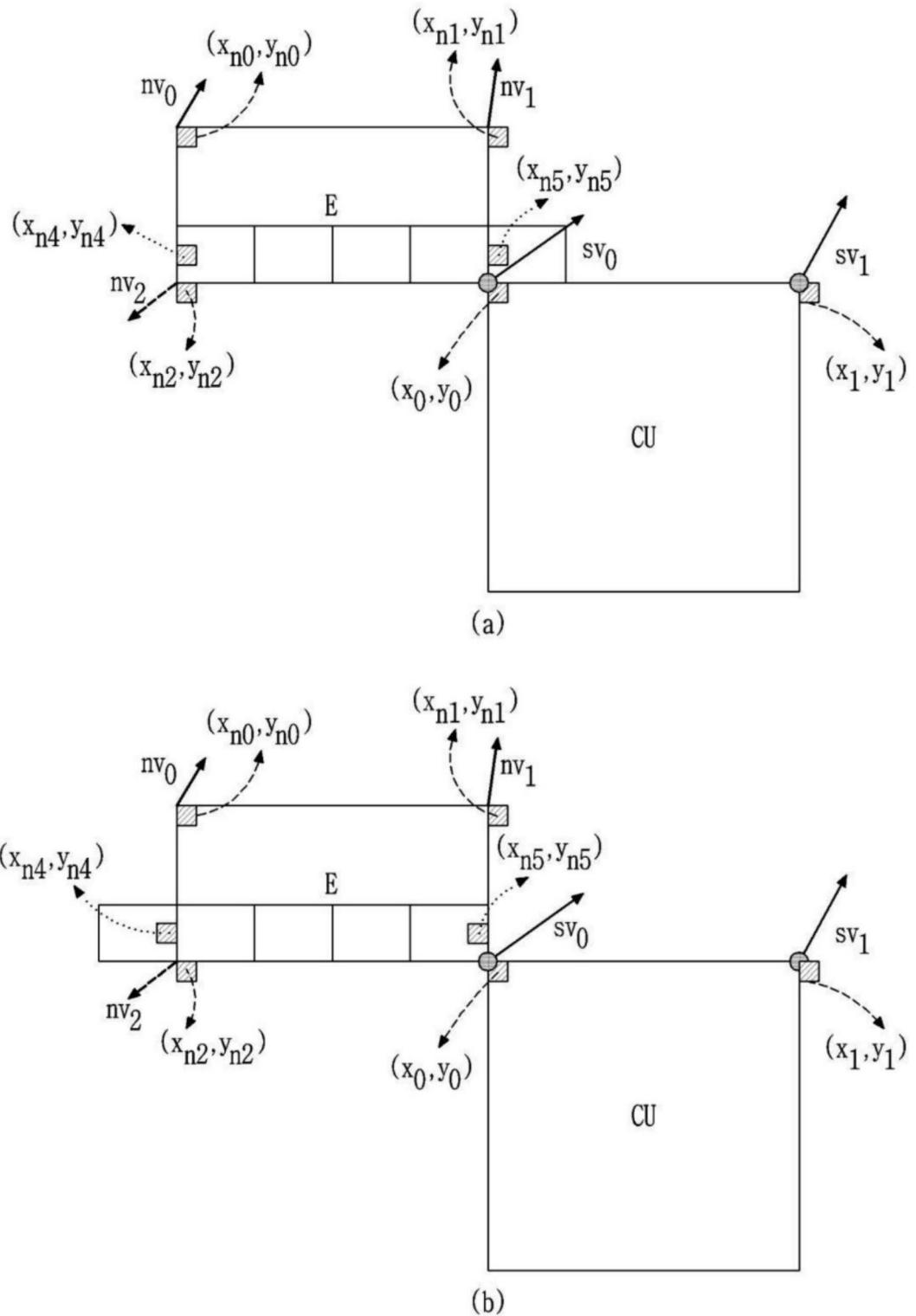


图16

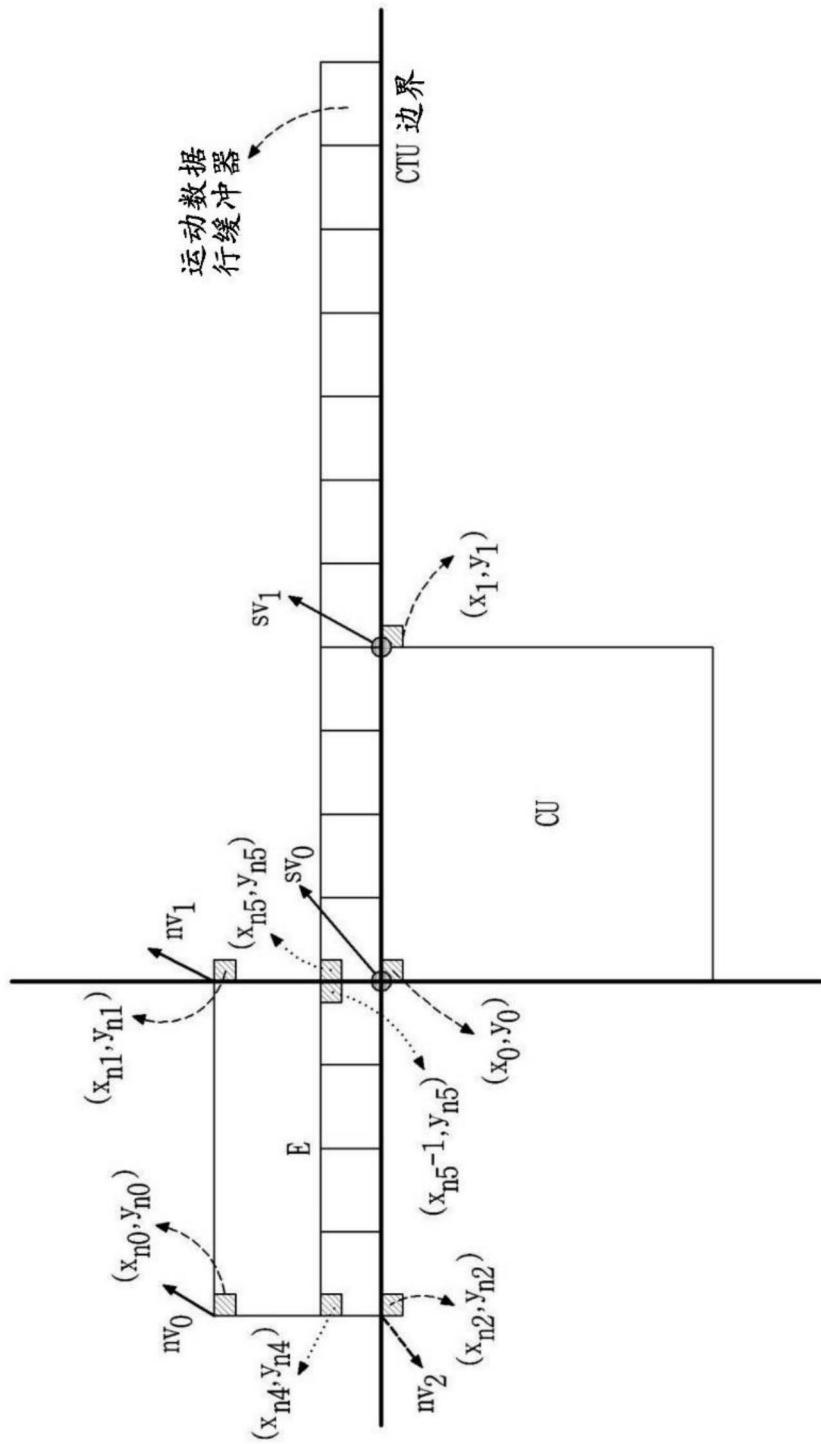


图17

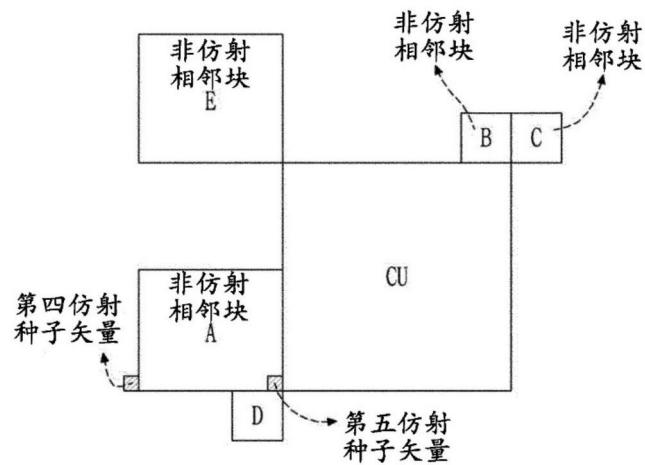


图18

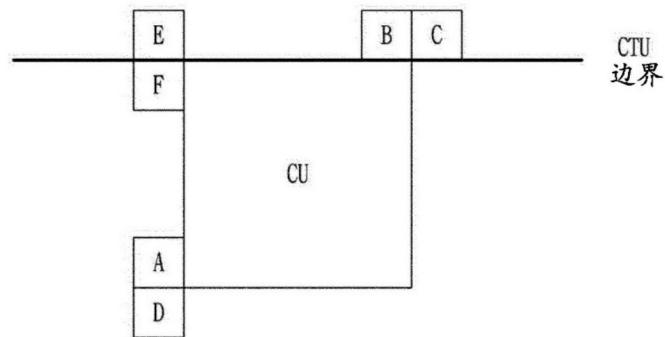


图19

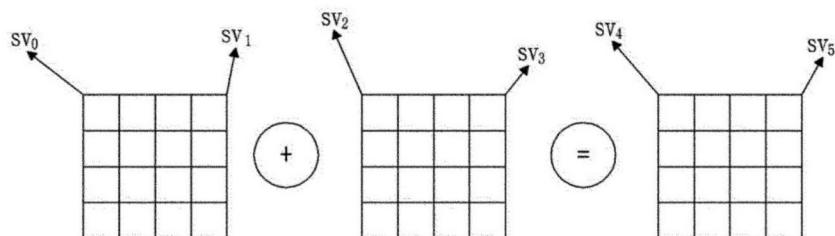


图20

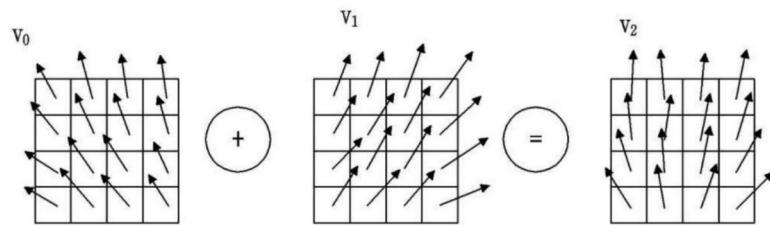


图21

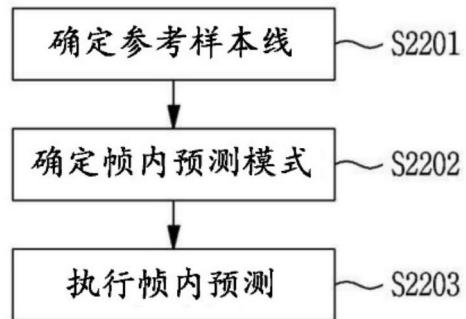


图22

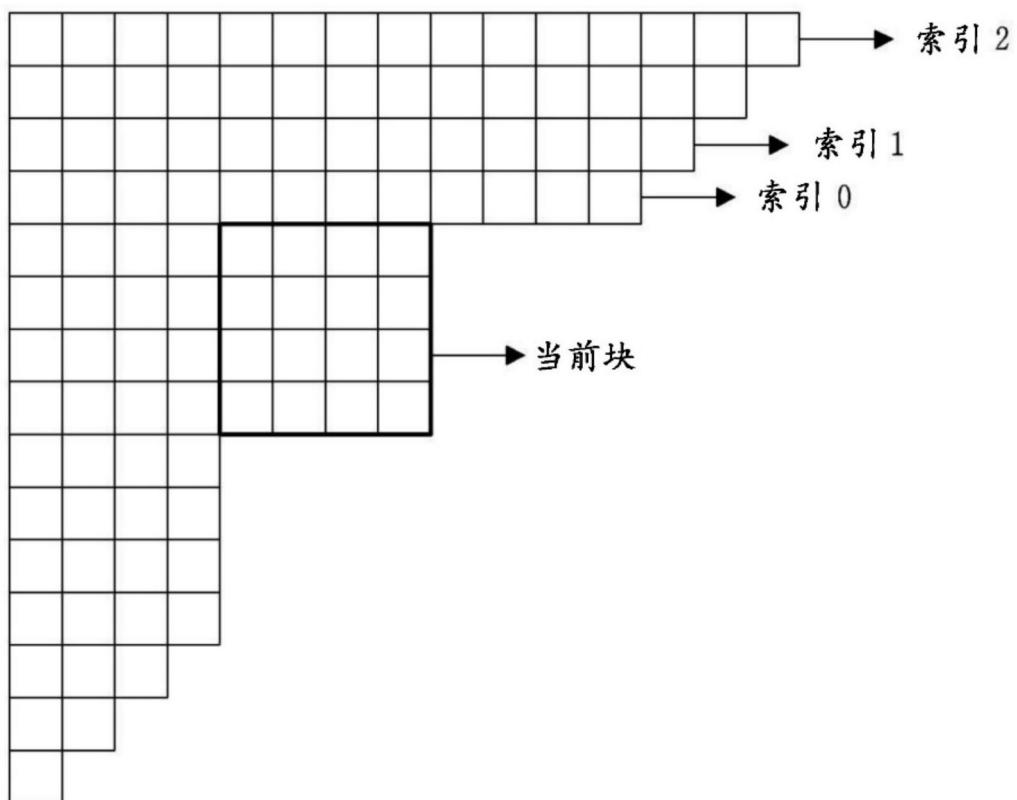


图23

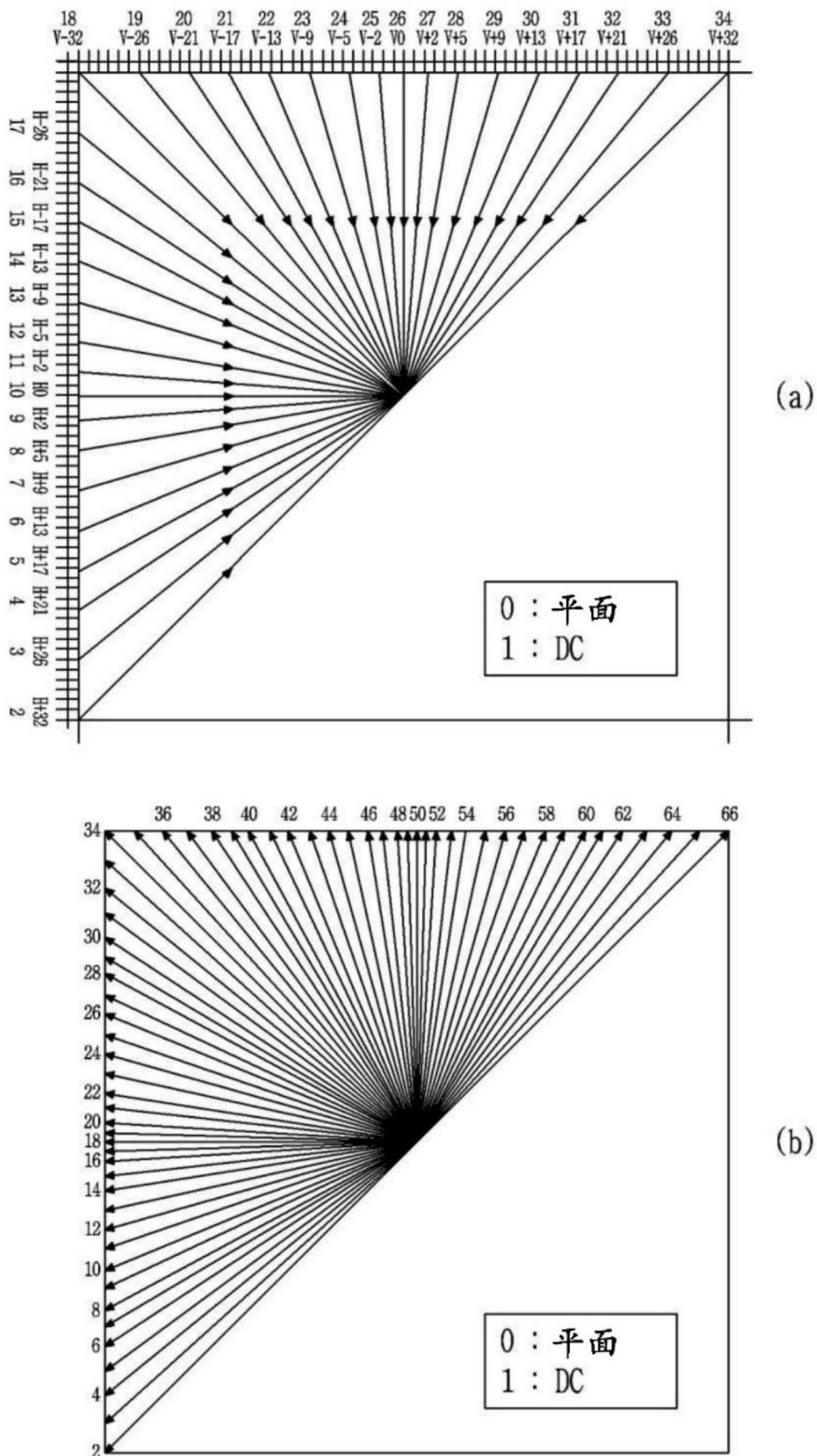


图24

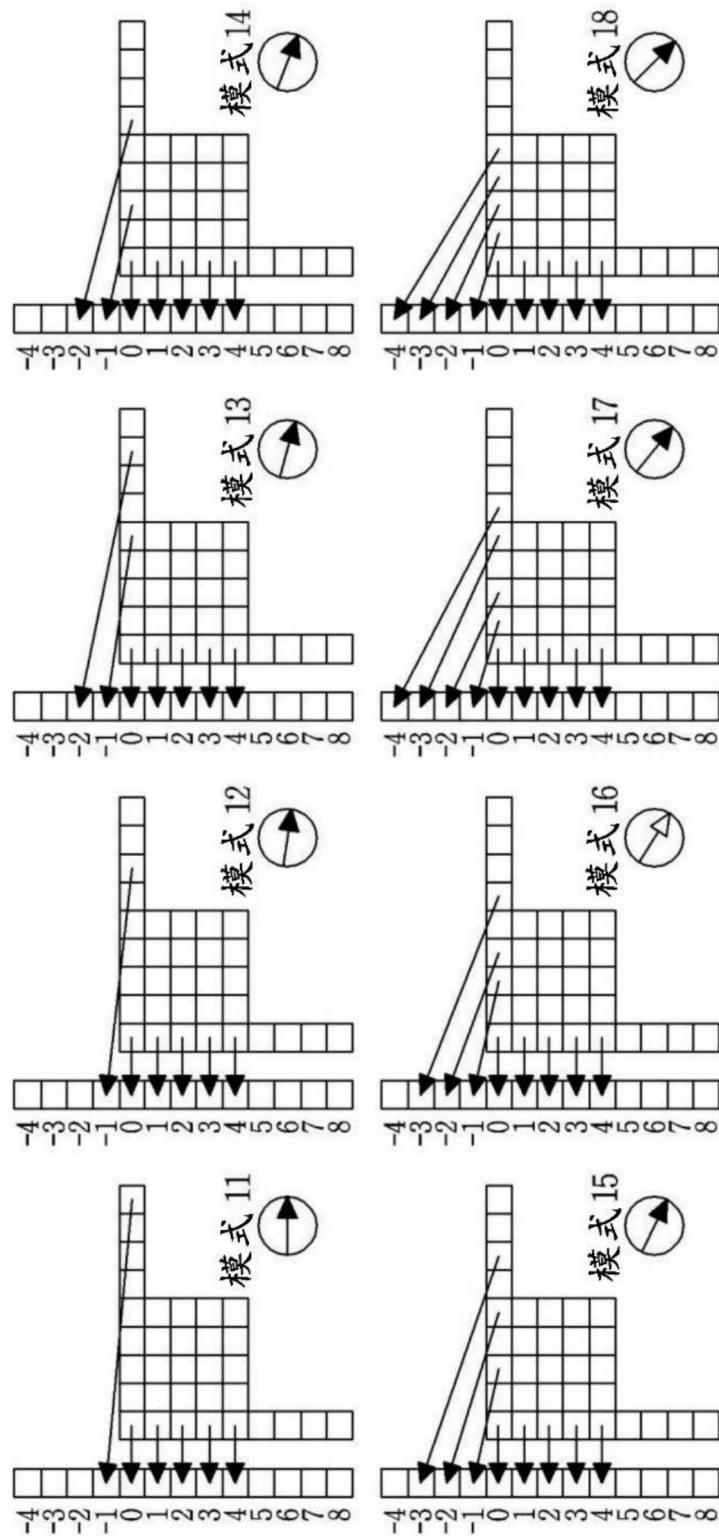


图25

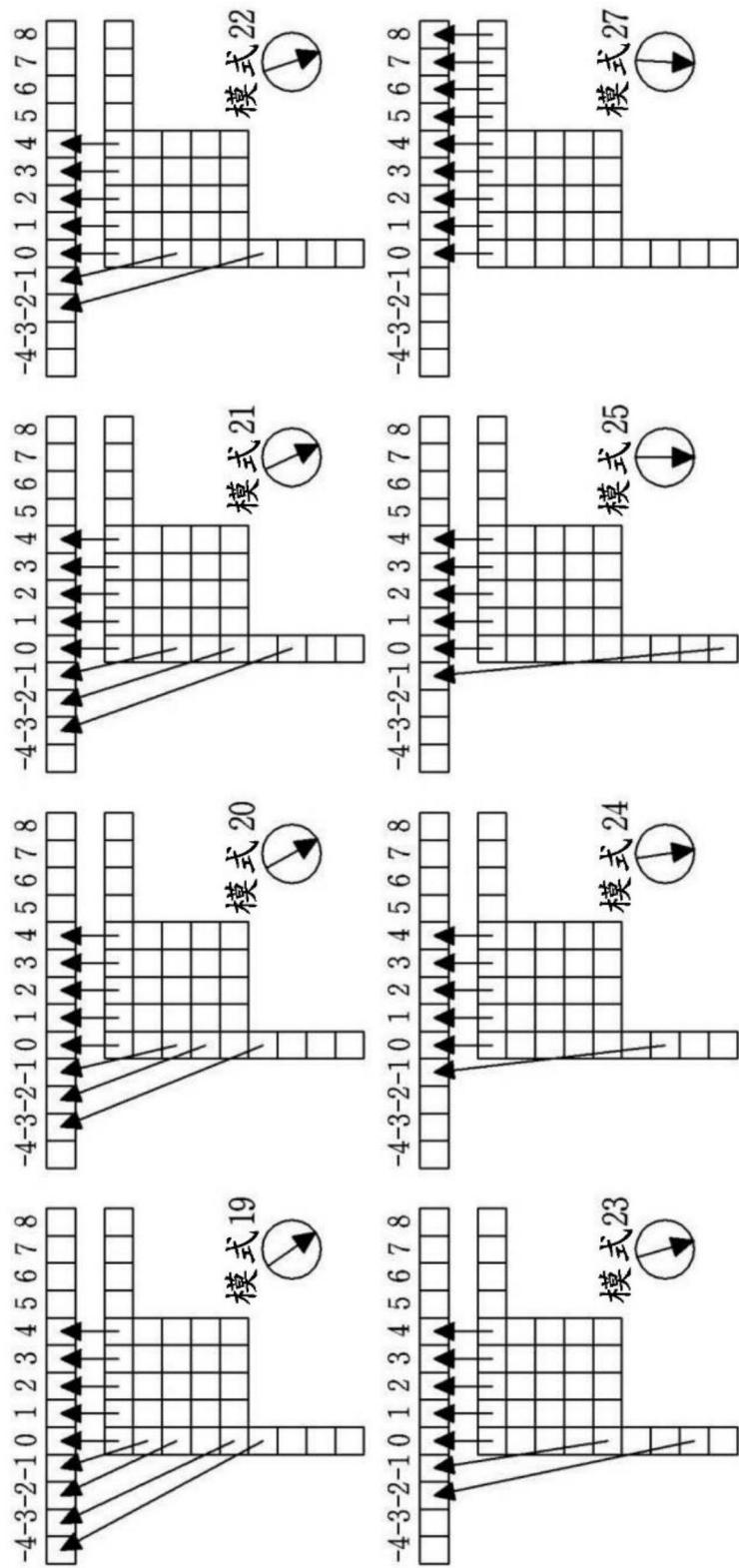


图 26

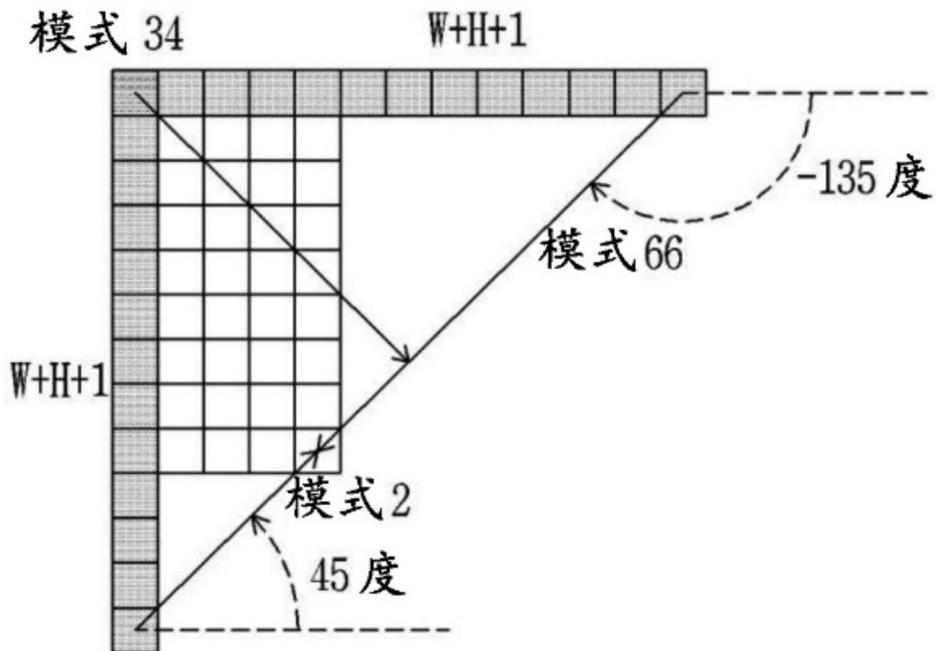


图27

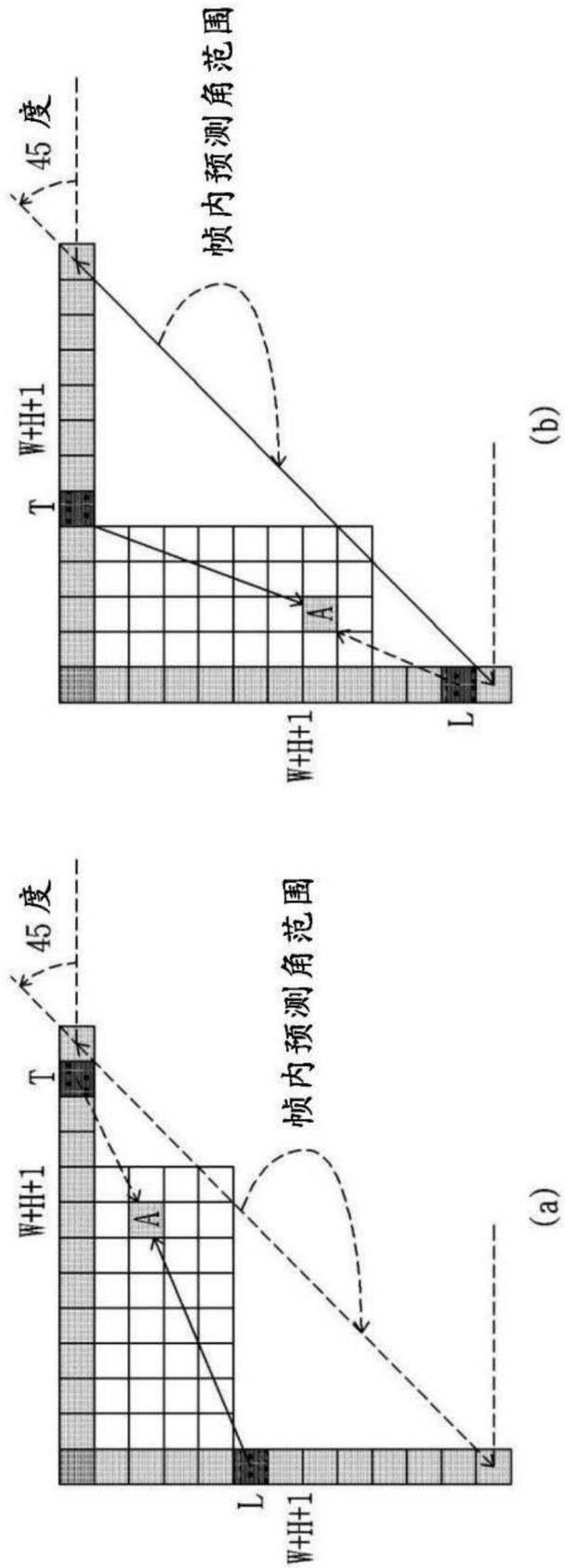


图28

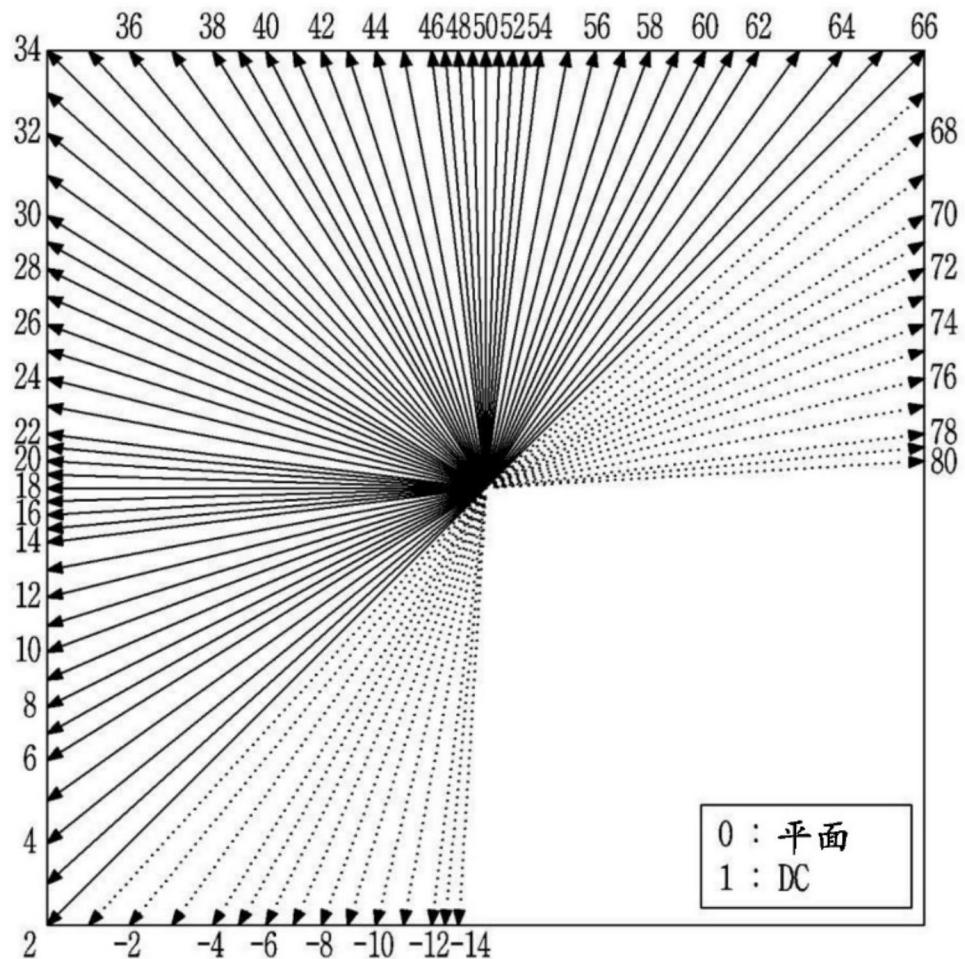


图29

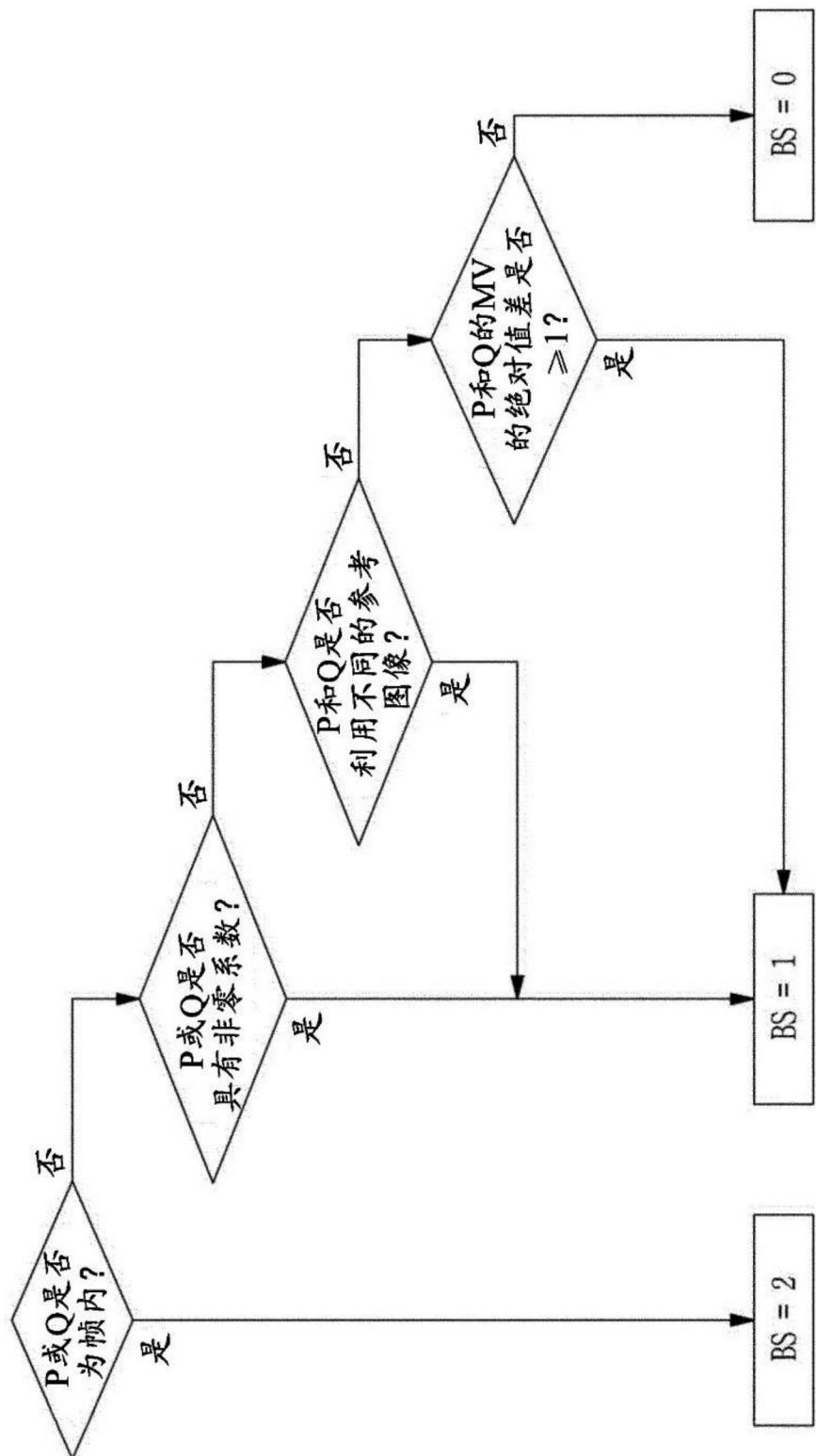


图30

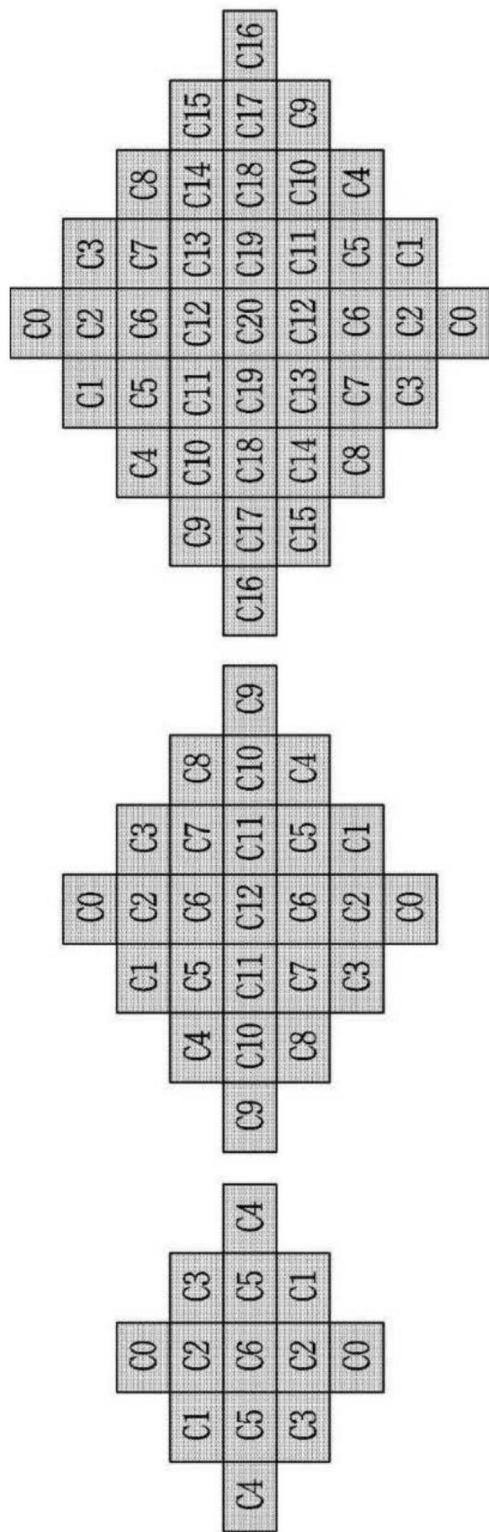


图31

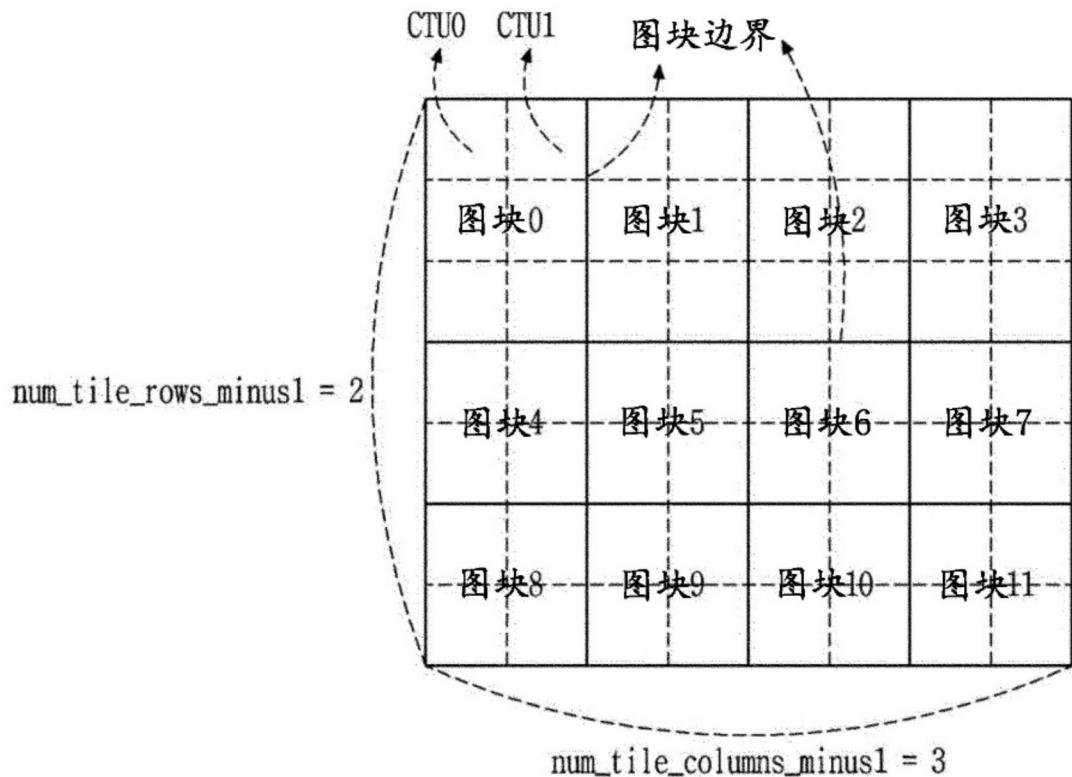


图32

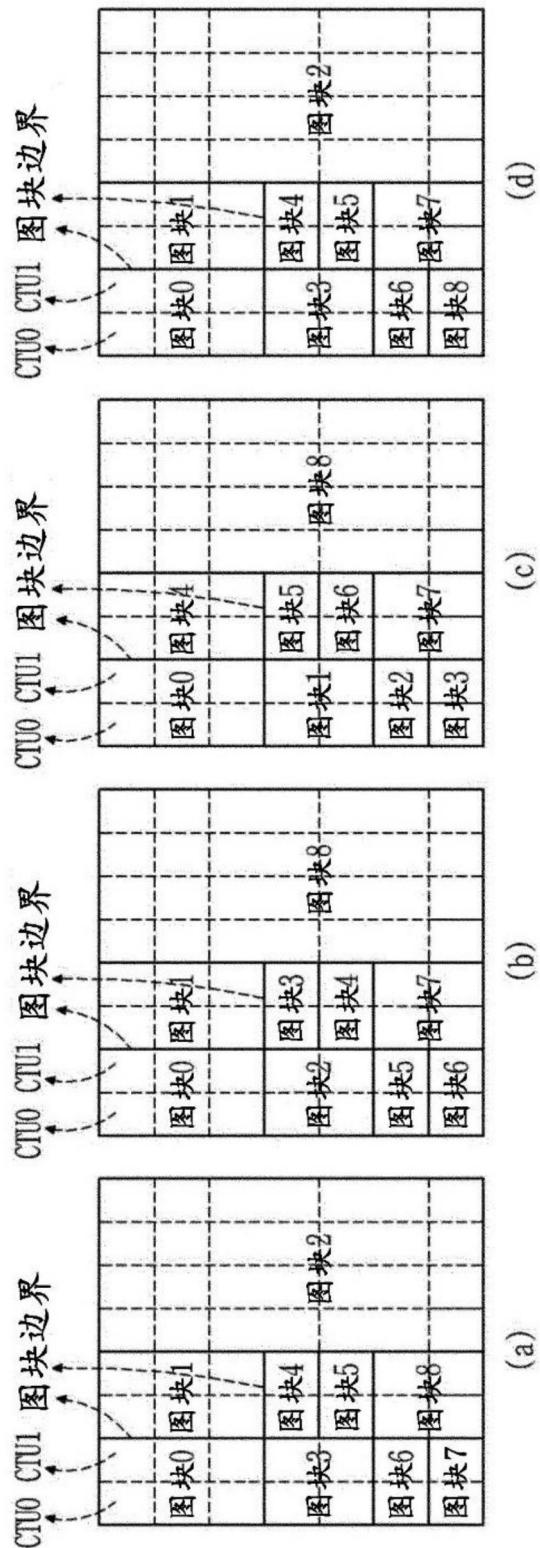


图33

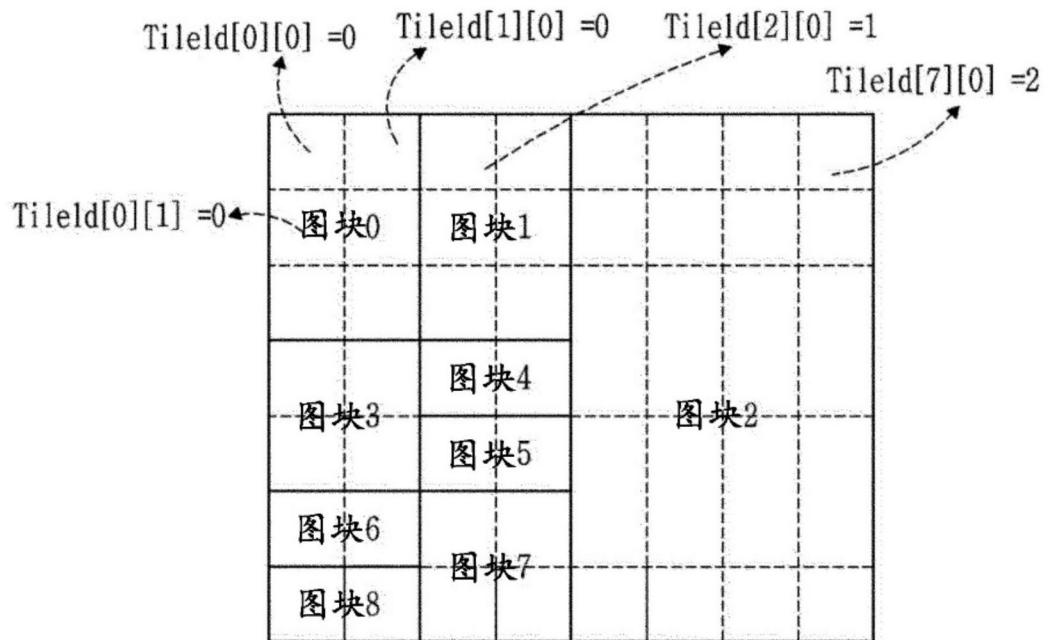


图34



图35