

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7449279号
(P7449279)

(45)発行日 令和6年3月13日(2024.3.13)

(24)登録日 令和6年3月5日(2024.3.5)

(51)国際特許分類
H 0 4 N 19/52 (2014.01)F I
H 0 4 N 19/52

請求項の数 15 (全90頁)

(21)出願番号 特願2021-515161(P2021-515161)
 (86)(22)出願日 令和1年9月20日(2019.9.20)
 (65)公表番号 特表2022-502899(P2022-502899)
 A)
 (43)公表日 令和4年1月11日(2022.1.11)
 (86)国際出願番号 PCT/KR2019/012290
 (87)国際公開番号 WO2020/060327
 (87)国際公開日 令和2年3月26日(2020.3.26)
 審査請求日 令和4年8月24日(2022.8.24)
 (31)優先権主張番号 10-2018-0114342
 (32)優先日 平成30年9月21日(2018.9.21)
 (33)優先権主張国・地域又は機関
韓国(KR)
 (31)優先権主張番号 10-2018-0114343
 (32)優先日 平成30年9月21日(2018.9.21)

最終頁に続く

(73)特許権者 516227559
 オッポ広東移動通信有限公司
 GUANGDONG OPPOMOBILE
 TELECOMMUNICATIONS CORP., LTD.
 中華人民共和国広東省東莞市長安鎮烏沙
 海浜路18号
 No. 18 Haibin Road,
 Wusha, Chang'an, Dong
 guan, Guangdong 52
 3860 China
 (74)代理人 100091487
 弁理士 中村 行孝
 (74)代理人 100120031
 弁理士 宮嶋 学

最終頁に続く

(54)【発明の名称】 ビデオ信号符号化 / 復号化方法及びそのための機器

(57)【特許請求の範囲】

【請求項1】

ビデオ復号化方法であって、
 現在ブロックのマージ候補リストを生成するステップと、
 前記マージ候補リストに含まれる複数のマージ候補のうちのいずれか1つを指定するステップと、
 指定されたマージ候補の第1アフィンシードベクトル及び第2アフィンシードベクトルに基づいて、前記現在ブロックの第1アフィンシードベクトル及び第2アフィンシードベクトルを導出するステップと、

前記現在ブロックの前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを用いることで、前記現在ブロック内のサブブロックのアフィンベクトルを導出するステップであって、前記サブブロックは、前記現在ブロックのサイズよりも小さいサイズの領域である、ステップと、

前記アフィンベクトルに基づいて、前記サブブロックに対して動き補償予測を実行するステップと、を含み、

前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルは、前記現在ブロックに隣接する隣接ブロックの動き情報に基づいて導出される、ビデオ復号化方法。

【請求項2】

前記隣接ブロックが、前記現在ブロックの符号化ツリーユニットと異なる符号化ツリー

ユニットに含まれる場合、前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルは、前記隣接ブロックの左下サブブロック及び右下サブブロックの動きベクトルに基づいて導出されることを特徴とする

請求項1に記載のビデオ復号化方法。

【請求項3】

前記左下サブブロックは、前記隣接ブロックの左下隅に位置する左下参照サンプルを含み、前記右下サブブロックは、前記隣接ブロックの右下隅に位置する右下参照サンプルを含むことを特徴とする

請求項2に記載のビデオ復号化方法。

【請求項4】

前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルは、スケールファクタを用いて、前記左下サブブロックと前記右下サブブロックとの動きベクトルの差分値に対してシフト操作を行うことで得られた値に基づいて導出され、

前記スケールファクタは、左下参照サンプルと右下参照サンプルとの水平距離とオフセットを加算することで得られた値に基づいて導出されることを特徴とする

請求項3に記載のビデオ復号化方法。

【請求項5】

前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルは、スケールファクタを用いて、前記左下サブブロックと前記右下サブブロックとの動きベクトルの差分値に対してシフト操作を行うことで得られた値に基づいて導出され、前記スケールファクタは、前記右下参照サンプルの右側に隣接する隣接サンプルと前記左下参照サンプルとの距離に基づいて導出することを特徴とする

請求項3に記載のビデオ復号化方法。

【請求項6】

前記マージ候補リストは、第1マージ候補及び第2マージ候補を含み、前記第1マージ候補は、前記現在ブロックの上方に位置する上隣接ブロックのうちの第1利用可能なブロックと決定された上隣接ブロックに基づいて導出され、前記第2マージ候補は、前記現在ブロックの左側に位置する左隣接ブロックのうちの第1利用可能なブロックと決定された左隣接ブロックに基づいて導出され、

及び/又は

前記隣接ブロックが、前記現在ブロックの符号化ツリーユニットと同じである符号化ツリーユニットに含まれる場合、前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルは、前記隣接ブロックの第1アフィンシードベクトル及び第2アフィンシードベクトルに基づいて導出されることを特徴とする

請求項1に記載のビデオ復号化方法。

【請求項7】

隣接アフィンベクトルの左下制御点に関わる第3アフィンシードベクトル及び右下制御点に関わる第4アフィンシードベクトルを用いて、前記現在ブロックの第1アフィンシードベクトル及び第2アフィンシードベクトルを導出し、

10

20

30

40

50

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

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

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

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

(x_{n2}, y_{n2}) は、アフィン隣接ブロックの左下制御点の座標を表し、また、(x_n, y_n) は、アフィン隣接ブロックの右下制御点の座標を表し、

(x_0, y_0) は、現在ブロックの左上制御点の座標を表し、また、(x_1, y_1) は、現在ブロックの右上制御点の座標を表し、

(nv_{2x}, nv_{2y}) は、アフィン隣接ブロックの左下制御点の第 3 アフィンシードベクトルを表し、また、(nv_{3x}, nv_{3y}) は、アフィン隣接ブロックの右下制御点の第 4 アフィンシードベクトルを表し、

(v_{0x}, v_{0y}) は、現在ブロックの左上制御点の第 1 アフィンシードベクトルを表し、また、(v_{1x}, v_{1y}) は、現在ブロックの右上制御点の第 2 アフィンシードベクトルを表すことを特徴とする

請求項 1 に記載のビデオ復号化方法。

【請求項 8】

ビデオ符号化方法であって、

現在ブロックのマージ候補リストを生成するステップと、

前記マージ候補リストに含まれる複数のマージ候補のうちのいずれか 1 つを指定するステップと、

指定されたマージ候補の第 1 アフィンシードベクトル及び第 2 アフィンシードベクトルに基づいて、前記現在ブロックの第 1 アフィンシードベクトル及び第 2 アフィンシードベクトルを導出するステップと、

前記現在ブロックの前記第 1 アフィンシードベクトル及び前記第 2 アフィンシードベクトルを用いることで、前記現在ブロック内のサブブロックのアフィンベクトルを導出するステップであって、前記サブブロックは、前記現在ブロックのサイズよりも小さいサイズの領域である、ステップと、

前記アフィンベクトルに基づいて、前記サブブロックに対して動き補償予測を実行するステップと、を含み、

前記マージ候補の前記第 1 アフィンシードベクトル及び前記第 2 アフィンシードベクトルは、前記現在ブロックに隣接する隣接ブロックの動き情報に基づいて導出される、ビデオ符号化方法。

【請求項 9】

前記隣接ブロックが、前記現在ブロックの符号化ツリーユニットと異なる符号化ツリーユニットに含まれる場合、前記マージ候補の前記第 1 アフィンシードベクトル及び前記第 2 アフィンシードベクトルは、前記隣接ブロックの左下サブブロック及び右下サブブロックの動きベクトルに基づいて導出されることを特徴とする

10

20

30

40

50

請求項 8 に記載のビデオ符号化方法。

【請求項 10】

前記左下サブブロックは、前記隣接ブロックの左下隅に位置する左下参照サンプルを含み、前記右下サブブロックは、前記隣接ブロックの右下隅に位置する右下参照サンプルを含むことを特徴とする

請求項 9 に記載のビデオ符号化方法。

【請求項 11】

前記マージ候補の前記第 1 アフィンシードベクトル及び前記第 2 アフィンシードベクトルは、スケールファクタを用いて、前記左下サブブロックと前記右下サブブロックとの動きベクトルの差分値に対してシフト操作を行うことで得られた値に基づいて導出され、

前記スケールファクタは、左下参照サンプルと右下参照サンプルとの水平距離とオフセットを加算することで得られた値に基づいて導出され、又は、前記スケールファクタは、前記右下参照サンプルの右側に隣接する隣接サンプルと前記左下参照サンプルとの距離に基づいて導出されることを特徴とする

請求項 10 に記載のビデオ符号化方法。

【請求項 12】

前記マージ候補リストは、第 1 マージ候補及び第 2 マージ候補を含み、前記第 1 マージ候補は、前記現在ブロックの上方に位置する上隣接ブロックのうちの第 1 利用可能なブロックと決定された上隣接ブロックに基づいて導出され、前記第 2 マージ候補は、前記現在ブロックの左側に位置する左隣接ブロックのうちの第 1 利用可能なブロックと決定された左隣接ブロックに基づいて導出され、

及び / 又は

前記隣接ブロックが、前記現在ブロックの符号化ツリーユニットと同じである符号化ツリーユニットに含まれる場合、前記マージ候補の前記第 1 アフィンシードベクトル及び前記第 2 アフィンシードベクトルは、前記隣接ブロックの第 1 アフィンシードベクトル及び第 2 アフィンシードベクトルに基づいて導出されることを特徴とする

請求項 8 に記載のビデオ符号化方法。

【請求項 13】

隣接アフィンベクトルの左下制御点に関わる第 3 アフィンシードベクトル及び右下制御点に関わる第 4 アフィンシードベクトルを用いて、前記現在ブロックの第 1 アフィンシードベクトル及び第 2 アフィンシードベクトルを導出し、

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

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

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

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

(x_{n2}, y_{n2}) は、アフィン隣接ブロックの左下制御点の座標を表し、また、(x_n

10

20

30

40

50

(x_3, y_3) は、アフィン隣接ブロックの右下制御点の座標を表し、

(x_0, y_0) は、現在ブロックの左上制御点の座標を表し、また、 (x_1, y_1) は、現在ブロックの右上制御点の座標を表し、

(nv_{2x}, nv_{2y}) は、アフィン隣接ブロックの左下制御点の第3アフィンシードベクトルを表し、また、 (nv_{3x}, nv_{3y}) は、アフィン隣接ブロックの右下制御点の第4アフィンシードベクトルを表し、

(v_{0x}, v_{0y}) は、現在ブロックの左上制御点の第1アフィンシードベクトルを表し、また、 (v_{1x}, v_{1y}) は、現在ブロックの右上制御点の第2アフィンシードベクトルを表すことを特徴とする

請求項8に記載のビデオ符号化方法。

10

【請求項14】

請求項1～7のいずれか一項に記載のビデオ復号化方法を実行するように構成されるビデオデコーダ。

【請求項15】

請求項8～13のいずれか一項に記載のビデオ符号化方法を実行するように構成されるビデオエンコーダ。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ビデオ信号符号化/復号化方法及びそのための機器に関する。

20

【背景技術】

【0002】

表示パネルがますます大きくなっていることにつれて、画質がより高いビデオサービスは必要になっている。高精細度ビデオサービスの最大の問題は、データ量の大幅な増加である。このような問題を解決するために、ビデオ圧縮率の向上に関する検討を積極的に進行している。代表的な例として、2009年に、動画像専門家グループ(Motion Picture Experts Group: MPEG)及び国際電気通信連合-電気通信(International Telecommunication Union - Telecommunication: ITU-T)傘下のビデオ符号化専門家グループ化(Video Coding Experts Group: VCEG)は、ビデオ符号化共同チームJCT-VC(Joint Collaborative Team on Video Coding)を確立した。JCT-VCは、ビデオ圧縮規格HEVC(高効率ビデオ符号化: High Efficiency Video Coding)を提出し、該規格が2013年1月25日に承認された。その圧縮性能は、H.264/AVCの約2倍である。高精細度ビデオサービスの急激な成長に伴い、HEVCの性能上の制限も次第に現れている。

30

【発明の概要】

【発明が解決しようとする課題】

【0003】

本発明の目的は、ビデオ信号に対して符号化/復号化を行う場合にアフィンモデルを用いるインター予測方法及び前記インター予測方法に用いられる機器を提供することである。

40

【0004】

本発明の目的は、ビデオ信号に対して符号化/復号化を行う場合にサブブロックの並進動きベクトルを用いてアフィンシードベクトルを導出する方法及び前記方法を実行するための機器を提供することである。

【0005】

本発明のもう1つの目的は、ビデオ信号に対して符号化/復号化を行う場合に隣接ブロックと現在ブロックとの距離を2のべき級数に変換することでアフィンシードベクトルを導出する方法及び前記方法を実行するための機器を提供することである。

【0006】

50

本発明が実現しようとする技術的課題は、上記言及した技術的課題に限定されず、本発明が属する技術分野において通常の知識を有する者は、下記説明により、言及されていない他の技術的課題を明瞭に理解する。

【課題を解決するための手段】

【0007】

本発明によるビデオ信号復号化／符号化方法は、現在ブロックのマージ候補リストを生成するステップと、前記マージ候補リストに含まれる複数のマージ候補のうちのいずれか1つを指定するステップと、指定されたマージ候補の第1アフィンシードベクトル及び第2アフィンシードベクトルに基づいて、前記現在ブロックの第1アフィンシードベクトル及び第2アフィンシードベクトルを導出するステップと、前記現在ブロックの前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを用いることで、前記現在ブロック内のサブブロックのアフィンベクトルを導出するステップと、前記アフィンベクトルに基づいて、前記サブブロックに対して動き補償予測を実行するステップと、を含む。この場合、前記サブブロックは、前記現在ブロックのサイズよりも小さいサイズの領域である。なお、前記現在ブロックに隣接する隣接ブロックの動き情報に基づいて、前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを導出することができる。

【0008】

本発明によるビデオ信号復号化／符号化方法において、前記隣接ブロックが、前記現在ブロックの符号化ツリーユニットと異なる符号化ツリーユニットに含まれる場合、前記隣接ブロックの左下サブブロック及び右下サブブロックの動きベクトルに基づいて、前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを導出することができる。

【0009】

本発明によるビデオ信号復号化／符号化方法において、前記左下サブブロックは、前記隣接ブロックの左下隅に位置する左下参照サンプルを含んでもよく、前記右下サブブロックは、前記隣接ブロックの右下隅に位置する右下参照サンプルを含んでもよい。

【0010】

本発明によるビデオ信号復号化／符号化方法において、スケールファクタを用いて、前記左下サブブロックと前記右下サブブロックとの動きベクトルの差分値に対してシフト操作を行うことで得られた値に基づいて、前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを導出することができ、また、左下参照サンプルと右下参照サンプルとの水平距離とオフセットを加算することで得られた値に基づいて、前記スケールファクタを導出することができる。

【0011】

本発明によるビデオ信号復号化／符号化方法において、スケールファクタを用いて、前記左下サブブロックと前記右下サブブロックとの動きベクトルの差分値に対してシフト操作を行うことで得られた値に基づいて、前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを導出することができ、前記右下参照サンプルの右側に隣接する隣接サンプルと前記左下参照サンプルとの距離に基づいて、前記スケールファクタを導出することができる。

【0012】

本発明によるビデオ信号復号化／符号化方法において、前記マージ候補リストは、第1マージ候補及び第2マージ候補を含み、前記第1マージ候補は、前記現在ブロックの上方に位置する上隣接ブロックのうちの第1利用可能なブロックと決定された上隣接ブロックに基づいて導出され、前記第2マージ候補は、前記現在ブロックの左側に位置する左隣接ブロックのうちの第1利用可能なブロックと決定された左隣接ブロックに基づいて導出される。

【0013】

本発明によるビデオ信号復号化／符号化方法において、前記隣接ブロックが、前記現在

10

20

30

40

50

ロックの符号化ツリーユニットと同じである符号化ツリーユニットに含まれる場合、前記隣接ロックの第1アフィンシードベクトル及び第2アフィンシードベクトルに基づいて前記マージ候補の前記第1アフィンシードベクトル及び前記第2アフィンシードベクトルを導出することができる。

【0014】

本発明について上記の簡単に要約された特徴は、後述する本発明の詳細な説明の例示的な実施形態だけであり、本発明の範囲を限定するものではない。

【発明の効果】

【0015】

本発明によれば、アフィンモデルを用いるインター予測方法により、予測効率を向上させる効果がある。

10

【0016】

本発明によれば、サブロックの並進動きベクトルを用いてアフィンシードベクトルを導出することで、符号化効率を向上させる効果がある。

【0017】

本発明によれば、隣接ロックと現在ロックとの距離を2のべき級数に変換することでアフィンシードベクトルを導出することにより、符号化効率を向上させる効果がある。

【0018】

本発明で取得可能な効果は、上記効果に限定されず、本発明が属する技術分野において通常の知識を有する者は、下記記述により、言及されていない他の効果を明瞭に理解することができる。

20

【図面の簡単な説明】

【0019】

【図1】本発明の実施例によるビデオエンコーダ(encoder)を示すロック図である。

【図2】本発明の実施例によるビデオデコーダ(decoder)を示すロック図である。

【図3】本発明の実施例による基本符号化ツリーユニットを示す図である。

【図4】符号化ロックの複数の分割タイプを示す図である。

【図5】符号化ツリーユニットの分割例を示す図である。

30

【図6】本発明の実施例によるインター予測方法を示すフローチャートである。

【図7】対象の非線形動きを示す図である。

【図8】本発明の実施例によるアフィン動きに基づくインター予測方法を示すフローチャートである。

【図9】各アフィン動きモデルのアフィンシードベクトルの例を示す図である。

【図10】4パラメータ動きモデルにおけるサブロックのアフィンベクトルの例を示す図である。

【図11】マージ候補の導出に利用可能な隣接ロックを示す図である。

【図12】アフィン隣接ロックのアフィンシードベクトルに基づいて現在ロックのアフィンシードベクトルを導出することを示す図である。

40

【図13】サブロックの動きベクトルをアフィン隣接ロックのアフィンシードベクトルとする例を示す図である。

【図14】参照サンプルの位置を示す図である。

【図15】参照サンプルの位置を示す図である。

【図16】参照サンプルの位置を示す図である。

【図17】修正したアフィンマージベクトル導出方法の適用例を示す図である。

【図18】非アフィン隣接ロックに基づいて現在ロックのアフィンシードベクトルを導出する例を示す図である。

【図19】もう1つの隣接ロックで隣接ロックを置き換える例を示す図である。

【図20】複数のマージ候補を用いた動き補償予測方法を示すフローチャートである。

50

【図21】複数のマージ候補を用いた動き補償予測方法を示すフローチャートである。

【図22】本発明の実施例によるイントラ予測方法を示すフローチャートである。

【図23】各参照サンプルラインに含まれる参照サンプルを示す図である。

【図24】イントラ予測モードを示す図である。

【図25】参照サンプルを一行に配列した一次元配列例を示す図である。

【図26】参照サンプルを一行に配列した一次元配列例を示す図である。

【図27】角度イントラ予測モードとx軸に平行である直線との間で形成された角度を示す図である。

【図28】現在ブロックが非正方形である場合の予測サンプルの取得例を示す図である。

【図29】広角イントラ予測モードを示す図である。

【図30】ブロック強度の決定プロセスを示すフローチャートである。

【図31】事前定義されたフィルタ候補を示す図である。

【図32】画像を複数のタイルに分割する例を示す図である。

【図33】柔軟なタイル技術に基づく画像分割モードを示す図である。

【図34】タイルIDを各符号化ツリーユニットに割り当てる例を示す図である。

【図35】各タイルに対してループフィルタを適用するかどうかを選択的に決定する例を示す図である。

【発明を実施するための形態】

【0020】

以下、図面を参照しながら、本発明の実施例を詳しく説明する。

【0021】

ビデオの符号化及び復号化は、ブロックを単位として実行される。例えば、符号化ブロック、変換ブロック又は予測ブロックに対して、変換、量子化、予測、ループフィルタリング又は再構成などの符号化／復号化処理を行うことができる。

【0022】

以下、符号化／復号化されるべきブロックを「現在ブロック」と呼ぶ。例えば、現在符号化／復号化処理ステップによれば、現在ブロックは、符号化ブロック、変換ブロック又は予測ブロックを表すことができる。

【0023】

なお、本明細書で用いられる用語「ユニット」は、特定の符号化／復号化プロセスを実行するための基本ユニットを表し、また、「ブロック」は、所定のサイズのサンプルアレイを表すと理解されてもよい。特に明記しない限り、「ブロック」と「ユニット」は、相互交換可能に用いられる。例えば、後述される実施例において、符号化ブロック及び符号化ユニットは、同様な意味を有すると理解されてもよい。

【0024】

図1は、本発明の実施例によるビデオエンコーダ(encoder)を示すブロック図である。

【0025】

図1を参照すると、ビデオ符号化機器100は、画像分割部110、予測部120、125、変換部130、量子化部135、再配列部160、エントロピー符号化部165、逆量子化部140、逆変換部145、フィルタ部150及びメモリ155を備えてもよい。

【0026】

図1に示す各部材は、単独で示されてビデオ符号化機器における互いに異なる特徴的機能を表すが、各部材が分離したハードウェア又は单一のソフトウェアアセンブリで構成されることを表しない。つまり、説明しやすくするために、代表的な構成要素として示されて各構成要素を含み、また、少なくとも2つの構成要素を1つの構成要素に合わせてもよいか又は1つの構成要素を複数の構成要素に分けてこれにより機能を実行することができる。本発明の本質を逸脱しない限り、このような各構成要素を整合する実施例及び各構成要素を分離させる実施例も本発明の権利範囲に属する。

【0027】

10

20

30

40

50

なお、一部の構成要素は、本発明における本質機能を実行するための必須な構成要素ではなく、性能を向上させるための選択可能な構成要素だけである。本発明は、本発明の本質の実現に必要な部材（性能を向上させるための構成要素を含まない）のみを含むことで実行されてもよく、また、必要な構成要素（性能を向上させるための構成要素を含まない）のみを含む構造も本発明の権利範囲に属する。

【0028】

画像分割部110は、入力された画像を少なくとも1つの処理ユニットに分割することができる。この場合、処理ユニットは、予測ユニット（Prediction Unit：PU）であってもよく、変換ユニット（Transform Unit：TU）であってもよく、更に符号化ユニット（Coding Unit：CU）であってもよい。画像分割部110は、1つの画像を複数の符号化ユニット、予測ユニット及び変換ユニットの組み合わせに分割する。所定の基準（例えば、コスト関数）に基づいて符号化ユニット、予測ユニット及び変換ユニットの組み合わせを選択して画像を符号化することができる。

【0029】

例えば、1つの画像を複数の符号化ユニットに分割することができる。画像を符号化ユニットに分割するために、四分木構造（Quad Tree Structure）のような再帰的ツリー構造を用いて、1つのビデオ又は最大符号化ユニット（largest coding unit）をルートとして符号化ユニットを他の符号化ユニットに分割することができる。前記符号化ユニットは、分割された符号化ユニットの数と同じである数の子ノードを有してもよい。幾つかの制限によって分割されない符号化ユニットは、リーフノードとなる。つまり、1つの符号化ユニットが正方形分割だけを実現できると仮定する場合、1つの符号化ユニットを最大限4つの他の符号化ユニットに分割することができる。

【0030】

以下、本発明の実施例において、符号化ユニットは、符号化を実行するユニットを指してもよく、復号化を実行するユニットを指してもよい。

【0031】

1つの符号化ユニットにおける予測ユニットを少なくとも1つのサイズが同じである正方形又は矩形を分割することができ、1つの符号化ユニットにおける1つの予測ユニットを形状及び/又はサイズがもう1つの予測ユニットと異なるものに分割することもできる。

【0032】

符号化ユニットに基づいてイントラ予測を実行する予測ユニットが最小符号化ユニットではない場合、複数の予測ユニット $N \times N$ に分割する必要がなく、イントラ予測を実行することができる。

【0033】

予測部120、125は、インター予測を実行するインター予測部120及びイントラ予測を実行するイントラ予測部125を含んでもよい。予測ユニットに対して、インター予測を用いるかそれともイントラ予測を用いるかを決定することができ、また、各予測方法に基づいて、具体的な情報（例えば、イントラ予測モード、動きベクトル、参照画像など）を決定することができる。この場合、予測を実行する処理ユニットは、予測方法及び具体的なコンテンツを決定する処理ユニットと異なることがある。例えば、予測ユニットにより、予測方法及び予測モードなどを決定することができ、また、変換ユニットにより予測を実行することができる。生成された予測ブロックとオリジナルブロックとの間の残差値（残差ブロック）を変換部130に入力することができる。なお、予測のための予測モード情報、動きベクトル情報などを残差値と共にエンタロピー符号化部165において符号化してデコーダに伝送することができる。特定の符号化モードを用いる場合、予測部120、125により予測ブロックを生成することなく、直接的にオリジナルブロックを符号化してデコーダに伝送することもできる。

【0034】

インター予測部120は、現在画像の直前の画像又は直後の画像における少なくとも1

10

20

30

40

50

つの画像の情報に基づいて予測ユニットを予測することができる。幾つかの場合に、現在画像における符号化された一部の領域の情報に基づいて予測ユニットを予測することもできる。インター予測部 120 は、参照画像補間部、動き予測部、動き補償部を含んでもよい。

【0035】

参照画像補間部は、メモリ 155 から参照画像情報を受信し、また、参照画像から、整数画素又は小数画素の画素情報を生成することができる。輝度画素について言えば、1/4 画素を単位として小数画素の画素情報を生成するために、フィルタ係数が異なる DCT に基づく 8 タップ補間フィルタ (DCT-based Interpolation Filter) を用いることができる。色度信号について言えば、1/8 画素を単位として小数画素の画素情報を生成するために、フィルタ係数が異なる DCT に基づく 4 タップ補間フィルタ (DCT-based Interpolation Filter) を用いることができる。

【0036】

動き予測部は、参照画像補間部により補間された参照画像に基づいて、動き予測を実行することができる。動きベクトルを算出するための方法として、全探索ブロックマッチングアルゴリズム (FBMA: Full search-based Block Matching Algorithm)、3段階探索法 (TSS: Three Step Search)、新たな 3段階探索アルゴリズム (NTS: New Three-Step Search Algorithm) 等の複数の方法を用いることができる。補間された画素によれば、動きベクトルは、1/2 画素又は 1/4 画素を単位とした動きベクトル値を有してもよい。動き予測部において、異なる動き予測方法を用いることで現在予測ユニットを予測することができる。動き予測方法として、スキップ (Skip) 方法、マージ (Merge) 方法、アドバンスト動きベクトル予測 (AMVP: Advanced Motion Vector Prediction) 方法、イントラブロック複製 (Intra Block Copy) 方法などの複数の方法を用いることができる。

【0037】

イントラ予測部 125 は、現在ブロックの周辺の参照画素情報（前記参照画素情報は、現在画像における画素情報である）に基づいて、予測ユニットを生成することができる。現在予測ユニットの隣接ブロックは、インター予測を実行したブロックであり、且つ参照画素は、インター予測を実行した画素である場合、インター予測を実行したブロックに含まれる参照画素を、周辺のイントラ予測を実行したブロックの参照画素情報とすることができる。つまり、参照画素が利用不可能である場合、利用不可能な参照画素情報の代わりに、利用可能な参照画素のうちの少なくとも 1 つの参照画素を用いることができる。

【0038】

イントラ予測において、予測モードは、予測方向に基づいて参照画素情報を用いる角度予測モード及び予測を実行する場合に方向情報を使用しない非角度モードを有してもよい。輝度情報を予測するためのモードは、色度情報を予測するためのモードと異なってもよい。色度情報を予測するために、輝度情報を予測するためのイントラ予測モード情報又は予測された輝度信号情報を用いることができる。

【0039】

イントラ予測を実行する場合、予測ユニットのサイズが変換ユニットのサイズと同じであると、予測ユニットの左側に位置する画素、左上に位置する画素、上方に位置する画素に基づいて、予測ユニットに対してイントラ予測を実行することができる。しかしながら、イントラ予測を実行する場合、予測ユニットのサイズが変換ユニットのサイズと異なると、変換ユニットの参照画素に基づいてイントラ予測を実行することができる。なお、最小符号化ユニットのみに対して、N × N 分割を用いたイントラ予測を適用することができる。

【0040】

予測モードに基づいて参照画素に対して適応的イントラ平滑化 (AIS: Adapti

10

20

30

40

50

ve Intra Smoothing) フィルタを適用した後、イントラ予測方法で、予測ブロックを生成することができる。参照画素に適用される適応的イントラ平滑化フィルタのタイプは異なる可能性がある。イントラ予測方法を実行するために、現在予測ユニットの周辺に位置する予測ユニットのイントラ予測モードに基づいて、現在予測ユニットのイントラ予測モードを予測することができる。周辺予測ユニットから予測されたモード情報を用いて現在予測ユニットの予測モードを予測する場合、現在予測ユニットのイントラ予測モードが周辺予測ユニットと同じであると、所定のフラグ情報を用いて、現在予測ユニットの予測モードが周辺予測ユニットと同じであることを表す情報を伝送することができる。現在予測ユニットの予測モードが周辺予測ユニットと異なると、エントロピー符号化を実行することで、現在ブロックの予測モード情報を符号化することができる。

10

【0041】

なお、残差情報を含む残差ブロックを生成することができる。前記残差情報は、予測部 120、125 で生成された予測ユニットに基づいて予測を実行する予測ユニットと予測ユニットのオリジナルブロックとの差分値である。生成された残差ブロックを変換部 130 に入力することができる。

【0042】

変換部 130 は、離散コサイン変換 (DCT)、離散サイン変換 (DST) 又は変換スキップなどのような変換方法で、残差ブロックを変換することができる。前記残差ブロックは、オリジナルブロックと、予測部 120 及び 125 により生成された予測ユニットと、の間の残差情報を含む。残差ブロックを変換するために DCT を適用するか、DST を適用するかそれとも KLT を適用するかについて、残差ブロックを生成するための予測ユニットのイントラ予測モード情報を基づいて決定することができる。

20

【0043】

量子化部 135 は、変換部 130 で周波数領域に変換された値を量子化することができる。量子化係数は、ブロック又は画像の重要度によって変わってもよい。量子化部 135 で算出された値は、逆量子化部 140 及び再配列部 160 に提供されてもよい。

【0044】

再配列部 160 は、量子化された残差値に対して係数値の再配列を実行することができる。

【0045】

再配列部 160 は、係数走査 (Coefficient Scanning) 方法で、2 次元のブロック形状係数を 1 次元のベクトル形態に変更することができる。例えば、再配列部 160 は、ジグザグ走査 (Zig-Zag Scan) 方法で、DC 係数乃至高周波数領域の係数を走査し、それを 1 次元ベクトル形態に変換することができる。変換ユニットのサイズ及びイントラ予測モードによれば、ジグザグ走査の代わりに、列方向に沿って 2 次元ブロック形状係数を走査する垂直走査及び行方向に沿って 2 次元ブロック形状係数を走査する水平走査を用いることもできる。つまり、変換ユニットのサイズ及びイントラ予測モードに基づいて、ジグザグ走査、垂直方向走査及び水平方向走査のうちのどちらを用いるかを決定することができる。

30

【0046】

エントロピー符号化部 165 は、再配列部 160 で算出された値に基づいてエントロピー符号化を実行することができる。例えば、エントロピー符号化は、指数ゴロムコード (Exponential Golomb)、コンテキスト適応型可変長符号化 (CAVL C, Context-Adaptive Variable Length Coding)、コンテキスト適応型二値算術符号化 (CABAC: Context-Adaptive Binary Arithmetic Coding) 等の複数の符号化方法を用いることができる。

40

【0047】

エントロピー符号化部 165 は、再配列部 160 及び予測部 120、125 からの符号化ユニットの残差値係数情報及びブロックタイプ情報、予測モード情報、分割ユニット情

50

報、予測ユニット情報及び伝送ユニット情報、動きベクトル情報、参照フレーム情報、ブロックの補間情報、フィルタリング情報などの複数の情報を符号化することができる。

【0048】

エントロピー符号化部165は、再配列部160から入力された符号化ユニットの係数値に対してエントロピー符号化を行うことができる。

【0049】

逆量子化部140及び逆変換部145は、量子化部135により量子化された複数の値に対して逆量子化を行い、変換部130により変換された値に対して逆変換を行う。逆量子化部140及び逆変換部145で生成された残差値と、予測部120、125に含まれる動き予測部、動き補償部及びイントラ予測部により予測された予測ユニットと、をマージすることで、再構成ブロック(Reconstructed Block)を生成することができる。

10

【0050】

フィルタ部150は、デブロッキングフィルタ、オフセット補正部(offset correction unit)、適応型ループフィルタ(Adaptive Loop Filter)のうちの少なくとも1つを含んでもよい。

【0051】

デブロッキングフィルタは、ブロック間の境界により、再構成された画像で生成されたブロック歪みを除去することができる。デブロッキングを実行するかどうかを判定するために、ブロックに含まれる幾つかの列又は行に含まれる画素に基づいて、現在ブロックに対してデブロッキングフィルタを適用するかどうかを判定することができる。ブロックに対してデブロッキングフィルタを適用する場合、必要なデブロッキングフィルタリング強度に基づいて強フィルタ(Strong Filter)又は弱フィルタ(Weak Filter)を適用することができる。なお、デブロッキングフィルタを用いる過程において、垂直フィルタリング又は水平フィルタリングを実行する場合、水平方向フィルタリング及び垂直方向フィルタリングを同期して実行することができる。

20

【0052】

オフセット補正部は、画素単位で、デブロッキング画像とオリジナル画像とのオフセットを補正することができる。下記方式で、指定した画像に対してオフセット補正を行うことができる。画像に含まれる画素を所定の数の領域に分割した後、オフセット補正を必要とする領域を決定し、対応する領域に対してオフセット補正を適用するか又は各画素の縁情報を考慮してオフセット補正を適用する。

30

【0053】

フィルタリングされた再構成画像とオリジナル画像を比較した値に基づいて、適応的ループフィルタリング(Adaptive Loop Filtering)を実行することができる。画像に含まれる画素を所定のグループに分割した後、対応するグループに用いられる1つのフィルタを決定し、各グループに対してフィルタリングを差別的に実行することができる。各符号化ユニット(Coding Unit: CU)に対して、輝度信号を介して、適応的ループフィルタリングを適用するかどうかに関わる情報を伝送することができる。適用される適応的ループフィルタの形状及びフィルタ係数は、各ブロックによって異なってもよい。なお、適用されるブロックの特性に関わらず、同一タイプ(一定のタイプ)のALFを適用することもできる。

40

【0054】

メモリ155は、フィルタ部150により算出された再構成ブロック又はブロックを記憶することができ、インター予測を実行する場合、記憶された再構成ブロック又は画像を予測部120、125に提供することができる。

【0055】

図2は、本発明の実施例によるビデオデコーダ(decoder)を示すブロック図である。

【0056】

50

図2を参照すると、ビデオデコーダ200は、エントロピー復号化部210、再配列部215、逆量子化部220、逆変換部225、予測部230、予測部235、フィルタ部240、メモリ245を備えてもよい。

【0057】

ビデオエンコーダからビデオビットストリームを入力する場合、ビデオエンコーダと逆であるステップで、入力されたビットストリームを復号化することができる。

【0058】

エントロピー復号化部210は、ビデオエンコーダのエントロピー符号化部でエントロピー符号化を実行するステップと逆であるステップで、エントロピー復号化を実行することができる。例えば、ビデオエンコーダで実行される方法に対応するように、指數ゴロムコード(Exponential Golomb)、コンテキスト適応型可変長符号化(CAVLC: Context-Adaptive Variable Length Coding)、コンテキスト適応型二値算術符号化(CABAC: Context-Adaptive Binary Arithmetic Coding)等の複数の方法を適用することができる。

10

【0059】

エントロピー復号化部210は、エンコーダにより実行されたイントラ予測及びインター予測に関わる情報に対して復号化を行うことができる。

【0060】

再配列部215は、符号化部においてエントロピー復号化部210によりエントロピー復号化されたビットストリームを再配列する方法で再配列を実行することができる。1次元ベクトル形態で表す複数の係数を2次元ブロック形状の係数に再構成することで再配列を行うことができる。再配列部215は、下記方式で再配列を実行することができる。符号化部により実行された係数走査に関わる情報を受信し、対応する符号化部により実行された走査順に応じて逆走査を行う。

20

【0061】

逆量子化部220は、エンコーダにより提供された量子化パラメータ及び再配列されたブロックの係数値に基づいて逆量子化を実行することができる。

【0062】

ビデオエンコーダにより実行された量子化結果について、逆変換部225は、変換部により実行された変換であるDCT、DST及びKLTに対して、逆変換を実行することができる。つまり、逆DCT、逆DST及び逆KLTを実行する。逆変換は、ビデオエンコーダで決定された伝送ユニットにより実行されてもよい。ビデオデコーダの逆変換部225において、予測方法、現在ブロックのサイズ及び予測方向などの複数の情報に基づいて、変換方法(例えば、DCT、DST、KLT)を選択的に実行することができる。

30

【0063】

予測部230、235は、エントロピー復号化部210により提供された、予測ブロックの生成に関わる情報及びメモリ245により提供された先に復号化されたブロック又は画像情報に基づいて、予測ブロックを生成することができる。

【0064】

上述したように、ビデオエンコーダにおける操作方式と同じである方式でイントラ予測を実行する場合、予測ユニットのサイズが変換ユニットのサイズと同じであると、予測ユニットの左側に位置する画素、左上に位置する画素、上方に位置する画素に基づいて、予測ユニットに対してイントラ予測を実行することができる。イントラ予測を実行する場合、予測ユニットのサイズが変換ユニットのサイズと異なると、変換ユニットの参照画素に基づいてイントラ予測を実行することができる。なお、最小符号化ユニットのみに対して、N×N分割を用いたイントラ予測を適用することができる。

40

【0065】

予測部230、235は、予測ユニット判定部、インター予測部及びイントラ予測部を含んでもよい。予測ユニット判定部は、エントロピー復号化部210から入力された予測

50

ユニット情報、イントラ予測方法の予測モード情報、インター予測方法の動き予測関連情報などの複数の情報を受信し、現在符号化ユニットに基づいて、予測ユニットを分類し、予測ユニットがインター予測を実行しているかそれともイントラ予測を実行しているかを決定する。イントラ予測部 230 は、ビデオエンコーダにより提供された、現在予測ユニットのインター予測の実行に必要な情報を用いて、現在予測ユニットが属する現在画像の直前の画像又は直後の画像のうちの少なくとも 1 つの画像に含まれる情報に基づいて、現在予測ユニットに対してインター予測を実行することができる。あるいは、現在予測ユニットが属する現在画像における再構成された一部の領域の情報に基づいて、インター予測を実行することもできる。

【0066】

10

インター予測を実行するために、符号化ユニットに基づいて、対応する符号化ユニットに含まれる予測ユニットの動き予測方法がスキップモード (S k i p M o d e) 、マージモード (M e r g e M o d e) 、アドバンスド動きベクトル予測モード (A M V P M o d e) 、イントラブロック複製モードのうちのどちらであるかを判定することができる。

【0067】

イントラ予測部 235 は、現在画像における画素情報に基づいて予測ブロックを生成することができる。予測ユニットが、イントラ予測を実行した予測ユニットである場合、ビデオエンコーダから提供された予測ユニットのイントラ予測モード情報に基づいて、イントラ予測を実行することができる。イントラ予測部 235 は、適応的イントラ平滑化 (A I S : A d a p t i v e I n t r a S m o o t h i n g) フィルタ、参照画素補間部、D C フィルタを含んでもよい。適応的イントラ平滑化フィルタは、現在ブロックの参照画素に対してフィルタリングを実行する部分であり、また、現在予測ユニットの予測モードに基づいて、フィルタを適用するかどうかを決定することができる。ビデオエンコーダから提供された予測ユニットの予測モード及び適応的イントラ平滑化フィルタ情報を用いて、現在ブロックの参照画素に対して適応的イントラ平滑化フィルタリングを実行することができる。現在ブロックの予測モードが、適応的イントラ平滑化フィルタリングを実行しないモードであると、適応的イントラ平滑化フィルタを適用しなくてもよい。

【0068】

20

予測ユニットの予測モードが、参照画素に対して補間を行う画素値に基づいてイントラ予測を実行する予測ユニットであると、参照画素補間部は、参照画素に対して補間を行うことで、整数値又は小数値の画素単位の参照画素を生成することができる。現在予測ユニットの予測モードが、参照画素に対して補間を行わない方式で予測ブロックを生成する予測モードであると、参照画素に対して補間を行わなくてもよい。現在ブロックの予測モードが D C モードであると、D C フィルタは、フィルタリングにより予測ブロックを生成することができる。

【0069】

30

再構成されたブロック又は画像をフィルタ部 240 に提供することができる。フィルタ部 240 は、デブロッキングフィルタ、オフセット補正部、A L F を含んでもよい。

【0070】

40

ビデオエンコーダから、対応するブロック又は画像に対してデブロッキングフィルタを適用するかどうかに関わる情報及びデブロッキングフィルタを適用する場合に強フィルタを用いるかそれとも弱フィルタを用いるかに関わる情報を受信することができる。ビデオデコーダのデブロッキングフィルタは、ビデオエンコーダから提供された、デブロッキングフィルタに関わる情報を受信することができ、また、ビデオデコーダは、対応するブロックに対してデブロッキングフィルタリングを実行することができる。

【0071】

オフセット補正部は、符号化を行う場合に画像に適用されるオフセット補正のタイプ及びオフセット情報に基づいて、再構成された画像に対してオフセット補正を実行することができる。

【0072】

50

エンコーダから提供された、ALFを適用するかどうかに関わる情報、ALF係数情報等に基づいて、ALFを符号化ユニットに適用することができる。このようなALF情報は、特定のパラメータ集合に含まれることで提供されてもよい。

【0073】

メモリ245は、再構成された画像又はブロックを記憶し、前記画像又はブロックを参照画像又は参照ブロックとして利用できるようになり、また、再構成された画像を出力部に提供することができる。

【0074】

図3は、本発明の実施例による基本符号化ツリーユニットを示す図である。

【0075】

最大サイズの符号化ブロックを符号化ツリーブロックと定義することができる。1つの画像は、複数の符号化ツリーユニット(Coding Tree Unit:CTU)に分割されてもよい。符号化ツリーユニットは、最大サイズの符号化ユニットであり、最大符号化ユニット(LCU:Largest Coding Unit)と呼ばれてもよい。図3は、1つの画像を複数の符号化ツリーユニットに分割する例を示す。

【0076】

符号化ツリーユニットのサイズは、画像レベル又はシーケンスレベルで定義されてもよい。このため、画像パラメータ集合又はシーケンスパラメータ集合により、信号を用いて、符号化ツリーユニットのサイズを示す情報を送信することができる。

【0077】

例えば、シーケンスにおける画像全体の符号化ツリーユニットのサイズを 128×128 とすることができる。あるいは、画像レベルの 128×128 又は 256×256 のうちのいずれか1つを符号化ツリーユニットのサイズと決定することができる。例えば、第1画像における符号化ツリーユニットのサイズを 128×128 とし、また、第2画像における符号化ツリーユニットのサイズを 256×256 とすることができる。

【0078】

符号化ツリーユニットの分割により符号化ブロックを生成することができる。符号化ブロックは、符号化／復号化処理を行うための基本ユニットを表す。例えば、異なる符号化ブロックに応じて予測又は変換を実行することができ、又は、異なる符号化ブロックに応じて予測符号化モードを決定することができる。ここで、予測符号化モードは、予測画像を生成する方法を表す。例えば、予測符号化モードは、画面内予測(Intra Prediction: イントラ予測)、画面間予測(Inter Prediction: インター予測)、現在画像参照(Current Picture Referencing: CPR)、イントラブロック複製(Intra Block Copy:IBC)又は組み合わせ予測(Combined Prediction)を含んでもよい。符号化ブロックについて、イントラ予測、インター予測、現在画像参照又は組み合わせ予測のうちの少なくとも1つの予測符号化モードを用いて、符号化ブロックに関わる予測ブロックを生成することができる。

【0079】

ビットストリームを介して、信号で、現在ブロックの予測符号化モードを表す情報を送信することができる。例えば、前記情報は、予測符号化モードがイントラモードであるかそれともインターモードであるかを示す1ビットフラグであってもよい。現在ブロックの予測符号化モードがインターモードであると決定された場合のみ、現在画像参照又は組み合わせ予測を用いることができる。

【0080】

現在画像参照は、現在画像を参照画像とし、現在画像における符号化／復号化された領域から現在ブロックの予測ブロックを取得するために用いられる。ここで、現在画像は、現在ブロックを含む画像を意味する。ビットストリームを介して、信号で、現在ブロックに対して現在画像参照を適用するかどうかを表す情報を送信することができる。例えば、前記情報は、1ビットフラグであってもよい。前記フラグが真である場合、現在ブロック

10

20

30

40

50

の予測符号化モードを現在画像参照として決定することができる。前記フラグが偽である場合、現在ブロックの予測モードをインター予測として決定することができる。

【0081】

あるいは、参照画像インデックスに基づいて、現在ブロックの予測符号化モードを決定することができる。例えば、参照画像インデックスが現在画像を指示する場合、現在ブロックの予測符号化モードを現在画像参照として決定することができる。参照画像インデックスが現在の画像ではない他の画像を指示する場合、現在ブロックの予測符号化モードをインター予測として決定することができる。つまり、現在画像参照は、現在画像における符号化／復号化された領域の情報を用いる予測方法であり、また、インター予測は、符号化／復号化された他の画像の情報を用いる予測方法である。

10

【0082】

組み合わせ予測は、イントラ予測、インター予測及び現在画像参照のうちの2つ以上を組み合わせることで形成された符号化モードである。例えば、組み合わせ予測を適用する場合、イントラ予測、インター予測又は現在画像参照のうちの1つに基づいて、第1予測ブロックを生成することができ、また、もう1つに基づいて、第2予測ブロックを生成することができる。第1予測ブロック及び第2予測ブロックを生成する場合、第1予測ブロック及び第2予測ブロックの平均演算及び加重加算により最終的予測ブロックを生成することができる。ビットストリームを介して、信号で、組み合わせ予測を適用するかどうかを表す情報を送信することができる。前記情報は、1ビットフラグであってもよい。

【0083】

20

図4は、符号化ブロックの複数の分割タイプを示す図である。

【0084】

四分木分割、二分木分割又は三分木分割に基づいて、符号化ブロックを複数の符号化ブロックに分割することができる。また、四分木分割、二分木分割又は三分木分割に基づいて、分割された符号化ブロックを更に複数の符号化ブロックに分割することもできる。

【0085】

四分木分割は、現在ブロックを4つのブロックに分割する分割技術である。四分木分割の結果として、現在ブロックを4つの正方形パーティション（図4の（a）における「SPLIT_QT」を参照する）に分割することができる。

【0086】

30

二分木分割は、現在ブロックを2つのブロックに分割する分割技術である。垂直方向に沿って（即ち、現在ブロックを横断する垂直線を用いる）現在ブロックを2つのブロックに分割する過程を垂直方向二分木分割と呼び、また、水平方向に沿って（即ち、現在ブロックを横断する水平線を用いる）現在ブロックを2つのブロックに分割する過程を水平方向二分木分割と呼ぶことができる。二分木分割の結果として、現在ブロックを2つの非正方形パーティションに分割することができる。図4の（b）における「SPLIT_BT_VER」は、垂直方向二分木分割結果を表し、また、図4（c）における「SPLIT_BT_HOR」は、水平方向二分木分割結果を表す。

【0087】

三分木分割は、現在ブロックを3つのブロックに分割する分割技術である。垂直方向に沿って（即ち、現在ブロックを横断する2本の垂直線を用いる）現在ブロックを3つのブロックに分割する過程を垂直方向三分木分割と呼び、また、水平方向に沿って（即ち、現在ブロックを横断する2本の水平線）現在ブロックを3つのブロックに分割する過程を水平方向三分木分割と呼ぶことができる。三分木分割結果として、現在ブロックを3つの非正方形パーティションに分割することができる。この場合、現在ブロックの中心に位置するパーティションの幅／高さは、他のパーティションの幅／高さの2倍であってもよい。図4（d）における「SPLIT_TT_VER」は、垂直方向三分木分割結果を表し、また、図4（e）における「SPLIT_TT_HOR」は、水平方向三分木分割結果を表す。

40

【0088】

50

符号化ツリーユニットの分割回数を分割深度 (Partitioning Depth) と定義することができる。シーケンス又は画像レベルで、符号化ツリーユニットの最大分割深度を決定することができる。従って、符号化ツリーユニットの最大分割深度は、シーケンス又は画像によって異なってもよい。

【 0 0 8 9 】

あるいは、複数の分割技術のそれぞれに対して最大分割深度を単独で決定することができる。例えば、四分木分割を許容する最大分割深度は、二分木分割及び / 又は三分木分割を許容する最大分割深度と異なってもよい。

【 0 0 9 0 】

エンコーダは、ビットストリームを介して、信号で、現在ブロックの分割タイプ又は分割深度のうちの少なくとも 1 つを表す情報を送信することができる。デコーダは、ビットストリームから解析された情報に基づいて、符号化ツリーユニットの分割タイプ及び分割深度を決定することができる。

10

【 0 0 9 1 】

図 5 は、符号化ツリーユニットの分割例を示す図である。

【 0 0 9 2 】

四分木分割、二分木分割及び / 又は三分木分割などの分割技術で符号化ブロックを分割する過程を多分木分割 (Multi Tree Partitioning) と呼ぶことができる。

20

【 0 0 9 3 】

符号化ブロックに対して多分木分割を適用することで生成された符号化ブロックを下流符号化ブロックと呼ぶことができる。符号化ブロックの分割深度が k である場合、複数の下流符号化ブロックの分割深度を $k + 1$ とする。

【 0 0 9 4 】

一方で、分割深度が $k + 1$ である符号化ブロックについて、分割深度が k である符号化ブロックを上流符号化ブロックと呼ぶことができる。

【 0 0 9 5 】

上流符号化ブロックの分割タイプ又は隣接符号化ブロックの分割タイプのうちの少なくとも 1 つに基づいて、現在符号化ブロックの分割タイプを決定することができる。ここで、隣接符号化ブロックは、現在符号化ブロックに隣接し、それは、現在符号化ブロックの上隣接ブロック、左隣接ブロック又は左上隅に隣接する隣接ブロックのうちの少なくとも 1 つを含んでもよい。ここで、分割タイプは、四分木分割を行うかどうか、二分木分割を行うかどうか、二分木分割方向、三分木分割を行うかどうか又は三分木分割方向のうちの少なくとも 1 つを含んでもよい。

30

【 0 0 9 6 】

符号化ブロックの分割タイプを決定するために、ビットストリームを介して、信号で、符号化ブロックが分割されたかどうかを表す情報を送信することができる。前記情報は、1 ビットフラグ「 s p l i t _ c u _ f l a g 」であり、前記フラグが真である場合、多分木分割技術で符号化ブロックを分割することを示す。

【 0 0 9 7 】

「 s p l i t _ c u _ f l a g 」が真である場合、ビットストリームを介して、信号で、符号化ブロックが四分木分割されたかどうかを表す情報を送信することができる。前記情報は、1 ビットフラグ「 s p l i t _ q t _ f l a g 」であり、前記フラグが真である場合、符号化ブロックは、4 つのブロックに分割されてもよい。

40

【 0 0 9 8 】

例えば、図 5 に示す例において、符号化ツリーユニットが四分木分割された場合に分割深度が 1 である 4 つの符号化ブロックを生成することが図示される。なお、四分木分割結果として生成された 4 つの符号化ブロックのうちの 1 番目の符号化ブロック及び 4 番目の符号化ブロックに対して再び四分木分割を適用することを図示する。最終的に、分割深度が 2 である 4 つの符号化ブロックを生成することができる。

50

【0099】

なお、分割深度が2である符号化ブロックに対して再び四分木分割を適用することで分割深度が3である符号化ブロックを生成することができる。

【0100】

符号化ブロックに対して四分木分割を適用していない場合、符号化ブロックのサイズ、符号化ブロックが画像の境界に位置するかどうか、最大分割深度又は隣接ブロックの分割タイプのうちの少なくとも1つを考慮することで、前記符号化ブロックに対して二分木分割を実行するかそれとも三分木分割を実行するかを決定することができる。前記符号化ブロックに対して二分木分割又は三分木分割を実行すると決定した場合、ビットストリームを介して、信号で、分割方向を表す情報を送信することができる。前記情報は、1ビットフラグ「m t t _ s p l i t _ c u _ v e r t i c a l _ f l a g」であってもよい。前記フラグに基づいて、分割方向が垂直方向であるかそれとも水平方向であるかを決定することができる。なお、ビットストリームを介して、信号で、前記符号化ブロックに対して二分木分割又は三分木分割のうちのどちらを適用するかを表す情報を送信することができる。前記情報は、1ビットフラグ「m t t _ s p l i t _ c u _ b i n a r y _ f l a g」であってもよい。前記フラグに基づいて、前記符号化ブロックに対して二分木分割を適用するかそれとも三分木分割を適用するかを決定することができる。

10

【0101】

例えば、図5に示す例において、分割深度が1である符号化ブロックに対して垂直方向二分木分割を適用することを図示する。前記分割結果として生成された符号化ブロックにおける左側符号化ブロックに対して垂直方向三分木分割を適用し、右側符号化ブロックに対して垂直方向二分木分割を適用する。

20

【0102】

インター予測は、直前の画像の情報を用いて現在ブロックを予測する予測符号化モードである。例えば、直前の画像における、現在ブロックと同一の位置のブロック（以下、コロケーテッドブロックと呼ぶ。C o l l o c a t e d b l o c k）を現在ブロックの予測ブロックとすることができる。以下、位置が現在ブロックと同じであるブロックに基づいて生成された予測ブロックをコロケーテッド予測ブロック（C o l l o c a t e d P r e d i c t i o n B l o c k）と呼ぶことができる。

30

【0103】

一方で、直前の画像に存在する対象が現在画像における他の位置に移動した場合、対称の動きにより、現在ブロックを効果的に予測することができる。例えば、直前の画像と現在画像を比較することで対象の移動方向及びサイズを了解できると、対象の動き情報を考慮して現在ブロックの予測ブロック（又は予測画像）を生成することができる。以下、動き情報を用いることで生成された予測ブロックを動き予測ブロックと呼ぶことができる。

【0104】

現在ブロックから予測ブロックを減算することで残差ブロック（r e s i d u a l b l o c k）を生成することができる。この場合、対象の動きが存在する場合、動き予測ブロックの代わりにコロケーテッド予測ブロックを用いることで、残差ブロックのエネルギーを減少させ、高残差ブロックの圧縮性能を向上させることができる。

40

【0105】

上述したように、動き情報を用いて予測ブロックを生成する過程を動き補償予測と呼ぶことができる。大部分のインター予測において、動き補償予測に基づいて予測ブロックを生成することができる。

【0106】

動き情報は、動きベクトル、参照画像インデックス、予測方向又は双方向加重値インデックスのうちの少なくとも1つを含んでもよい。動きベクトルは、対象の移動方向及びサイズを表す。参照画像インデックスは、参照画像リストに含まれる複数の参照画像において現在ブロックの参照画像を指定する。予測方向は、一方向L0予測、一方向L1予測又は双方向予測（L0予測及びL1予測）のうちのいずれか1つを指す。現在ブロックの予

50

測方向に基づいて、L0方向の動き情報又はL1方向の動き情報のうちの少なくとも1つを用いることができる。双方向加重値インデックスは、L0予測ブロックに適用される加重値及びL1予測ブロックに適用される加重値を指定する。

【0107】

図6は、本発明の実施例によるインター予測方法を示すフローチャートである。

【0108】

図6を参照すると、インター予測方法は、現在ブロックのインター予測モードを決定するステップ(S601)と、決定されたインター予測モードに基づいて、現在ブロックの動き情報を得るステップ(S602)と、得られた動き情報に基づいて、現在ブロックに対して動き補償予測を実行するステップ(S603)と、を含む。

10

【0109】

ここで、インター予測モードは、現在ブロックの動き情報を決定するための複数の技術を表し、また、並進(Translation)動き情報を用いたインター予測モード及びアフィン(Affine)動き情報を用いたインター予測モードを含んでもよい。例えば、並進動き情報を用いたインター予測モードは、マージモード及びアドバンスド動きベクトル予測モードを含んでもよい。アフィン動き情報を用いたインター予測モードは、アフィンマージモード及びアフィン動きベクトル予測モードを含んでもよい。インター予測モードによれば、現在ブロックに隣接する隣接ブロック又はビットストリームから解析された情報に基づいて、現在ブロックの動き情報を決定することができる。

【0110】

20

以下、アフィン動き情報を用いたインター予測方法を詳しく説明する。

【0111】

図7は、対象の非線形動きを示す図である。

【0112】

ビデオ内の対称の動きは、非線形動きである可能性がある。例えば、図7に示す例のように、カメラ拡大(Zoom-in)、縮小(Zoom-out)、回転(Rotation)又はアフィン変換などの対称の非線形動きを発生する可能性がある。対象の非線形動きを発生した場合、並進動きベクトルにより対象の動きを有効に表すことができない。従って、対象の非線形動きを発生した部分において、並進動きの代わりに、アフィン動きを用いることで、符号化効率を向上させることができる。

30

【0113】

図8は、本発明の実施例によるアフィン動きに基づくインター予測方法を示すフローチャートである。

【0114】

ビットストリームから解析された情報に基づいて、現在ブロックに対してアフィン動きに基づくインター予測技術を適用するかどうかを決定することができる。具体的には、現在ブロックに対してアフィンマージモードを適用するかどうかを示すフラグ又は現在ブロックに対してアフィン動きベクトル予測モードを適用するかどうかを示すフラグのうちの少なくとも1つに基づいて、現在ブロックに対してアフィン動きに基づくインター予測技術を適用するかどうかを決定することができる。

40

【0115】

アフィン動きに基づくインター予測技術を現在ブロックに適用する場合、現在ブロックのアフィン動きモデルを決定することができる(S801)。アフィン動きモデルは、6パラメータアフィン動きモデル又は4パラメータアフィン動きモデルのうちの少なくとも1つと決定されてもよい。6パラメータアフィン動きモデルは、6つのパラメータでアフィン動きを表し、また、4パラメータアフィン動きモデルは、4つのパラメータでアフィン動きを表す。

【0116】

等式1は、6つのパラメータでアフィン動きを表す場合である。アフィン動きは、アフィンシードベクトルにより決定された所定領域の並進動きを表す。

50

【数1】

【等式1】

$$v_x = ax - by + e$$

$$v_y = cx + dy + f$$

【0117】

10

6つのパラメータでアフィン動きを表す場合、複雑な動きを表すことができるが、各パラメータに対して符号化を行う場合に必要なビット数は増加するため、符号化効率を低下させる。従って、4つのパラメータで、アフィン動きを表すこともできる。等式2は、4つのパラメータで、アフィン動きを表す場合である。

【数2】

【等式2】

$$v_x = ax - by + e$$

$$v_y = bx + ay + f$$

20

【0118】

30

現在ブロックのアフィン動きモデルを決定するための情報に対して符号化を行い、ビットストリームを介して、信号で送信することができる。例えば、前記情報は、1ビットフラグ「`a f f i n e_t y p e_f l a g`」であってもよい。前記フラグの値が0である場合、4パラメータアフィン動きモデルを適用することを表す。前記フラグの値が1である場合、6パラメータアフィン動きモデルを適用することを表す。前記フラグは、スライス(`s l i c e`)、タイル又はブロック(例えば、符号化ブロック又は符号化ツリーユニット)を単位として符号化を行うことができる。スライスレベルで、信号を用いてフラグを送信する場合、前記スライスに属する全てのブロックに対して、前記スライスレベルで決定されたアフィン動きモデルを適用することができる。

【0119】

あるいは、現在ブロックのアフィンインター予測モードに基づいて、現在ブロックのアフィン動きモデルを決定することができる。例えば、アフィンマージモードを適用する場合、現在ブロックのアフィン動きモデルを4パラメータ動きモデルとして決定することができる。一方で、アフィン動きベクトル予測モードを適用する場合、現在ブロックのアフィン動きモデルを決定するための情報に対して符号化を行い、ビットストリームを介して信号で送信することができる。例えば、現在ブロックに対してアフィン動きベクトル予測モードを適用する場合、1ビットフラグ「`a f f i n e_t y p e_f l a g`」に基づいて、現在ブロックのアフィン動きモデルを決定することができる。

40

【0120】

次に、現在ブロックのアフィンシードベクトルを導出することができる(S802)。4パラメータアフィン動きモデルを選択した場合、現在ブロックの2つの制御点での動きベクトルを導出することができる。なお、6パラメータアフィン動きモデルを選択した場合、現在ブロックの3つの制御点での動きベクトルを導出することができる。制御点での動きベクトルをアフィンシードベクトルと呼ぶことができる。制御点は、現在ブロックの左上隅、右上隅又は左下隅のうちの少なくとも1つを含んでもよい。

【0121】

50

図9は、各アフィン動きモデルのアフィンシードベクトルの例を示す図である。

【0122】

4パラメータアフィン動きモデルにおいて、左上隅、右上隅又は左下隅のうちの2つに関わるアフィンシードベクトルを導出することができる。例えば、図9の(a)に示す例のように、4パラメータアフィン動きモデルを選択した場合、現在ブロックの左上隅(例えば、左上サンプル(x_0, y_0))に関わるアフィンシードベクトル sv_0 及び現在ブロックの右上隅(例えば、右上サンプル(x_1, y_1))に関わるアフィンシードベクトル sv_1 を用いてアフィンベクトルを導出することができる。また、左上隅に関わるアフィンシードベクトルの代わりに、左下隅に関わるアフィンシードベクトルを用いることもできる。又は、右上隅に関わるアフィンシードベクトルの代わりに、左下隅に関わるアフィンシードベクトルを用いることもできる。

10

【0123】

6パラメータアフィン動きモデルにおいて、左上隅、右上隅及び左下隅に関わるアフィンシードベクトルを導出することができる。例えば、図9の(b)に示す例のように、6パラメータアフィン動きモデルを選択し場合、現在ブロックの左上隅(例えば、左上サンプル(x_0, y_0))に関わるアフィンシードベクトル sv_0 、現在ブロックの右上隅(例えば、右上サンプル(x_1, y_1))に関わるアフィンシードベクトル sv_1 及び現在ブロックの左上隅(例えば、左上サンプル(x_2, y_2))に関わるアフィンシードベクトル sv_2 を用いてアフィンベクトルを導出することができる。

20

【0124】

後述する実施例において、4パラメータアフィン動きモデルで、左上制御点及び右上制御点のアフィンシードベクトルをそれぞれ第1アフィンシードベクトル及び第2アフィンシードベクトルと呼ぶ。後述する、第1アフィンシードベクトル及び第2アフィンシードベクトルを用いる実施例において、第1アフィンシードベクトル及び第2アフィンシードベクトルのうちの少なくとも1つを左下制御点のアフィンシードベクトル(第3アフィンシードベクトル)又は右下制御点のアフィンシードベクトル(第4アフィンシードベクトル)に置き換えることができる。

20

【0125】

なお、6パラメータアフィン動きモデルで、左上制御点、右上制御点及び左下制御点のアフィンシードベクトルをそれぞれ第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルと呼ぶ。後述する、第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルを用いる実施例において、第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルのうちの少なくとも1つを右下制御点のアフィンシードベクトル(第4アフィンシードベクトル)に置き換えることができる。

30

【0126】

アフィンシードベクトルを用いて、異なるサブブロックに対して、アフィンベクトルを導出することができる(S803)。ここで、アフィンベクトルは、アフィンシードベクトルに基づいて導出された並進動きベクトルを表す。サブブロックのアフィンベクトルをアフィンサブブロック動きベクトル又はサブブロック動きベクトルと呼ぶことができる。

40

【0127】

図10は、4パラメータ動きモデルにおけるサブブロックのアフィンベクトルの例を示す図である。

【0128】

制御点の位置、サブブロックの位置及びアフィンシードベクトルに基づいてサブブロックのアフィンベクトルを導出することができる。例えば、等式3は、アフィンサブブロックベクトルの導出例を示す。

【数3】

50

【等式3】

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

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

10

【0129】

等式3において、(x, y)は、サブブロックの位置を表す。ここで、サブブロックの位置は、サブブロックに含まれる参照サンプルの位置を表す。参照サンプルは、サブブロックの左上隅に位置するサンプルであってもよく、又はそのx軸又はy軸座標における少なくとも1つの、中心位置に位置するサンプルであってもよい。(x₀, y₀)は、第1制御点の位置を表し、(sv_{0x}, sv_{0y})は、第1アフィンシードベクトルを表す。なお、(x₁, y₁)は、第2制御点の位置を表し、(sv_{1x}, sv_{1y})は、第2アフィンシードベクトルを表す。

【0130】

第1制御点及び第2制御点はそれぞれ現在ブロックの左上隅及び右上隅に対応する場合、x₁ - x₀を現在ブロックの幅と同じである値と設定することができる。

20

【0131】

続いて、各サブブロックのアフィンベクトルを用いて各サブブロックに対して動き補償予測を実行することができる(S804)。動き補償予測を実行した後、各サブブロックに関わる予測ブロックを生成することができる。サブブロックの予測ブロックを現在ブロックの予測ブロックと設定することができる。

【0132】

現在ブロックに隣接する隣接ブロックのアフィンシードベクトルに基づいて、現在ブロックのアフィンシードベクトルを導出することができる。現在ブロックのインター予測モードがアフィンマージモードである場合、マージ候補リストに含まれるマージ候補のアフィンシードベクトルを現在ブロックのアフィンシードベクトルとして決定することができる。なお、現在ブロックのインター予測モードがアフィンマージモードである場合、現在ブロックの参照画像インデックス、特定方向予測フラグ又は双方向加重値のうちの少なくとも1つを含む動き情報をマージ候補と同じであるように設定することができる。

30

【0133】

現在ブロックの隣接ブロックに基づいてマージ候補を導出することができる。前記隣接ブロックは、現在ブロックに空間的に隣接する空間的隣接ブロック及び現在画像と異なる画像に含まれる時間的隣接ブロックのうちの少なくとも1つを含んでもよい。

【0134】

図11は、マージ候補の導出に利用可能な隣接ブロックを示す図である。

40

【0135】

現在ブロックの隣接ブロックは、現在ブロックの左側に隣接する隣接ブロックの(A)、現在ブロックの情報に隣接する隣接ブロックの(B)、現在ブロックの右上隅に隣接する隣接ブロック(C)、現在ブロックの左下隅に隣接する隣接ブロック(D)又は現在ブロックの左上隅に隣接する隣接ブロックのうちの少なくとも1つを含んでもよい。現在ブロックの左上サンプルの座標が(x₀, y₀)であると、左隣接ブロックAは、(x₀ - 1, y₀ + H - 1)位置でのサンプルを含み、また、上隣接ブロックBは、(x₀ + W - 1, y₀ - 1)位置でのサンプルを含む。ここで、W及びHは、それぞれ、現在ブロックの幅及び高さを表す。右上隣接ブロックCは、(x₀ + W, y₀ - 1)位置でのサンプル

50

を含み、また、左下隣接ブロック D は、($x 0 - 1$, $y 0 + H$) 位置でのサンプルを含む。左上隣接ブロック E は、($x 0 - 1$, $y 0 - 1$) 位置でのサンプルを含む。

【 0 1 3 6 】

アフィンインター予測モードで、隣接ブロックに対して符号化を行う場合、対応する隣接ブロックのアフィンシードベクトルに基づいて、マージ候補のアフィンシードベクトルを導出することができる。後述において、アフィンインター予測モードで符号化される隣接ブロックは、アフィン隣接ブロックと呼ばれる。

【 0 1 3 7 】

事前定義された走査順に応じて隣接ブロックを探索することで、現在ブロックのマージ候補を生成することができる。エンコーダ及びデコーダにおいて走査順を事前定義することができる。例えば、A、B、C、D、E の順番に応じて隣接ブロックを探索することができる。なお、探索されたアフィン隣接ブロックから、マージ候補を順に導出することができる。あるいは、現在ブロックのサイズ、形状又はアフィン動きモデルのうちの少なくとも 1 つに基づいて、走査順を適応的に決定することができる。つまり、異なるサイズ、形状又はアフィン動きモデルのうちの少なくとも 1 つを有するブロックの走査順は異なる。

10

【 0 1 3 8 】

あるいは、現在ブロックの上方に位置するブロックを順に探索し、最初に発見されたアフィン隣接ブロックからマージ候補を導出し、また、現在ブロックの左側に位置するブロックを順に探索し、最初に発見されたアフィン隣接ブロックからマージ候補を導出する。ここで、現在ブロックの上方に位置する複数の隣接ブロックは、隣接ブロック E、隣接ブロック B 又は隣接ブロック C のうちのすくなくとも 1 つを含み、現在ブロックの左側に位置する複数のブロックは、ブロック A 又はブロック D のうちの少なくとも 1 つを含んでもよい。この場合、隣接ブロック E を現在ブロックの左側に位置するブロックと分類してもよい。

20

【 0 1 3 9 】

図示されていないが、現在ブロックの時間的隣接ブロックからマージ候補を導出することができる。ここで、時間的隣接ブロックは、コロケートド画像における現在ブロックと同一の位置に位置するブロック又はそれに隣接するブロックを含んでもよい。具体的には、アフィンインター予測モードで、現在ブロックの時間的隣接ブロックに対して符号化を行う場合、時間的マージ候補のアフィンシードベクトルに基づいてマージ候補を導出することができる。

30

【 0 1 4 0 】

マージ候補を含むマージ候補リストを生成することができ、また、マージ候補リストに含まれるマージ候補のうちの 1 つのアフィンシードベクトルを現在ブロックのアフィンシードベクトルとして決定することができる。このために、複数のマージ候補のうちのいずれか 1 つを標識するインデックス情報に対して符号化を行い、ビットストリームを介して伝送することができる。

【 0 1 4 1 】

もう 1 つの例として、走査順に応じて複数の隣接ブロックを探索することができ、また、最初に発見されたアフィン隣接ブロックのアフィンシードベクトルから現在ブロックのアフィンシードベクトルを導出することができる。

40

【 0 1 4 2 】

上述したように、アフィンマージモードで、隣接ブロックのアフィンシードベクトルを用いて現在ブロックのアフィンシードベクトルを導出することができる。

【 0 1 4 3 】

現在ブロックのインター予測モードがアフィン動きベクトル予測モードである場合、動きベクトル予測候補リストに含まれる動きベクトル予測候補のアフィンシードベクトルを現在ブロックのアフィンシードベクトル予測値として決定することができる。前記アフィンシードベクトル予測値にアフィンシードベクトル差分値を加算することで、現在ブロックのアフィンシードベクトルを導出することができる。

50

【0144】

現在ブロックの隣接ブロックに基づいて、アフィンシードベクトル予測候補を導出することができる。具体的には、所定の走査順に応じて、現在ブロックの上方に位置する複数の隣接ブロックを探索し、また、最初に発見されたアフィン隣接ブロックから第1アフィンシードベクトル予測候補を導出することができる。なお、所定の走査順に応じて、現在ブロックの左側に位置する複数の隣接ブロックを探索し、また、最初に発見されたアフィン隣接ブロックから第2アフィンシードベクトル予測候補を導出することができる。

【0145】

アフィンシードベクトル差分値を決定するための情報に対して符号化を行い、ビットストリームを介して伝送することができる。前記情報は、アフィンシードベクトル差分値のサイズを表すサイズ情報及びアフィンシードベクトル差分値のシンボルを表すシンボル情報を含んでもよい。各制御点に関わるアフィンシードベクトル差分値を同じであるように設定することができる。あるいは、各制御点に対して、アフィンシードベクトル差分値を異なるように設定することができる。

10

【0146】

上述したように、アフィン隣接ブロックのアフィンシードベクトルからマージ候補又はアフィンシードベクトル予測候補のアフィンシードベクトルを導出することができ、また、導出されたマージ候補又はアフィンシードベクトル予測候補のアフィンシードベクトルを用いて、現在ブロックのアフィンシードベクトルを導出することができる。あるいは、所定の走査順に応じて複数のアフィン隣接ブロックを探索した後、最初に発見されたアフィン隣接ブロックのアフィンシードベクトルから、現在ブロックのアフィンシードベクトルを導出することもできる。

20

【0147】

後述において、アフィン隣接ブロックのアフィンシードベクトルから、現在ブロックのアフィンシードベクトル、マージ候補又はアフィンシードベクトル予測候補を導出する方法を詳しく説明する。後述する実施例において、現在ブロックのアフィンシードベクトルの導出は、マージ候補のアフィンシードベクトルの導出又はアフィンシードベクトル予測候補のアフィンシードベクトルの導出と理解されてもよい。

【0148】

図12は、アフィン隣接ブロックのアフィンシードベクトルに基づいて現在ブロックのアフィンシードベクトルを導出することを示す図である。

30

【0149】

アフィン隣接ブロックに、左上制御点に関わる第1アフィンシードベクトル $n v_0$ 及び右上制御点に関わる第2アフィンシードベクトル $n v_1$ が記憶された場合、前記第1アフィンシードベクトル及び第2アフィンシードベクトルに基づいて、アフィン隣接ブロックの左下制御点に関わる第3アフィンシードベクトル $n v_2$ を導出することができる。等式4は、第3アフィンシードベクトルの導出例を示す。

【数4】

【等式4】

40

$$n v_{2x} = \frac{(n v_{1x} - n v_{0x})}{(x_{n1} - x_{n0})} (x_{n2} - x_{n0}) - \frac{(n v_{1y} - n v_{0y})}{(x_{n1} - x_{n0})} (y_{n2} - y_{n0}) + n v_{0x}$$

$$n v_{2y} = \frac{(n v_{1y} - n v_{0y})}{(x_{n1} - x_{n0})} (x_{n2} - x_{n0}) - \frac{(n v_{1x} - n v_{0x})}{(x_{n1} - x_{n0})} (y_{n2} - y_{n0}) + n v_{0y}$$

50

【0150】

等式4において、 (nv_{0x}, nv_{0y}) は、第1アフィンシードベクトル nv_0 を表し、 (nv_{1x}, nv_{1y}) は、第2アフィンシードベクトル nv_1 を表し、また、 (nv_{2x}, nv_{2y}) は、第3アフィンシードベクトル nv_2 を表す。なお、 (x_{n0}, y_{n0}) は、第1制御点の位置を表し、 (x_{n1}, y_{n1}) は、第2制御点の位置を表し、また、 (x_{n2}, y_{n2}) は、第3制御点の位置を表す。

【0151】

続いて、第1アフィンシードベクトル、第2アフィンシードベクトル及び第3アフィンシードベクトルを用いて現在ブロックのアフィンシードベクトルを導出することができる。等式5は、現在ブロックの第1アフィンシードベクトル nv_0 の導出例を示し、また、等式6は、現在ブロックの第2アフィンシードベクトル nv_1 の導出例を示す。

10

【数5】

【等式5】

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

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

20

【数6】

【等式6】

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

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

30

【0152】

等式5及び等式6において、 (v_{0x}, v_{0y}) は、現在ブロックの第1アフィンシードベクトル sv_0 を表し、また、 (v_{1x}, v_{1y}) は、現在ブロックの第2アフィンシードベクトル sv_1 を表す。なお、 (x_0, y_0) は、第1制御点の位置を表し、また、 (x_1, y_1) は、第2制御点の位置を表す。例えば、第1制御点は、現在ブロックの左上隅を表し、また、第2制御点は、現在ブロックの右上隅を表す。

40

【0153】

前記例において、アフィン隣接ブロックに関わる3つのアフィンシードベクトルを用いて現在ブロックの複数のアフィンシードベクトルを導出することを説明した。もう1つの例として、アフィン隣接ブロックの複数のアフィンシードベクトルのうちの2つのみを用いて現在ブロックのアフィンシードベクトルを導出することもできる。

【0154】

あるいは、アフィン隣接ブロックに関わる左上隅の第1アフィンシードベクトル、右上

50

隅の第2アフィンシードベクトル又は左下隅の第3アフィンシードベクトルを用いることなく、右下隅に関わる第4アフィンシードベクトルを用いて、現在ブロックの複数のアフィンシードベクトルを導出することができる。

【0155】

特に、現在ブロックの上境界が符号化ツリーユニットの上境界と接触しており、且つ現在ブロックの上方に隣接するアフィン隣接ブロック（以下、上方アフィン隣接ブロックと呼ぶ）の上方制御点（例えば、左上隅又は右上隅）のアフィンシードベクトルを使用しようとする場合、これらをメモリに事前記憶する必要がある。これは、ラインバッファの数の増加という問題を引き起こす可能性がある。従って、現在ブロックの上境界が符号化ツリーユニットの上境界と接触した場合、上方アフィン隣接ブロックに対して、上方制御点のアフィンシードベクトルを用いることなく、下方制御点（例えば、左下隅又は右下隅）のアフィンシードベクトルを用いると設定することができる。例えば、上方アフィン隣接ブロックの左下隅に関わる第3アフィンシードベクトル及び右下隅に関わる第4アフィンシードベクトルを用いて、現在ブロックの複数のアフィンシードベクトルを導出することができる。この場合、下隅に関わるアフィンシードベクトルを、上隅に関わるアフィンシードベクトルを複製することで導出することができ、又は、複数の上隅に関わるアフィンシードベクトルから導出することができる。例えば、第1アフィンシードベクトル、第2アフィンシードベクトル又は第3アフィンシードベクトルを右下隅に関わる第4アフィンシードベクトルに変換する／置き換えることができる。

【0156】

等式7及び等式8は、隣接アフィンベクトルの左下制御点に関わる第3アフィンシードベクトル及び右下制御点に関わる第4アフィンシードベクトルを用いて現在ブロックの第1アフィンシードベクトル及び第2アフィンシードベクトルを導出する例を示す。

【数7】

【等式7】

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

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

【数8】

【等式8】

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

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

【0157】

等式7及び8において、(x_{n2}, y_{n2})は、アフィン隣接ブロックの左下制御点の座

10

20

30

40

50

標を表し、また、 (x_{n3}, y_{n3}) は、アフィン隣接ブロックの右下制御点の座標を表す。 (x_0, y_0) は、現在ブロックの左上制御点の座標を表し、また、 (x_1, y_1) は、現在ブロックの右上制御点の座標を表す。 (nv_{2x}, nv_{2y}) は、アフィン隣接ブロックの左下制御点のアフィンシードベクトル（即ち、第3アフィンシードベクトル）を表し、また、 (nv_{3x}, nv_{3y}) は、アフィン隣接ブロックの右下制御点のアフィンシードベクトル（即ち、第4アフィンシードベクトル）を表す。 (v_{0x}, v_{0y}) は、現在ブロックの左上制御点のアフィンシードベクトル（即ち、第1アフィンシードベクトル）を表し、また、 (v_{1x}, v_{1y}) は、現在ブロックの右上制御点のアフィンシードベクトル（即ち、第2アフィンシードベクトル）を表す。

【0158】

10

等式7及び等式8に含まれる除算をシフト操作に変更することもできる。シフト操作を、左下制御点と右下制御点との幅（即ち $(x_{n3} - x_{n2})$ ）から導出された値に基づいて実行することができる。

【0159】

上記例において、符号化／復号化されたアフィン隣接ブロックの複数のアフィンシードベクトルに基づいて、現在ブロックの複数のアフィンシードベクトルを導出することができる。このために、符号化／復号化されたアフィン隣接ブロックの複数のアフィンシードベクトルをメモリに記憶することができる。しかしながら、アフィン隣接ブロックに含まれる複数のサブブロックの複数の並進動きベクトル（即ち、複数のアフィンベクトル）に加えて、アフィン隣接ブロックの複数のアフィンシードベクトルを更にメモリに記憶したため、メモリの使用量の増加という問題を招く。このような問題を解消するために、アフィン隣接ブロックの制御点に隣接するサブブロックの動きベクトルを用いて現在ブロックのアフィンシードベクトルを導出することができる。これにより、アフィン隣接ブロックのアフィンシードベクトルに取って代わる。つまり、アフィン隣接ブロックの制御点に隣接するサブブロックの動きベクトルをアフィン隣接ブロックのアフィンシードベクトルと設定することができる。ここで、サブブロックは、エンコーダ及びデコーダで事前定義されたサイズ／形状を有するブロックであり、また、動きベクトルを記憶する基本サイズ／形状を有するブロックであってもよい。例えば、サブブロックは、サイズが 4×4 である正方形ブロックであってもよい。あるいは、サンプル位置を指定する動きベクトルをアフィン隣接ブロックのアフィンシードベクトルと設定することができる。

20

【0160】

30

図13は、サブブロックの動きベクトルをアフィン隣接ブロックのアフィンシードベクトルとする例を示す図である。

【0161】

制御点に隣接するサブブロックの動きベクトルを対応する制御点のアフィンシードベクトルとすることができる。例えば、図13に示す例において、アフィン隣接ブロックの左下隅に隣接するサブブロック（左下サブブロック）の動きベクトル (nv_{4x}, nv_{4y}) を左下制御点のアフィンシードベクトル (nv_{2x}, nv_{2y}) とすることができ、また、右下隅に隣接するサブブロック（右下サブブロック）の動きベクトル (nv_{5x}, nv_{5y}) を右下隅の制御点のアフィンシードベクトル (nv_{3x}, nv_{3y}) とすることができる。ここで、左下サブブロックは、隣接アフィンブロックの左下制御点 (x_{n2}, y_{n2}) に隣接するサンプル（例えば、 $(x_{n2}, y_{n2} - 1)$ 位置でのサンプル）を含むサブブロックであり、また、右下サブブロックは、隣接アフィンブロックの右下制御点 (x_{n3}, y_{n3}) に隣接するサンプル（例えば、 $(x_{n3} - 1, y_{n3} - 1)$ 位置でのサンプル）を含むブロックである。等式7及び等式8に基づいて現在ブロックのアフィンシードベクトルを導出する場合、アフィン隣接ブロックの第3アフィンシードベクトルの代わりに、左下サブブロックの動きベクトルを用いることができ、また、第4アフィンシードベクトルの代わりに、右下サブブロックの動きベクトルを用いることができる。

40

【0162】

以下、後述する実施例において、アフィン隣接ブロックのアフィンシードベクトルとし

50

て用いられるサブプロックをアフィンサブプロックと呼ぶ。

【0163】

本発明の一実施例によれば、特定位置でのサンプルに基づいて、アフィンサブプロックを決定することができる。例えば、特定位置でのサンプルを含むサブプロックをアフィンサブプロックとすることもできる。以下、特定位置でのサンプルをアフィン参照サンプルと呼ぶ。なお、左下制御点のアフィンサブプロックを決定するための参照サンプルを左下参照サンプルと呼び、また、右下制御点のアフィンサブプロックを決定するための参照サンプルを右下参照サンプルと呼ぶ。

【0164】

左下参照サンプル及び右下参照サンプルは、アフィン隣接プロックに含まれる複数のサンプルから選ばれてもよい。例えば、左下サブプロックの左上サンプル、左下サンプル、右上サンプル又は左下サンプルのうちの少なくとも1つを左下参照サンプルとし、また、右下サブプロックの左上サンプル、左下サンプル、右上サンプル又は左下サンプルのうちの少なくとも1つを右下参照サンプルとすることができます。従って、左下参照サンプルを含む左下サブプロック及び右下参照サンプルを含む右下サブプロックの動きベクトルを、それぞれ、左下制御点に関わるアフィンシードベクトル及び右下制御点に関わるアフィンシードベクトルとすることができます。

10

【0165】

もう1つの例として、左下参照サンプル又は右下参照サンプルのうちの少なくとも1つをアフィン隣接プロック外に位置するサンプルとすることができます。これに対して、図14から図16を参照しながら詳しく説明する。

20

【0166】

図14から図16は、参照サンプルの位置を示す図である。

【0167】

例えば、図14の(a)に示す例のように、左下制御点に対して、左下サブプロックの左上サンプルを参照サンプル(x_{n4}, y_{n4})とすることができます。従って、参照サンプル(x_{n4}, y_{n4})を含む左下サブプロックを、左下制御点に関わるアフィンサブプロックとすることができます。

【0168】

右下制御点に対して、右下サブプロックの右上サンプルの右側に位置するサンプルを参照サンプル(x_{n5}, y_{n5})とすることができます。従って、参照サンプル(x_{n5}, y_{n5})を含む右下サブプロックの右側に隣接するサブプロックを、右下制御点に関わるアフィンサブプロックとすることができます。

30

【0169】

あるいは、図14の(b)に示す例のように、左下制御点に対して、左下サブプロックの左上サンプルの左側に位置するサンプルを参照サンプル(x_{n4}, y_{n4})とすることができます。従って、参照サンプル(x_{n4}, y_{n4})を含む左下サブプロックの左側に隣接するサブプロックを左下制御点に関わるアフィンサブプロックとすることができます。

【0170】

右下制御点に対して、右下サブプロックの右上サンプルを参照サンプル(x_{n5}, y_{n5})とすることができます。従って、参照サンプル(x_{n5}, y_{n5})を含む右下サブプロックを、右下制御点に関わるアフィンサブプロックとすることができます。

40

【0171】

あるいは、図15の(a)に示す例のように、左下制御点に対して、左下サブプロックの左下サンプルを参照サンプル(x_{n4}, y_{n4})とすることができます。従って、参照サンプル(x_{n4}, y_{n4})を含む左下サブプロックを、左下制御点に関わるアフィンサブプロックとすることができます。

【0172】

右下制御点に対して、右下サブプロックの右下サンプルの右側に位置するサンプルを参照サンプル(x_{n5}, y_{n5})とすることができます。従って、参照サンプル(x_{n5}, y_{n5})

50

5) を含む右下サブブロックの右側に隣接するサブブロックを、右下制御点に関わるアフィンサブブロックとすることができます。

【 0 1 7 3 】

あるいは、図 15 の (b) に示す例のように、左下制御点に対して、左下サブブロックの左下サンプルの左側に位置するサンプルを参照サンプル (x_{n4}, y_{n4}) とすることができる。従って、参照サンプル (x_{n4}, y_{n4}) を含む左下サブブロックの左側に隣接するサブブロックを、左下制御点に関わるアフィンサブブロックとすることができます。

【 0 1 7 4 】

右下制御点に対して、右下サブブロックの右下サンプルを参照サンプル (x_{n5}, y_{n5}) とすることができる。従って、参照サンプル (x_{n5}, y_{n5}) を含む右下サブブロックを、右下制御点に関わるアフィンサブブロックとすることができます。 10

【 0 1 7 5 】

あるいは、図 16 の (a) に示す例のように、左下制御点に対して、左下サブブロックの左上サンプルと左下サンプルとの間に位置するサンプル (例えは、左側中間サンプル) を参照サンプル (x_{n4}, y_{n4}) とすることができる。従って、参照サンプル (x_{n4}, y_{n4}) を含む左下サブブロックを、左下制御点に関わるアフィンサブブロックとすることができます。

【 0 1 7 6 】

右下制御点に対して、右下サブブロックの右上サンプルと右下サンプルとの間のサンプル (例えは、右側中間サンプル) の右側に位置するサンプルを参照サンプル (x_{n5}, y_{n5}) とすることができる。従って、参照サンプル (x_{n5}, y_{n5}) を含む右下サブブロックの右側に隣接するサブブロックを、右下制御点に関わるアフィンサブブロックとすることができます。 20

【 0 1 7 7 】

あるいは、図 16 の (b) に示す例のように、左下制御点に対して、左下サブブロックの左上サンプルと左下サンプルとの間のサンプルの左側に位置するサンプルを参照サンプル (x_{n4}, y_{n4}) とすることができる。従って、参照サンプル (x_{n4}, y_{n4}) を含む左下サブブロックの左側に隣接するサブブロックを、左下制御点に関わるアフィンサブブロックとすることができます。

【 0 1 7 8 】

右下制御点に対して、右下サブブロックの右上サンプルと右下サンプルとの間に位置するサンプルを参照サンプル (x_{n5}, y_{n5}) とすることができる。従って、参照サンプル (x_{n5}, y_{n5}) を含む右下サブブロックを、右下制御点に関わるアフィンサブブロックとすることができます。 30

【 0 1 7 9 】

等式 7 及び等式 8 に基づいて現在ブロックの複数のアフィンシードベクトルを導出する場合、アフィン隣接ブロックの第 3 アフィンシードベクトルの代わりに、左下制御点に関わるアフィンサブブロックの動きベクトルを用いることができ、また、第 4 アフィンシードベクトルの代わりに、右下制御点に関わるアフィンサブブロックの動きベクトルを用いることができる。なお、左下制御点の位置の代わりに、左下参照サンプルの位置を用いることができ、また、右下制御点の位置の代わりに、右下参照サンプルの位置を用いることができる。 40

【 0 1 8 0 】

図 14 から図 16 における記載と異なっており、参照サンプルに隣接するサンプルを含むサブブロックを、アフィンサブブロックとすることもできる。具体的には、アフィン隣接サブブロック外に位置するサンプルを参照サンプルとすることができます、また、アフィン隣接ブロックに含まれるサブブロックを、アフィンサブブロックとすることができます。例えば、図 14 の (a) に示す例において、右下サブブロックの右上サンプルの右側に位置するサンプルを参照サンプル (x_{n5}, y_{n5}) とすることができます、また、右下サブブロックを、右下隅に関わるアフィンサブブロックとすることができます。あるいは、図 14 の (50

b) に示す例において、左下サブブロックの左上サンプルの左側に位置するサンプルを参照サンプル (x_{n4}, y_{n4}) とすることができる、また、左下サブブロックを、左下隅に関わるアフィンサブブロックとすることができる。

【0181】

図15及び図16に記載の実施例を同様に適用することができる。つまり、図15の(a)又は図16の(a)に示す例において、右下サブブロックの右下サンプル又は右側中間サンプルの右側に位置するサンプルを参照サンプル (x_{n5}, y_{n5}) とすることができる、また、右下サブブロックを、右下隅に関わるアフィンサブブロックとすることができる。あるいは、図15の(b)又は図16の(b)に示す例において、左下サブブロックの左下サンプル又は左側中間サンプルの左側に位置するサンプルを参照サンプル (x_{n4}, y_{n4}) とでき、また、左下サブブロックを、左下隅に関わるアフィンサブブロックとすることができる。

【0182】

上述した例において、アフィンサブブロックの動きベクトルを用いることでアフィン隣接ブロックのアフィンシードベクトルを導出することができる。このために、符号化/復号化されたブロックについて言えば、サブブロックを単位として動きベクトルを記憶することができる。

【0183】

もう1つの例として、アフィン隣接ブロックに最小数のアフィンシードベクトルを記憶した後、記憶された複数のアフィンシードベクトルを用いてアフィンサブブロックの動きベクトルを導出することができる。

【0184】

等式9及び等式10は、アフィン隣接ブロックのアフィンシードベクトルを用いてアフィンサブブロックの動きベクトルを導出する例を示す。

【数9】

【等式9】

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

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

【数10】

【等式10】

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

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

10

20

40

50

【0185】

等式9及び等式10において、 $(n v_{4x}, n v_{4y})$ は、左下制御点に関わるアフィンサブブロックの動きベクトルを表し、また、 $(n v_{5x}, n v_{5y})$ は、右下制御点に関わるアフィンサブブロックの動きベクトルを表す。アフィンサブブロックの動きベクトルと制御点のアフィンシードベクトルが同じであるように設定されたため、 $(n v_{4x}, n v_{4y})$ の代わりに、左下制御点に関わるアフィンシードベクトル $(n v_{2x}, n v_{2y})$ を用いることができ、又は、 $(n v_{5x}, n v_{5y})$ の代わりに、右下制御点に関わるアフィンシードベクトル $(n v_{3x}, n v_{3y})$ を用いることができる。

【0186】

(x_{n4}, y_{n4}) は、左下サブブロックの参照サンプルの位置を表す。あるいは、該位置の代わりに、左下サブブロックの中心位置又は左下制御点の位置を用いることもできる。 (x_{n5}, y_{n5}) は、右下サブブロックの参照サンプルの位置を表す。あるいは、該位置の代わりに、右下サブブロックの中心位置又は右下制御点の位置を用いることもできる。

【0187】

現在ブロックが符号化ツリーユニットの境界と接触しない場合に等式9及び等式10を適用することができる。現在ブロックが符号化ツリーユニットの上境界と接触した場合、等式9及び等式10の適用の代わりに、左下参照サンプルに基づいて決定されたアフィンサブブロックの並進動きベクトルを第3アフィンシードベクトルとし、また、右下参照サンプルに基づいて決定されたアフィンサブブロックの並進動きベクトルを第4アフィンシードベクトルとすることができる。

【0188】

等式7及び等式8において、 $(x_{n3} - x_{n2})$ は、左下制御点と右下制御点との幅を表す。上述したように、 x_{n3} の代わりに、右下参照サンプルの位置 x_{n5} を用いることができ、また、 x_{n2} の代わりに、左下参照サンプルの位置 x_{n4} を用いることができる。以下、 $(x_{n3} - x_{n2})$ 又は、前記等式の代わりに参照サンプルの位置を用いることで得られた値（例えば、 $(x_{n5} - x_{n4})$ ）を変数 W_{seed} と定義し、前記変数をサブシードベクトル幅と呼ぶ。

【0189】

参照サンプルの位置によれば、サブシードベクトル幅が2のべき級数（例えば、 2^n ）ではないことを発生する可能性がある。例えば、左下サブブロックの左下サンプルを左下参照サンプルとして、右下サブブロックの右下サンプルを右下参照サンプルとする場合、サブシードベクトル幅は、2の倍数ではない。上述したように、サブシードベクトル幅が2のべき級数ではない場合、サブシードベクトル幅を2のべき級数に変換することができる。前記変換は、サブシードベクトル幅にオフセットだけ加算／減算すること、又は、参照サンプルの位置の代わりに前記参照サンプルに隣接するサンプルの位置を用いることを含んでもよい。例えば、左下参照サンプルと右下参照サンプルとの幅に1を加算することで、変換されたサブシードベクトル幅を導出することができる。あるいは、右下参照サンプルの右側に隣接する隣接参照サンプルと左下参照サンプルとの幅を、変換されたサブシードベクトル幅とすることができます。続いて、変換されたサブシードベクトル幅を等式7及び等式8に代入することで、現在ブロックのアフィンシードベクトルを導出することができる。

【0190】

等式7及び等式8に含まれる除算をシフト操作に変更することもできる。シフト操作を、変換されたサブシードベクトル幅（即ち、2のべき級数として表される値）から導出された値に基づいて実行することができる。

【0191】

アフィンサブブロックを決定するための参照サンプルがアフィン隣接ブロックに属しない場合、アフィン隣接ブロックに含まれる複数のサンプルのうちの、参照サンプルに隣接するサンプルに基づいて、アフィン隣接ブロックのアフィンシードベクトルを導出することができる。具体的には、アフィン隣接ブロックにおける参照サンプルに隣接するサンプ

10

20

30

40

50

ル（以下、隣接参照サンプルと呼ばれる）を含むサブブロックの並進動きベクトルを、アフィン隣接ブロックのアフィンシードベクトルとすることができる。上述したように、隣接参照サンプルを用いてアフィンシードベクトルを導出する方法を、修正されたアフィンマージベクトル導出方法と定義することができる。

【0192】

図17は、修正したアフィンマージベクトル導出方法の適用例を示す図である。

【0193】

アフィン隣接ブロックEの右下参照サンプル(x_{n5}, y_{n5})がアフィン隣接ブロックに属しない場合、アフィン隣接ブロックに含まれるサンプルのうちの、右下参照サンプルの左側に隣接するサンプル(x_{n5-1}, y_{n5})に基づいて、アフィンシードベクトルを導出することができる。具体的には、隣接参照サンプル(x_{n5-1}, y_{n5})を含むサブブロックの並進動きベクトルを、右下制御点のアフィンシードベクトルとすることができる。

10

【0194】

図17に示す例において、右下サブブロックの右上サンプルの右側に隣接するサンプルを、右下参照サンプルとして示す。右下サブブロックの右下サンプルの右側に隣接するサンプル又は右下サブブロックの右側中間サンプルの右側に隣接するサンプルを右下参照サンプルとした場合、隣接参照サンプルの左側に隣接するサンプルに基づいて、アフィンシードベクトルを導出することができる。

20

【0195】

なお、左下参照サンプルがアフィン隣接ブロックに属しない場合、記述された実施例に応じて、左下参照サンプルの右側に隣接するサンプルに基づいて、アフィンシードベクトルを導出することもできる。

【0196】

異なる方式で参照サンプルの位置及びアフィンシードベクトル導出用サブブロックを設定することで、サブシードベクトル幅を2のべき級数とすることができる。

【0197】

現在ブロックの周辺の、アフィンインターモードで符号化されていない隣接ブロックを用いることで、現在ブロックのマージ候補、アフィンシードベクトル予測候補又はアフィンシードベクトルを導出することができる。具体的には、アフィンインターモードで符号化されていないブロックを組み合わせ、また、その組み合わせをマージ候補又はアフィンシードベクトル予測候補とすることができる。例えば、現在ブロックの左上隅に隣接する隣接ブロックのうちのいずれか1つの隣接ブロックの動きベクトル、現在ブロックの右上隅に隣接する隣接ブロックのうちのいずれか1つの隣接ブロックの動きベクトル、及び現在ブロックの左下隅に隣接する隣接ブロックのうちのいずれか1つの隣接ブロックの動きベクトルのうちの少なくとも1つの組み合わせを、マージ候補又はアフィンシードベクトル予測候補とすることができる。この場合、左上隅に隣接する隣接ブロックの動きベクトル、右上隅に隣接する隣接ブロックの動きベクトル及び左下隅に隣接する隣接ブロックの動きベクトルをそれぞれ、左上制御点の第1アフィンシードベクトル、右上制御点の第2アフィンシードベクトル及び左下制御点の第3アフィンシードベクトルとすることができる。

30

【0198】

あるいは、上記修正したアフィンマージベクトル導出方法において、アフィンインターモードで符号化されていない隣接ブロックを用いて、現在ブロックのマージ候補、アフィンシードベクトル予測候補又はアフィンシードベクトルを導出することができる。下記において、アフィンインターモードで符号化されていない隣接ブロックは、非アフィン隣接ブロックと呼ばれる。

40

【0199】

図18は、非アフィン隣接ブロックに基づいて現在ブロックのアフィンシードベクトルを導出する例を示す図である。

50

【0200】

図18に示す例において、現在ブロックに隣接する全ての隣接ブロックがいずれも非アフィン隣接ブロックであると仮定する。

【0201】

現在ブロックに隣接する隣接ブロックのうちの非アフィン隣接ブロックAから現在ブロックのアフィンシードベクトルを導出しようとする場合、Aの左下参照サンプル及び右下参照サンプルを設定することができる。例えば、ブロックAの左下サンプルの左側に隣接するサンプルを左下参照サンプルとすることができます、また、ブロックAの右下サンプルを右下参照サンプルとすることができます。左下参照サンプルがブロックAの外にあるため、左下参照サンプルの右側に隣接するサンプルを含むサブブロックの動きベクトルをブロックAの第3アフィンシードベクトルとすることができます。なお、右下参照サンプルを含むサブブロックの動きベクトルをブロックAの第4アフィンシードベクトルとすることができます。続いて、等式9及び10に基づいて、ブロックAから、現在ブロックの第1アフィンシードベクトル及び第2アフィンシードベクトルを導出することができる。

10

【0202】

サブブロックを単位として非アフィン隣接ブロックに対する動き補償予測を実行する場合のみ、非アフィン隣接ブロックからアフィンシードベクトルを導出する方法を用いることができる。ここで、サブブロックをとして動き補償予測を実行するための予測技術は、STMVP、ATMVP、双方向オプティカルフロー(BIO)、オーバラップブロック動き補償(OBMC)及びデコーダ側動きベクトル修正(DMVR)のうちの少なくとも1つを含んでもよい。

20

【0203】

上記実施例において、現在ブロックの上境界が符号化ツリーユニットの境界と接触した場合、現在ブロックの上方に位置するアフィン隣接ブロックの左下制御点の第3アフィンシードベクトル及び右下制御点の第4アフィンシードベクトルを用いて、現在ブロックのマージ候補、アフィンシードベクトル予測候補又はアフィンシードベクトルを導出することを説明した。

【0204】

もう1つの例として、現在ブロックの上境界が符号化ツリーユニットの境界と接触しており、且つ現在ブロックの上方に位置する隣接ブロックが現在ブロックの符号化ツリーユニットと異なる符号化ツリーユニットに属すると、前記隣接ブロックを用いることなく、現在ブロックが属する符号化ツリーユニットに含まれるこれらのブロックのうちの、前記隣接ブロックに最も近接する隣接ブロックを用いて、現在ブロックのマージ候補、アフィンシードベクトル予測候補又はアフィンシードベクトルを導出する。

30

【0205】

図19に示す例において、現在ブロックが符号化ツリーユニットの上境界と接触しており、且つ現在ブロックの上方に位置するブロックB、C及びEが現在ブロックの符号化ツリーユニットと異なる符号化ツリーユニットに属することを示す。従って、ブロックEの代わりに、現在ブロックが属する符号化ツリーユニットに含まれるこれらのブロックのうちの、ブロックEに隣接するブロックFを用いて、現在ブロックのアフィンシードベクトルを導出する。

40

【0206】

現在ブロックの動き補償予測のために、複数のブロックのアフィンシードベクトルを用いることができる。例えば、マージ候補リストから複数のマージ候補を選択し、また、選択されたマージ候補のアフィンシードベクトルに基づいて、現在ブロックのアフィンシードベクトル又はサブブロックベクトルを導出することができる。複数のブロックのアフィンシードベクトルを用いて現在ブロックに対して符号化/復号化を行うことは、マルチアフィンマージ符号化方法と呼ばれてもよい。

【0207】

現在ブロックに対してマルチアフィンマージ符号化方法を適用したかどうかを示す情報

50

に対して符号化を行い、ビットストリームを介して伝送することができる。あるいは、現在ブロックに隣接する隣接ブロックのうちのアフィン隣接ブロックの数、マージ候補リストに含まれるマージ候補の数、及び現在ブロックのアフィン動きモデルのうちの少なくとも1つに基づいて、現在ブロックに対してマルチアフィンマージ符号化方法を適用したかどうかを決定することができる。

【0208】

図20及び図21は、複数のマージ候補を用いた動き補償予測方法を示すフローチャートである。

【0209】

図20は、複数のマージ候補のアフィンシードベクトルを用いることで現在ブロックのアフィンシードベクトルを導出する例を示す図である。図21は、複数のマージ候補のアフィンシードベクトルを用いることで各サブブロックの動きベクトルを導出する例を示す図である。

10

【0210】

2つのマージ候補のアフィンシードベクトルの和、差、平均値、又は加重加算に基づいて、現在ブロックのアフィンシードベクトルを生成することができる。

【0211】

下記等式11及び12は、マージ候補のアフィンシードベクトルを加算することで現在ブロックのアフィンシードベクトルを導出する例を示す。

【数11】

20

【等式11】

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

【数12】

【等式12】

30

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

【0212】

等式11及び12において、 sv_4 は、現在ブロックの第1アフィンシードベクトルを表し、 sv_0 は、第1マージ候補の第1アフィンシードベクトルを表し、また、 sv_2 は、第2マージ候補の第1アフィンシードベクトルを表す。なお、 sv_5 は、現在ブロックの第2アフィンシードベクトルを表し、 sv_1 は、第1マージ候補の第2アフィンシードベクトルを表し、 sv_3 は、第2マージ候補の第2アフィンシードベクトルを表す。

40

【0213】

なお、下記等式13及び14は、マージ候補のアフィンシードベクトルの加重加算により、現在ブロックのアフィンシードベクトルを導出する例を示す。

【数13】

50

【等式13】

$$(sv_{4x}, sv_{4y}) = (sv_{0x}, sv_{0y}) + ((sv_{2x}, sv_{2y}) \gg 1)$$

【数14】

【等式14】

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

【0214】

もう1つの例として、第1マージ候補のアフィンシードベクトル及び第2マージ候補のアフィンシードベクトルに基づいて、現在ブロックにおける各サブブロックに対して、第1サブブロック動きベクトル及び第2サブブロック動きベクトルを生成することができる。続いて、第1サブブロック動きベクトル及び第2サブブロック動きベクトルの和、差、平均値又は加重加算に基づいて、最終的サブブロック動きベクトルを生成することができる。

【0215】

下記等式15は、第1サブブロック動きベクトルと第2サブブロック動きベクトルを加算することで最終的サブブロック動きベクトルを得る例を示す。

【数15】

【等式15】

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

30

【0216】

等式15において、 V_0 は、第1サブブロック動きベクトルを表し、 V_1 は、第2サブブロック動きベクトルを表し、また、 V_2 は、最終的サブブロック動きベクトルを表す。

【0217】

なお、下記等式16は、第1サブブロック動きベクトル及び第2サブブロック動きベクトルの加重加算により、最終的サブブロック動きベクトルを導出する例を示す。

【数16】

【等式16】

$$(V_{2x}, V_{2y}) = (V_{0x}, V_{0y}) + ((V_{1x}, V_{1y}) \gg 1)$$

40

【0218】

50

イントラ予測は、現在ブロックの周辺の符号化 / 復号化された再構成サンプルを用いて現在ブロックを予測することである。この場合、現在ブロックのイントラ予測は、インループフィルタを適用する前の再構成サンプルを用いることができる。

【0219】

イントラ予測技術は、マトリックス (Matrix) に基づくイントラ予測及び周辺の再構成サンプルとの方向性を考慮した一般的なイントラ予測を含む。ビットストリームを介して、信号で、現在ブロックのイントラ予測技術を示す情報を送信することができる。前記情報は、1ビットフラグであってもよい。あるいは、現在ブロックの位置、サイズ、形状又は隣接ブロックのイントラ予測技術のうちの少なくとも1つに基づいて、現在ブロックのイントラ予測技術を決定することができる。例えば、現在ブロックが画像の境界に跨って存在する場合、現在ブロックを、マトリックスに基づくイントラ予測を適用しないように設定することができる。

【0220】

マトリックスに基づくイントラ予測は、エンコーダ及びデコーダに記憶されたマトリックスと現在ブロックの周辺の再構成サンプルとのマトリックス乗算に基づいて現在ブロックの予測ブロックを得る方法である。ビットストリームを介して、信号で、記憶された複数のマトリックスのうちのいずれか1つを示すための情報を送信することができる。デコーダは、前記情報及び現在ブロックのサイズに基づいて、現在ブロックのイントラ予測に用いられるマトリックスを決定することができる。

【0221】

一般的なイントラ予測は、非角度イントラ予測モード又は角度イントラ予測モードに基づいて現在ブロックの予測ブロックを得る方法である。以下、図面を参照しながら、一般的なイントラ予測に基づいてイントラ予測を実行するプロセスを詳しく説明する。

【0222】

図22は、本発明の実施例によるイントラ予測方法を示すフローチャートである。

【0223】

現在ブロックの参照サンプルラインを決定することができる (S2201)。参照サンプルラインは、現在ブロックの上方及び / 又は左側からずれたK番目のラインに含まれる参照サンプルの集合である。現在ブロックの周辺の符号化 / 復号化された再構成サンプルから、参照サンプルを導出することができる。

【0224】

ビットストリームを介して、信号で、複数の参照サンプルラインのうちの現在ブロックの参照サンプルラインを標識するインデックス情報を送信することができる。複数の参照サンプルラインは、現在ブロックの上方及び / 又は左側に位置する第1ライン、第2ライン、第3ライン又は第4ラインのうちの少なくとも1つに含まれてもよい。表1は、各参照サンプルラインに割り当てられたインデックスを示す。表1において、第1ライン、第2ライン及び第4ラインが参照サンプルライン候補として用いられる仮定する。

【表1】

インデックス	参照サンプルライン
0	第1参照サンプルライン
1	第2参照サンプルライン
2	第4参照サンプルライン

【0225】

現在ブロックの位置、サイズ、形状又は隣接ブロックの予測符号化モードのうちの少な

10

20

30

40

50

くとも 1 つに基づいて、現在ブロックの参照サンプルラインを決定することもできる。例えば、現在ブロックが画像、タイル、スライス又は符号化ツリーユニットの境界と接した場合、第 1 参照サンプルラインを、現在ブロックの参照サンプルラインとして決定することができる。

【 0 2 2 6 】

参照サンプルラインは、現在ブロックの上方に位置する上方参照サンプル及び現在ブロックの左側に位置する左側参照サンプルを含んでもよい。現在ブロックの周辺の再構成サンプルから、上方参照サンプル及び左側参照サンプルを導出することができる。前記再構成サンプルは、インループフィルタを適用する前の状態であってもよい。

【 0 2 2 7 】

図 2 3 は、各参照サンプルラインに含まれる参照サンプルを示す図である。

【 0 2 2 8 】

現在ブロックのイントラ予測モードによれば、参照サンプルラインに属する参照サンプルのうちの少なくとも 1 つを利用して予測サンプルを取得することができる。

【 0 2 2 9 】

次に、現在ブロックのイントラ予測モードを決定することができる (S 2 2 0 2)。現在ブロックのイントラ予測モードについて言えば、非角度イントラ予測モード又は角度イントラ予測モードのうちの少なくとも 1 つを現在ブロックのイントラ予測モードとして決定することができる。非角度イントラ予測モードは、平面 (P l a n e r) 及び D C を含み、また、角度イントラ予測モードは、左下対角線方向から右上対角線への 3 3 個又は 6 5 個のモードを含む。

【 0 2 3 0 】

図 2 4 は、イントラ予測モードを示す図である。

【 0 2 3 1 】

図 2 4 (a) は、3 5 個のイントラ予測モードを示し、また、図 2 4 (b) は、6 7 個のイントラ予測モードを示す。

【 0 2 3 2 】

図 2 4 に示した数より多いか又は少ない数のイントラ予測モードを定義することもできる。

【 0 2 3 3 】

現在ブロックに隣接する隣接ブロックのイントラ予測モードに基づいて、最確モード (M P M : M o s t P r o b a b l e M o d e) を設定することができる。ここで、隣接ブロックは、現在ブロックの左側に隣接する左隣接ブロック及び現在ブロックの上方に隣接する上隣接ブロックを含んでもよい。現在ブロックの左上サンプルの座標が (0 , 0) である場合、左隣接ブロックは、(- 1 , 0)、(- 1 , H - 1) 又は (- 1 , (H - 1) / 2) 位置でのサンプルを含んでもよい。ここで、H は、現在ブロックの高さを表す。上隣接ブロックは、(0 , - 1)、(W - 1 , - 1) 又は ((W - 1) / 2 , - 1) 位置でのサンプルを含んでもよい。ここで、W は、現在ブロックの幅を表す。

【 0 2 3 4 】

一般的なイントラ予測により隣接ブロックに対して符号化を行う場合、隣接ブロックのイントラ予測モードに基づいて M P M を導出することができる。具体的には、左隣接ブロックのイントラ予測モードを変数 c a n d I n t r a P r e d M o d e A とし、上方隣接ブロックのイントラ予測モードを変数 c a n d I n t r a P r e d M o d e B とすることができる。

【 0 2 3 5 】

この場合、隣接ブロックが利用不可能である時 (例えば、隣接ブロックが符号化 / 復号化されていないか又は隣接ブロックの位置が画像の境界からはずれた時) 、隣接ブロックがマトリックスに基づくイントラ予測により符号化を行う場合、隣接ブロックがインター予測により符号化を行う場合又は隣接ブロックが現在ブロックと異なる符号化ツリーユニットに含まれる場合、隣接ブロックのイントラ予測モードに基づいて導出された変数 c a n

10

20

30

40

50

`candIntraPredModeX` (ここで、Xは、A又はBである)を、デフォルトモードとすることができます。ここで、デフォルトモードは、平面モード、DCモード、垂直方向モード又は水平方向モードのうちの少なくとも1つを含んでもよい。

【0236】

あるいは、マトリックスに基づくイントラ予測により隣接ブロックを符号化する場合、マトリックスのいずれか1つを指定するためのインデックス値に対応するイントラ予測モードを`candIntraPredModeX`とすることができます。このために、マトリックスを指定するためのインデックス値とイントラ予測モードとのマッピング関係を示すルックアップテーブルをエンコーダ及びデコーダに予め記憶することができる。

【0237】

変数`candIntraPredModeA`及び変数`candIntraPredModeB`に基づいてMPMを導出することができる。エンコーダ及びデコーダにおいて、MPMリストに含まれるMPMの数を事前定義することができる。例えば、MPMの数は、3つ、4つ、5つ又は6つであってもよい。あるいは、ビットストリームを介して、信号で、MPMの数を表す情報を送信することができる。あるいは、隣接ブロックの予測符号化モード、現在ブロックのサイズ又は形状のうちの少なくとも1つに基づいて、MPMの数を決定することができる。

【0238】

後述する実施例において、MPMの数が3つであると仮定し、該3つのMPMは、`MPM[0]`、`MPM[1]`及び`MPM[2]`と呼ばれる。MPMの数が3つより多い場合、MPMは、後述する実施例で説明される3つのMPMを含んでもよい。

20

【0239】

`candIntraPredA`が`candIntraPredB`と同じであり、且つ`candIntraPredA`が平面モード又はDCモードである場合、`MPM[0]`及び`MPM[1]`をそれぞれ、平面モード及びDCモードとすることができます。`MPM[2]`を垂直イントラ予測モード、水平イントラ予測モード又は対角線イントラ予測モードとすることができる。対角線イントラ予測モードは、左下対角線イントラ予測モード、左上イントラ予測モード又は右上イントラ予測モードであってもよい。

【0240】

`candIntraPredA`が`candIntraPredB`と同じであり、且つ`candIntraPredA`がイントラ予測モードである場合、`MPM[0]`を`candIntraPredA`と同じであるように設定することができる。`MPM[1]`及び`MPM[2]`を`candIntraPredA`と類似したイントラ予測モードと設定することができる。`candIntraPredA`と類似したイントラ予測モードは、`candIntraPredA`とのインデックス差分値が ± 1 又は ± 2 であるイントラ予測モードであってもよい。モジュロ演算(%)及びオフセットを用いて、`candIntraPredA`と類似したイントラ予測モードを導出することができる。

30

【0241】

`candIntraPredA`が`candIntraPredB`と異なる場合、`MPM[0]`と`candIntraPredA`と同じであるように設定し、`MPM[1]`と`candIntraPredB`と同じであるように設定することができる。この場合、`candIntraPredA`及び`candIntraPredB`がいずれも非角度イントラ予測モードである場合、`MPM[2]`を垂直イントラ予測モード、水平イントラ予測モード又は対角線イントラ予測モードとすることができます。あるいは、`candIntraPredA`及び`candIntraPredB`のうちの少なくとも1つが角度イントラ予測モードである場合、`MPM[2]`を平面、DC、`candIntraPredA`又は`candIntraPredB`のうちの大きい方の値に対してオフセットを加算するか又は減算することで導出されたイントラ予測モードとすることができます。ここで、オフセットは、1又は2であってもよい。

40

【0242】

50

複数の MPM を含む MPM リストを生成することができ、また、ビットストリームを介して、信号で、現在ブロックのイントラ予測モードと同じである MPM が MPM リストに含まれるかどうかを表す情報を送信することができる。前記情報は、1 ビットフラグであり、MPM フラグと呼ばれてもよい。MPM フラグは、現在ブロックと同じである MPM が MPM リストに含まれることを示す場合、ビットストリームを介して、信号で、MPM の 1 つを標識するインデックス情報を送信することができる。前記インデックス情報により指定された MPM を現在ブロックのイントラ予測モードとすることができます。MPM フラグは、現在ブロックと同じである MPM が MPM リストに含まれないことを示す場合、ビットストリームを介して、信号で、MPM 以外の残差イントラ予測モードのうちのいずれか 1 つを指示する残差モード情報を送信することができる。残差モード情報は、MPM 以外の残差イントラ予測モードにインデックスを再び割り当てる場合に現在ブロックのイントラ予測モードに対応するインデックス値を表す。デコーダは、昇順に応じて MPM を配列し、また、残差モード情報と MPM を比較することで、現在ブロックのイントラ予測モードを決定することができる。例えば、残差モード情報が MPM と同じであるか又はより小さい場合、残差モード情報に 1 を加算することで現在ブロックのイントラ予測モードを導出することができる。

【 0 2 4 3 】

デフォルトモードを MPM とする操作の代わりに、ビットストリームを介して、信号で、現在ブロックのイントラ予測モードがデフォルトモードであるかどうかを表す情報を送信することができる。前記情報は、1 ビットフラグであり、また、前記フラグは、デフォルトモードフラグと呼ばれてもよい。MPM フラグは、現在ブロックと同じである MPM が MPM リストに含まれることを示す場合のみ、信号で、デフォルトモードフラグを送信することができる。上述したように、デフォルトモードは、平面、DC、垂直方向モード又は水平方向モードのうちの少なくとも 1 つを含んでもよい。例えば、平面をデフォルトモードとした場合、デフォルトモードフラグは、現在ブロックのイントラ予測モードが平面であるかどうかを示すことができる。デフォルトモードフラグは、現在ブロックのイントラ予測モードがデフォルトモードではないことを示す場合、インデックス情報により示される MPM の 1 つを現在ブロックのイントラ予測モードとすることができます。

【 0 2 4 4 】

複数のイントラ予測モードをデフォルトモードとした場合、更に、信号で、デフォルトモードのうちのいずれか 1 つを示すインデックス情報を送信することができる。現在ブロックのイントラ予測モードを前記インデックス情報により示されるデフォルトモードとすることができる。

【 0 2 4 5 】

現在ブロックの参照サンプルラインのインデックスが 0 ではない場合、デフォルトモードを使用しないと設定する。従って、参照サンプルラインのインデックスが 0 ではない場合、信号でデフォルトモードフラグを送信することなく、また、前記デフォルトモードフラグの値を事前定義された値（即ち、偽である）と設定することができる。

【 0 2 4 6 】

現在ブロックのイントラ予測モードが決定された場合、決定されたイントラ予測モードに基づいて、現在ブロックに関わる予測サンプルを得ることができる（S 2 2 0 3）。

【 0 2 4 7 】

DC モードを選択した場合、参照サンプルの平均値に基づいて、現在ブロックに関わる予測サンプルを生成することができる。具体的には、参照サンプルの平均値に基づいて、予測ブロックの全てのサンプルの値を生成することができる。現在ブロックの上方に位置する上方参照サンプル及び現在ブロックの左側に位置する左側参照サンプルのうちの少なくとも 1 つを用いて平均値を導出することができる。

【 0 2 4 8 】

平均値を導出するための参照サンプルの数又は範囲は、現在ブロックの形状によって異なる可能性がある。例えば、現在ブロックは、幅が高さより大きい非正方形ブロックであ

る場合、上方参照サンプルのみを用いて平均値を算出することができる。一方で、現在ブロックは、幅が高さより小さい非正方形ブロックである場合、左側参照サンプルのみを用いて平均値を算出することができる。つまり、現在ブロックの幅と高さが異なる場合、長い長さに隣接する参照サンプルのみを用いて平均値を算出することができる。あるいは、現在ブロックの幅と高さとの比に基づいて、上方参照サンプルのみを用いて平均値を算出するか又は左側参照サンプルのみを用いて平均値を算出するかを決定することができる。

【0249】

平面モードを選択した場合、水平方向予測サンプル及び垂直方向予測サンプルを用いて予測サンプルを得ることができる。ここで、予測サンプルと同じである水平線に位置する左側参照サンプル及び右側参照サンプルに基づいて水平方向予測サンプルを得て、また、予測サンプルと同じである垂直線に位置する上方参照サンプル及び下方参照サンプルに基づいて垂直方向予測サンプルを得る。ここで、現在ブロックの右上隅に隣接する参照サンプルを複製することで右側参照サンプルを生成することができ、現在ブロックの左下隅に隣接する参照サンプルを複製することで下方参照サンプルを生成することができる。左側参照サンプル及び右側参照サンプルの加重加算に基づいて、水平方向予測サンプルを得ることができ、また、上方参照サンプル及び下方参照サンプルの加重加算に基づいて、垂直方向予測サンプルを得ることができる。この場合、予測サンプルの位置に基づいて、各参照サンプルに付与された加重値を決定することができる。水平方向予測サンプル及び垂直方向予測サンプルの平均演算又は加重加算に基づいて、予測サンプルを得ることができる。加重加算を実行する場合、予測サンプルの位置に基づいて、水平方向予測サンプル及び垂直方向予測サンプルに付与された加重値を決定することができる。

【0250】

角度予測モードを選択した場合、選択された角度予測モードの予測方向（又は予測角度）を表すパラメータを決定することができる。下記表2に、各イントラ予測モードのイントラ予測パラメータ `intraPredAng` を示す。

【表2】

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

【0251】

表2は、35個のイントラ予測モードを定義した場合に2から34のうちのいずれか1つのインデックスを有する各イントラ予測モードのイントラ方向パラメータを示す。33種を越える角度イントラ予測モードを定義した場合、表2は、各角度イントラ予測モードのイントラ方向パラメータを更に詳しく設定する。

10

20

30

40

50

【0252】

現在ブロックの上方参照サンプル及び左側参照サンプルを一行に配列した後、イントラ方向パラメータの値に基づいて予測サンプルを得ることができる。この場合、イントラ方向パラメータの値が負数である場合、左側参照サンプル及び上方参照サンプルを一行に配列することができる。

【0253】

図25及び図26は、参照サンプルを一行に配列する一次元配列例を示す図である。

【0254】

図25は、垂直方向で参照サンプルを配列する垂直方向一次元アレイの例を示し、また、図26は、水平方向で参照サンプルを配列する水平方向一次元アレイの例を示す。35個のイントラ予測モードを定義したと決定して、図25及び26の実施例を説明する。

10

【0255】

イントラ予測モードインデックスが11から18のうちのいずれか1つである場合、上方参照サンプルを反時計回りに回転する水平方向一次元アレイを適用することができる、また、イントラ予測モードインデックスが19から25のうちのいずれか1つである場合、左側参照サンプルを時計回りに回転する垂直方向一次元アレイを適用することができる。参照サンプルを一行に配列した場合、イントラ予測モード角度を考慮することができる。

【0256】

イントラ方向パラメータに基づいて、参照サンプル決定パラメータを決定することができる。参照サンプル決定パラメータは、参照サンプルを指定するための参照サンプルインデックス及び参照サンプルに適用される加重値を決定するための加重値パラメータを含んでもよい。

20

【0257】

参照サンプルインデックス i_{Idx} 及び加重値パラメータ i_{fact} はそれぞれ、下記等式17及び等式18により得られる。

【数17】

【等式17】

$$i_{Idx} = (y+1) * P_{ang} / 32$$

30

【数18】

【等式18】

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

40

【0258】

等式17及び18において、 P_{ang} は、イントラ方向パラメータを表す。参照サンプルインデックス i_{Idx} により指定された参照サンプルは、整数画素 (Integer pixel) に相当する。

【0259】

予測サンプルを導出するためには、1つ以上の参照サンプルを指定することができる。具体的には、予測モードの勾配を考慮して、予測サンプルを導出するための参照サンプルの

50

位置を指定することができる。例えば、参照サンプルインデックス i_{Idx} を用いて、予測サンプルを導出するための参照サンプルを指定することができる。

【0260】

この場合、イントラ予測モードの勾配が1つの参照サンプルで表されない場合、複数の参照サンプルに対して補間を行うことで予測サンプルを生成することができる。例えば、イントラ予測モードの勾配は、予測サンプルと第1参照サンプルとの勾配と、予測サンプルと第2参照サンプルとの勾配と、の間の値である場合、第1参照サンプル及び第2参照サンプルに対して補間を行うことで予測サンプルを得ることができる。つまり、イントラ予測角度に追従した角度線 (Angular Line) は、整数画素に位置する参照サンプルを経過しない場合、前記角度線が経過した位置の左右又は上下に隣接する参照サンプルに対して補間を行うことで予測サンプルを得ることができる。

10

【0261】

下記等式19は、参照サンプルに基づいて予測サンプルを得る例を示す。

【数19】

【等式19】

$$P(x,y) = ((32-i_{fact})/32) * Ref_1D(x+i_{Idx}+1) + (i_{fact}/32) * Ref_1D(x+i_{Idx}+2)$$

20

【0262】

等式19において、 P は、予測サンプルを表し、また、 Ref_1D は、一次元アレイの参照サンプルのうちのいずれか1つを表す。この場合、予測サンプルの位置 (x, y) 及び参照サンプルインデックス i_{Idx} に基づいて、参照サンプルの位置を決定することができる。

【0263】

イントラ予測モードの勾配は、1つの参照サンプルとして表される場合、加重値パラメータ i_{fact} を0とすることができます。従って、等式19を下記等式20のように簡略化することができる。

30

【数20】

【等式20】

$$P(x,y) = Ref_1D(x+i_{Idx}+1)$$

【0264】

複数のイントラ予測モードに基づいて現在ブロックに対してイントラ予測を実行することもできる。例えば、異なる予測サンプルに対してイントラ予測モードを導出し、また、各予測サンプルに割り当てられたイントラ予測モードに基づいて予測サンプルを導出することができる。

40

【0265】

あるいは、異なる領域に対してイントラ予測モードを導出することができ、また、各領域に割り当てられたイントラ予測モードに基づいて各領域に対してイントラ予測を実行することができる。ここで、前記領域は、少なくとも1つのサンプルを含んでもよい。現在ブロックのサイズ、形状又はイントラ予測モードのうちの少なくとも1つに基づいて、前記領域のサイズ又は形状のうちの少なくとも1つを適応的に決定することができる。あるいは、エンコーダ及びデコーダにおいて、現在ブロックのサイズ又は形状に関わらず、領

50

域のサイズ又は形状のうちの少なくとも 1 つを事前定義することができる。

【 0 2 6 6 】

あるいは、それぞれ、複数のイントラ予測に基づいてイントラ予測を実行することができ、また、複数回のイントラ予測により得られた複数の予測サンプルの平均演算又は加重加算に基づいて、最終的予測サンプルを導出することができる。例えば、第 1 イントラ予測モードに基づいてイントラ予測を実行することで、第 1 予測サンプルを得ることができ、また、第 2 イントラ予測モードに基づいてイントラ予測を実行することで、第 2 予測サンプルを得ることができる。続いて、第 1 予測サンプルと第 2 予測サンプルとの平均演算又は加重加算に基づいて、最終的予測サンプルを得ることができる。この場合、第 1 イントラ予測モードが非角度 / 角度予測モードであるかどうか、第 2 イントラ予測モードが非角度 / 角度予測モードであるかどうか又は隣接ブロックのイントラ予測モードのうちの少なくとも 1 つを考慮することで、第 1 予測サンプル及び第 2 予測サンプルにそれぞれ割り当てられた加重値を決定することができる。

【 0 2 6 7 】

複数のイントラ予測モードは、非角度イントラ予測モードと角度予測モードの組み合わせ、角度予測モードの組み合わせ又は非角度予測モードの組み合わせであってもよい。

【 0 2 6 8 】

図 2 7 は、角度イントラ予測モードと x 軸に垂直である直線との間で形成された角度を示す図である。

【 0 2 6 9 】

図 2 7 に示す例において、角度予測モードは、左下対角線方向と右上対角線方向との間に存在してもよい。x 軸と角度予測モードで形成された角度と記述された場合、角度予測モードは、45 度（左下対角方向）から -135 度（右側上方対角方向）との間に存在してもよい。

【 0 2 7 0 】

現在ブロックが非正方形形状であると、現在ブロックのイントラ予測モードに基づいて、イントラ予測角度に追従した角度線に位置する参照サンプルのうちの、予測サンプルに近接する参照サンプルを用いることなく、予測サンプルから最も離れた参照サンプルを用いて、予測サンプルを導出する。

【 0 2 7 1 】

図 2 8 は、現在ブロックが非正方形である場合に予測サンプルを得る例を示す図である。

【 0 2 7 2 】

例えば、図 2 8 (a) に示す例において、現在ブロックは幅が高さより大きい非正方形形状であり、且つ現在ブロックのイントラ予測モードは角度が 0 度から 45 度である角度イントラ予測モードであると仮定する。この場合、現在ブロックの右側列の近傍の予測サンプル A を導出する場合、前記角度の角度モードでの参照サンプルのうちの、前記予測サンプルから離れた左側参照サンプル L を、前記予測サンプルに近接する上方参照サンプル T の代わりに用いることがある。

【 0 2 7 3 】

もう 1 つの例において、図 2 8 の (b) に示す例において、現在ブロックは高さが幅より大きい非正方形形状であり、且つ現在ブロックのイントラ予測モードは角度が -90 度から -135 度である角度イントラ予測モードであると仮定する。上述した場合、現在ブロックの下方行の近傍の予測サンプル A を導出する場合、前記角度の角度モードでの参照サンプルのうちの、前記予測サンプルから離れた上方参照サンプル T を、前記予測サンプルに近接する左側参照サンプル L の代わりに用いることがある。

【 0 2 7 4 】

上述した問題を解決するために、現在ブロックが非正方形である場合、現在ブロックのイントラ予測モードを逆方向のイントラ予測モードに置き換えることができる。従って、非正方形形状のブロックに対して、図 2 4 に示した角度予測モードの角度より大きいか又は小さい角度を有する角度予測モードを用いることができる。このような角度イントラ予

測モードは、広各イントラ予測モードと定義されてもよい。広角イントラ予測モードは、45度から-135度の範囲に収まらない角度イントラ予測モードを表す。

【0275】

図29は、広角イントラ予測モードを示す図である。

【0276】

図29に示す例において、インデックスが-1から-14であるイントラ予測モード及びインデックスが67から80であるイントラ予測モードは、広角イントラ予測モードを表す。

【0277】

図29に、45度より大きい角度を有する14個の広角イントラ予測モード(-1から-14)及び-135度未満の角度を有する14個の広角イントラ予測モード(67から80)を示したが、より多いか又はより少ない数の広角イントラ予測モードを定義することができる。

10

【0278】

広角イントラ予測モードを用いる場合、上方参照サンプルの長さを $2W + 1$ とし、また、左側参照サンプルの長さを $2H + 1$ とする。

【0279】

広角イントラ予測モードを用いる場合、参照サンプルTを用いて図28(a)に示すサンプルAを予測し、参照サンプルLを用いて図28(b)に示すサンプルAを予測することができる。

20

【0280】

既存のイントラ予測モードとN個の広角イントラ予測モードを加算することで、計 $67 + N$ 個のイントラ予測モードを用いることができる。例えば、表3は、20個の広角イントラ予測モードを定義した場合のイントラ予測モードのイントラ方向パラメータを示す。

20

30

40

50

【表 3】

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					

【0281】

現在ブロックが非正方形形状であり、且つステップ S 2202 で取得された現在ブロックのイントラ予測モードが変換範囲内に収まった場合、現在ブロックのイントラ予測モードを広角イントラ予測モードに変換することができる。現在ブロックのサイズ、形状又は比率のうちの少なくとも 1 つに基づいて前記変換範囲を決定することができる。ここで、前記比率は、現在ブロックの幅と高さとの比率を表すことができる。

【0282】

現在ブロックは、幅が高さより大きい非正方形である場合、変換範囲を右側上方対角方向のイントラ予測モードインデックス（例えば、66）から（右側上方対角方向のイントラ予測モードのインデックス - N）とすることができる。ここで、N は、現在ブロックの比率に基づいて決定されてもよい。現在ブロックのイントラ予測モードが変換範囲内に収まった場合、前記イントラ予測モードを広角イントラ予測モードに変換することができる。前記イントラ予測モードから、事前定義された値を減算することで、前記変換を実行す

10

20

30

40

50

ることができる。前記事前定義された値は、広角イントラ予測モード以外のイントラ予測モードの総数（例えば、67）であってもよい。

【0283】

前記実施例によれば、66番目と53番目との間のイントラ予測モードをそれぞれ-1番目と-14番目との間の広角イントラ予測モードに変換することができる。

【0284】

現在ブロックは、高さが幅より大きい非正方形である場合、変換範囲を、左下対角方向のイントラ予測モードインデックス（例えば、2）から（左下対角方向のイントラ予測モードのインデックス+M）とすることができます。ここで、Mは、現在ブロックの比率に基づいて決定されてもよい。現在ブロックのイントラ予測モードが変換範囲内に収まった場合、前記イントラ予測モードを広角イントラ予測モードに変換することができる。前記イントラ予測モードに事前定義された値を加算することで前記変換を実行することができる。事前定義された値は、広角イントラ予測モード以外の角度イントラ予測モードの総数（例えば、65）であってもよい。

10

【0285】

前記実施例によれば、2番目と15番目との間のイントラ予測モードをそれぞれ67番目と80番目との間の広角イントラ予測モードに変換することができる。

【0286】

以下、変換範囲内に収まったイントラ予測モードを広角イントラ置き換え予測モードと呼ぶ。

20

【0287】

変換範囲は、現在ブロックの比率に基づいて決定されてもよい。例えば、表4及び表5はそれぞれ、広角イントラ予測モード以外の35個のイントラ予測モード及び67個のイントラ予測モードを定義した場合の変換範囲を示す。

30

【表4】

条件	置き換えイントラ予測モード
$W/H = 2$	モード 2, 3, 4
$W/H > 2$	モード 2, 3, 4, 5, 6
$W/H = 1$	無し
$H/W = 1/2$	モード 32, 33, 34
$H/W < 1/2$	モード 30, 31, 32, 33, 34

【表5】

40

条件	置き換えイントラ予測モード
$W/H = 2$	モード 2, 3, 4, 5, 6, 7
$W/H > 2$	モード 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
$W/H = 1$	無し
$H/W = 1/2$	モード 61, 62, 63, 64, 65, 66
$H/W < 1/2$	モード 57, 58, 59, 60, 61, 62, 63, 64, 65, 66

50

【0288】

表4及び表5に示す例のように、変換範囲に収まつた広角イントラ置き換え予測モードの数は、現在ブロックの比率によって異なつてもよい。

【0289】

既存のイントラ予測モードに加えて広角イントラ予測モードを更に用いることに伴い、広角イントラ予測モードの符号化に必要なリソースは増加する。従つて、符号化効率を低下させる可能性がある。従つて、広角イントラ予測モードに対する直接的な符号化の代わりに、広角イントラ予測モードに関わる置き換えイントラ予測モードに対して符号化を行うことで、符号化効率を向上させる。

【0290】

例えば、67番目の広角イントラ予測モードを用いて現在ブロックに対して符号化を行う場合、67番目の広角置き換えイントラ予測モードとした番号2を現在ブロックのイントラ予測モードに符号化することができる。なお、-1番目の広角イントラ予測モードを用いて現在ブロックに対して符号化を行う場合、-1番目の広角置き換えイントラ予測モードとした番号66を現在ブロックのイントラ予測モードに符号化することができる。

【0291】

デコーダは、現在ブロックのイントラ予測モードに対して復号化を行い、復号化されたイントラ予測モードが変換範囲に含まれるかどうかを決定することができる。復号化されたイントラ予測モードが広角置き換えイントラ予測モードである場合、イントラ予測モードを広角イントラ予測モードに変換することができる。

【0292】

あるいは、広角イントラ予測モードで現在ブロックに対して符号化を行う場合、広角イントラ予測モードに対して直接的に符号化を行うこともできる。

【0293】

イントラ予測モードの符号化は、前記MPMリストに基づいて実現してもよい。後述において、MPMリストの設定方法を詳しく説明する。後述する実施例において、角度が45度より大きい10個の広角イントラ予測モード(-1から-10)及び角度が-135度未満である10個の広角イントラ予測モード(67から76)を定義したと仮定する。

【0294】

広角イントラ予測モードで隣接ブロックに対して符号化を行う場合、広角イントラ予測モードに対応する広角置き換えイントラ予測モードに基づいてMPMを設定することができる。例えば、隣接ブロックに対して広角イントラ予測モードで符号化を行う場合、変数c and IntraPredX(Xは、A又はBである)を広角置き換えイントラ予測モードとすることができます。

【0295】

あるいは、現在ブロックの形状に基づいて、MPMの導出方法を決定することができる。例えば、現在ブロックは、幅と高さが同じである正方形形状である場合、c and IntraPredXを広角置き換えイントラ予測モードとすることができます。一方で、現在ブロックが非正方形形状である場合、c and IntraPredXを広角イントラ予測モードとすることができます。

【0296】

あるいは、隣接ブロックの広角イントラ予測モードが現在ブロックに適用できるかどうかに基づいて、c and IntraPredXを広角イントラ予測モードとするかどうかを決定することができる。例えば、現在ブロックは、幅が高さより大きい非正方形形状であると、インデックスが右上対角線方向のイントラ予測モードのインデックスより大きい広角イントラ予測モードをそのままc and IntraPredXとする。インデックスが左下対角線方向のイントラ予測モードのインデックス未満である広角イントラ予測モードに対して、前記広角イントラ予測モードに対応する広角置き換えイントラ予測モードをc and IntraPredXとする。一方で、現在ブロックは、高さが幅より大きい非正方形形状であると、インデックスが左下対角線方向のイントラ予測モードのインデック

10

20

30

40

50

ス未満である広角イントラ予測モードをそのまま `candIntraPredX` とする。インデックスが右上対角線方向のイントラ予測モードのインデックスより大きい広角イントラ予測モードに対して、前記広角イントラ予測モードに対応する広角置き換えイントラ予測モードを `candIntraPredX` とする。

【0297】

つまり、広角イントラ予測モードで符号化された隣接ブロックの形状が現在ブロックの形状と同じであるか又は類似するかに基づいて、広角イントラ予測モードを用いて MPM を導出するかどうか又は広角置き換えイントラ予測モードを用いて MPM を導出するかどうかを決定することができる。

【0298】

あるいは、現在ブロックの形状に関わらず、隣接ブロックの広角イントラ予測モードを `candIntraPredX` とすることができます。

【0299】

要するに、`candIntraPredX` を隣接ブロックの広角イントラ予測モード又は広角置き換えイントラ予測モードとすることができます。

【0300】

`candIntraPredA` 及び `candIntraPredB` に基づいて MPM を導出することができる。この場合、`candIntraPredA` 又は `candIntraPredB` と類似したイントラ予測モードで、MPM を導出することができる。モジュロ演算及びオフセットに基づいて、`candIntraPredA` 又は `candIntraPredB` と類似したイントラ予測モードを導出することができる。この場合、現在ブロックの形状によって、モジュロ演算に用いられる定数及びオフセットを異なるように決定することができる。

【0301】

表 6 は、現在ブロックの形状に基づいて MPM を導出する例を示す。

【表 6】

i) `cuWidth` が `cuHeight` に等しいと

`candModeList[0] = candIntraPredModeA`

`candModeList[1] = 2 + ((candIntraPredModeA + 61) % 64)`

`candModeList[2] = 2 + ((candIntraPredModeA - 1) % 64)`

ii) そうでなければ(つまり、`cuWidth` は `cuHeight` に等しくない)

`candModeList[0] = candIntraPredModeA`

`candModeList[1] = 2 + ((candIntraPredModeA + 81) % 84)`

`candModeList[2] = 2 + ((candIntraPredModeA - 1) % 84)`

【0302】

`candIntraPredA` と `candIntraPredB` が同じであり、且つ `candIntraPredA` が角度イントラ予測モードであると仮定する。現在ブロックが正方形形状である場合、広角イントラ予測モード以外の角度イントラ予測モードの総数から 1 を減算することで得られた値に基づくモジュロ演算により、`candIntraPredA` と類似したイントラ予測モードを得ることができる。例えば、広角イントラ予測モード以外の角度イントラ予測モードの数が 65 である場合、`candIntraPredA` 及び 64 に基づくモジュロ演算で得られた値により、MPM を導出することができる

10

20

30

40

50

。一方で、現在ブロックは、非正方形形状である場合、広角イントラ予測モードを含む角度イントラ予測モードの総数から 1 を減算することで得られた値に基づくモジュロ演算により、`c and IntraPredA` と類似したイントラ予測モードを得ることができる。例えば、広角イントラ予測モードの数が 20 である場合、`c and IntraPredA` 及び 84 に基づくモジュロ演算で得られた値により、MPM を導出することができる。

【0303】

モジュロ演算に用いられる定数は現在ブロックの形状によって異なるように設定されたため、広角イントラ予測モードを `c and IntraPredA` と類似した角度イントラ予測モードと設定できるかどうかを決定することができる。例えば、64 を用いたモジュロ演算において、広角イントラ予測モードを `c and IntraPredA` と類似した角度イントラ予測モードとすることができないことがあるが、84 を用いたモジュロ演算において、広角イントラ予測モードを `c and IntraPredA` と類似した角度イントラ予測モードとすることができます。

10

【0304】

あるいは、`c and IntraPredA` と `c and IntraPredB` が同じである場合、現在ブロックの形状及び `c and IntraPredA` が広角イントラ予測モードであるかどうかを考慮して、MPM を導出することができる。

【0305】

表 7 は、現在ブロックの形状に基づいて MPM を導出する例を示す。

20

30

40

50

【表 7】

- candIntraPredModeB が candIntraPredModeA に等しいと、下記条件を適用する：

- candIntraPredModeA が 2 未満であると(つまり、INTRA_PLANAR 又は INTRA_DC に等しい)、 $x = 0 \dots 2$ の candModeList [x] は、以下のように導出される。

candModeList [0] = INTRA_PLANAR

candModeList [1] = INTRA_DC

candModeList [2] = INTRA_ANGULAR50

- そうでなければ、 $x = 0 \dots 2$ の candModeList [x] は、以下のように導出される：

i) cuWidth が cuHeight に等しいと

1. candIntrapredA が 0 未満であるか又は candIntraPredModeA が 66 より大きいと、

candModeList [0] = INTRA_PLANAR

candModeList [1] = INTRA_DC

candModeList [2] = INTRA_ANGULAR50

2. そうでなければ、 $candIntraPredA \geq 0 \ \&\& \ candIntraPredModeA < 67$ であると、

candModeList [0] = candIntraPredModeA

candModeList [1] = $2 + ((candIntraPredModeA + 61) \% 64)$

candModeList [2] = $2 + ((candIntraPredModeA - 1) \% 64)$

ii) そうでなければ、cuWidth が cuHeight より大きいと

candModeList [0] = candIntraPredModeA

candModeList [1] = $2 + ((candIntraPredModeA + 81) \% 84)$

candModeList [2] = $2 + ((candIntraPredModeA - 1) \% 84)$

iii) そうでなければ(つまり、cuWidth が cuHeight より大きい)

candModeList [0] = candIntraPredModeA

10

20

30

40

【0306】

candModeList [1] = 2 + ((candIntraPredModeA + 81) % 84)

candModeList [2] = -8 + ((candIntraPredModeA - 1) % 84)

- そうでなければ (candIntraPredModeB が candIntraPredModeA に等しくない)、下記条件を適用する：

- candModeList [0] 及び candModeList [1] は、以下のように導出される。

candModeList [0] = candIntraPredModeA

candModeList [1] = candIntraPredModeB

- candModeList [0] 及び candModeList [1] がいずれも INTRA_PLANAR に等しくないと、candModeList [2] を INTRA_PLANAR とする

- そうでなければ、candModeList [0] 及び candModeList [1] がいずれも INTRA_DC に等しくないと、candModeList [2] を INTRA_DC に等しいように設定し、

- そうでなければ、candModeList [2] を INTRA_ANGULAR 50 とする。

10

20

30

【0307】

c and I n t r a P r e d A と c and I n t r a P r e d B が同じであると仮定する。

【0308】

現在ブロックが正方形形状であり、且つ c and I n t r a P r e d A が広角イントラ予測モードである場合、 M P M をデフォルトモードとすることができます。例えば、 M P M [0] 、 M P M [1] 及び M P M [2] をそれぞれ平面モード、 D C モード及び垂直イントラ予測モードとすることができます。

【0309】

現在ブロックが正方形形状であり、且つ c and I n t r a P r e d A が広角イントラ予測モード以外の角度イントラ予測モードである場合、 M P M を c and I n t r a P r e d A と類似した角度イントラ予測モードとすることができます。例えば、 M P M [0] を c and I n t r a P r e d A とすることができます、また、 M P M [1] 及び M P M [2] を c and I n t r a P r e d A と類似した角度イントラ予測モードとすることができます。

【0310】

現在ブロックが非正方形形状であり、且つ c and I n t r a P r e d A が角度イントラ予測モードである場合、 M P M を c and I n t r a P r e d A と類似した角度イントラ予測モードとすることができます。例えば、 M P M [0] を c and I n t r a P r e d A とすることができます、また、 M P M [1] 及び M P M [2] を c and I n t r a P r e d A と類似した角度イントラ予測モードとすることができます。

【0311】

モジュロ演算及びオフセットを用いて、 c and I n t r a P r e d A と類似した角度イントラ予測モードを導出することができます。この場合、モジュロ演算に用いられる定数は、現在ブロックの形状によって異なることがある。なお、現在ブロックの形状に基づいて、 c and I n t r a P r e d A と類似した角度イントラ予測モードを導出するためのオフセットを異なるように設定することができる。例えば、現在ブロックは、幅が高さよ

40

50

り大きい非正方形形状である場合、オフセット 2 を用いることで、c a n d I n t r a P r e d A と類似した角度イントラ予測モードを導出することができる。一方で、現在ブロックは、高さが幅より大きい非正方形形状である場合、オフセット 2 及び - 8 を用いて、c a n d I n t r a P r e d A と類似した角度イントラ予測モードを導出することができる。

【 0 3 1 2 】

あるいは、c a n d I n t r a P r e d X が最大インデックスを有する広角イントラ予測モードであるかそれとも最小インデックスを有する広角イントラ予測モードであるかを考慮することで、M P M を導出することができる。

【 0 3 1 3 】

表 8 は、広角イントラ予測モードインデックスを考慮することでM P M を導出する例を示す。

10

20

30

40

50

【表 8】

- candIntraPredModeB が candIntraPredModeA に等しいと、下記条件を適用する：

- candIntraPredModeA が 2 未満であると(つまり、INTRA_PLANAR 又は INTRA_DC に等しい)、 $x = 0 \dots 2$ の candModeList [x] は以下のように導出される：

```

    candModeList [0] = INTRA_PLANAR
    candModeList [1] = INTRA_DC
    candModeList [2] = INTRA_ANGULAR50
  
```

- そうでなければ、 $x = 0 \dots 2$ の candModeList [x] は以下のように導出される：

- $(\text{candIntraPredModeA} < 0 \ \&\& \ \text{candIntraPredModeA} \neq -10)$ であると、

```

    candModeList [0] = candIntraPredModeA
    candModeList [1] = 2 + ((candIntraPredModeA - 3) \% 84)
    candModeList [2] = 2 + ((candIntraPredModeA - 1) \% 84)
  
```

- $(\text{candIntraPredModeA} = -10)$ であると、

```

    candModeList [0] = candIntraPredModeA
    candModeList [1] = -1
    candModeList [2] = -9
  
```

- $(\text{candIntraPredModeA} > 67 \ \&\& \ \text{candIntraPredModeA} < 76)$ であると、

```

    candModeList [0] = candIntraPredModeA
    candModeList [1] = 2 + ((candIntraPredModeA + 81) \% 84)
    candModeList [2] = 2 + ((candIntraPredModeA - 1) \% 84)
  
```

- $\text{candIntraPredModeA}$ は、67 に等しい

```

    candModeList [0] = candIntraPredModeA
    candModeList [1] = 66
    candModeList [2] = 68
  
```

10

20

30

40

【0314】

50

v) そうでなければ, candIntraPredModeA は、77 に等しい

candModeList [0] = candIntraPredModeA

candModeList [1] = 76

candModeList [2] = 67

vi) そうでなければ、(candIntraPredModeA >= 0 && 10
candIntraPredModeA < 67) であると、

candModeList [0] = candIntraPredModeA

candModeList [1] = 2 + ((candIntraPredModeA + 61) % 64)

candModeList [2] = 2 + ((candIntraPredModeA - 1) % 64)

- そうでなければ (candIntraPredModeB が candIntraPredModeA に等しくない)、下記条件を適用する：

- candModeList [0] 及び candModeList [1] は以下のように導出される：

candModeList [0] = candIntraPredModeA

candModeList [1] = candIntraPredModeB

- candModeList [0] 及び candModeList [1] がいずれも INTRA_PLANAR に等しくないと、candModeList [2] を INTRA_PLANAR とする

- そうでなければ、candModeList [0] 及び candModeList [1] がいずれも INTRA_DC に等しくないと、candModeList [2] を INTRA_DC に等しいように設定し、

- そうでなければ、candModeList [2] を INTRA_ANGULAR 50 とする。

10

20

30

40

【0315】

candIntraPredA と candIntraPredB が同じであると仮定する。説明しやすくするために、インデックス値が左下対角線方向のイントラ予測モードのインデックス値未満である広角イントラ予測モードを下向き広角イントラ予測モードと呼び、また、インデックス値が右上対角線方向のイントラ予測モードのインデックス値より大きい広角イントラ予測モードを右向き広角イントラ予測モードと呼ぶ。

【0316】

candIntraPredA が下向き広角イントラ予測モードである場合、MPM を candIntraPredA と類似した角度イントラ予測モードとすることができます。この場合、candIntraPredA は、最小値を有する下向き広角イントラ予測モードである場合、MPM を、事前定義されたインデックス値を有する下向き広角イントラ予測モードとすることができます。ここで、事前定義されたインデックスは、下向き広角イントラ予測モードのインデックスのうちの最大値を有するインデックスであってもよい。例えば、candIntraPredA が -10 である場合、MPM [0]、MPM [1] 50

] 及び MPM [2] をそれぞれ - 10、 - 1 及び - 9 とすることができます。

【 0 3 1 7 】

c and I n t r a P r e d A が右向き広角イントラ予測モードである場合、 MPM を c and I n t r a P r e d A と類似した角度イントラ予測モードとすることができます。この場合、 c and I n t r a P r e d A は、最大値を有する右向き広角イントラ予測モードである場合、 MPM を、事前定義されたインデックス値を有する右向き広角イントラ予測モードとすることができます。ここで、事前定義されたインデックスは、右向き広角イントラ予測モードのインデックスのうちの最小値を有するインデックスであってもよい。例えば、 c and I n t r a P r e d A が 77 である場合、 MPM [0] 、 MPM [1] 及び MPM [2] をそれぞれ 77、 76 及び 67 とすることができます。

10

【 0 3 1 8 】

あるいは、 c and I n t r a P r e d A のインデックスから 1 を減算することで得られたインデックスはイントラ予測モードのインデックスのうちの最小値未満であるか、又は、 1 を加算することで得られたインデックスは最大値より大きい場合、 MPM をデフォルトモードとすることができます。ここで、デフォルトモードは、平面モード、 DC モード、垂直イントラ予測モード、水平イントラ予測モード及び対角イントラ予測モードの少なくとも 1 つを含んでもよい。

【 0 3 1 9 】

あるいは、 c and I n t r a P r e d A のインデックスから 1 を減算することで得られたインデックスはイントラ予測モードのインデックスのうちの最小値未満であるか、又は、 1 を加算することで得られたインデックスは最大値より大きい場合、 MPM を c and I n t r a P r e d A と逆であるイントラ予測モードとするか又は c and I n t r a P r e d A と逆であるイントラ予測モードと類似したイントラ予測モードとすることができます。

20

【 0 3 2 0 】

あるいは、現在ブロックの形状及び隣接ブロックの形状を考慮して MPM 候補を導出することができる。例えば、現在ブロック及び隣接ブロックがいずれも非正方形形状である場合に MPM を導出する方法は、現在ブロックが正方形形状であるが、隣接ブロックが非正方形形状である場合に MPM を導出する方法と異なることがある。

30

【 0 3 2 1 】

現在ブロックのサイズ、現在ブロックの形状、隣接ブロックのサイズ及び隣接ブロックの形状のうちの少なくとも 1 つを考慮して MPM リストにおける MPM を再配列（又は再順序付け）することができます。ここで、再配列は、各 MPM に割り当てられたインデックスを再割り当てる음을表す。例えば、小さいインデックスを、現在ブロックのサイズ又は形状と同じであるサイズ又は形状を有する隣接ブロックのイントラ予測モードと同じである MPM に割り当てることができる。

【 0 3 2 2 】

MPM [0] 及び MPM [1] をそれぞれ、左隣接ブロックのイントラ予測モード c and I n t r a P r e d A 及び上隣接ブロックのイントラ予測モード c and I n t r a P r e d B とすると仮定する。

40

【 0 3 2 3 】

現在ブロック及び上隣接ブロックは幅が高さより大きい非正方形形状である場合、 MPM を、上隣接ブロックのイントラ予測モード c and I n t r a P r e d B に小さいインデックスを持たせるように再配列することができる。つまり、 c and I n t r a P r e d B を MPM [0] に再配列し、また、 c and I n t r a P r e d A を MPM [1] に再配列することができる。

【 0 3 2 4 】

あるいは、現在ブロック及び上隣接ブロックは高さが幅より大きい非正方形形状である場合、 MPM を、上隣接ブロックのイントラ予測モード c and I n t r a P r e d B に小さいインデックスを持たせるように再配列することができる。つまり、 c and I n t

50

`raPredB`を`MPM[0]`に再配列し、また、`candIntraPredA`を`MPM[1]`に再配列することができる。

【0325】

あるいは、現在ブロック及び上隣接ブロックが正方形形状である場合、`MPM`を、上隣接ブロックのイントラ予測モード`candIntraPredB`に小さいインデックスを持たせるように再配列することができる。つまり、`candIntraPredB`を`MPM[0]`に再配列し、また、`candIntraPredA`を`MPM[1]`に再配列することができる。

【0326】

`MPM`の再配列の代わりに、最初に`candIntraPredX`を`MPM`に割り当てる場合、現在ブロックのサイズ、現在ブロックの形状、隣接ブロックのサイズ及び隣接ブロックの形状のうちの少なくとも1つを考慮することができる。

10

【0327】

現在ブロックのサイズ又は形状に基づいて`MPM`を再配列することができる。例えば、現在ブロックは、幅が高さより大きい非正方形形状であると、降順に応じて`MPM`を再配列することができる。一方で、現在ブロックは、高さが幅より大きい非正方形形状であると、昇順に応じて`MPM`を再配列することができる。

【0328】

オリジナル画像と予測画像を減算することで、導出される残差画像を導出することができる。この場合、残差画像を周波数領域に変更する場合、周波数成分のうちの高周波数成分を除去しても、ビデオの主観画質を大幅に低下させることができない。従って、高周波数成分の値を小さい値に変換するか又は高周波数成分の値を0とすると、明らかな視覚的歪みを引き起こすことなく、圧縮効率を向上させるという効果を有する。上記特性を反映するために、現在ブロックを変換することで残差画像を2次元周波数成分に分解することができる。離散コサイン変換(`DCT`: Discrete Cosine Transform)又は離散サイン変換(`DST`: Discrete Sine Transform)等の変換技術を用いて前記変換を実行することができる。

20

【0329】

`DCT`は、コサイン変換を用いて残差画像を2次元周波数成分に分解(又は変換)する。`DST`は、サイン変換を用いて残差画像を2次元周波数成分に分解(又は変換)する。残差画像の変換結果として、周波数成分は、基本画像として表されてもよい。例えば、サイズが $N \times N$ であるブロックに対して`DCT`変換を実行する場合、 N^2 個の基本パターン成分を得ることができる。変換により、サイズが $N \times N$ であるブロックに含まれる各基本パターン成分のサイズを得ることができる。用いられる変換技術によれば、基本パターン成分のサイズを`DCT`係数又は`DST`係数と呼んでもよい。

30

【0330】

変換技術`DCT`は主に、0ではない低周波数成分が多く分布した画像に対して変換を行うために用いられる。変換技術`DST`は主に、高周波数成分が多く分布した画像に用いられる。

40

【0331】

`DCT`又は`DST`以外の変換技術を用いて残差画像を変換することもできる。

【0332】

以下、残差画像を2次元周波数成分に変換するプロセスを2次元画像変換と呼ぶ。なお、変換により得られた基本パターン成分のサイズを変換係数と呼んでもよい。例えば、変換係数は、`DCT`係数又は`DST`係数であってもよい。後述する主変換及び二次変換を同時に適用する場合、変換係数は、二次変換の結果により生成された基本パターン成分のサイズを表すことができる。

【0333】

ブロックを単位として変換技術を決定することができる。現在ブロックの予測符号化モード、現在ブロックのサイズ又は現在ブロックの形状のうちの少なくとも1つに基づいて

50

、変換技術を決定することができる。例えば、イントラ予測モードで、現在ブロックに対して符号化を行い、且つ現在ブロックのサイズが $N \times N$ 未満である場合、変換技術を用いて D S T を実行することができる。一方で、前記条件を満たすことができない場合、変換技術 D C T を用いて変換を実行することができる。

【 0 3 3 4 】

残差画像において、一部のブロックに対して 2 次元画像変換を実行しなくてもよい。2 次元画像変換を実行しないことは、変換スキップ (T r a n s f o r m S k i p) と呼ばれてもよい。変換スキップを適用する場合、量子化を変換が実行されていない残差値に適用することができる。

【 0 3 3 5 】

D C T 又は D S T を用いて現在ブロックを変換した後、変換された現在ブロックに対して再び変換を行うことができる。この場合、D C T 又は D S T に基づく変換を主変換と定義し、また、主変換を適用するブロックに対して再び変換を行うプロセスを二次変換と呼ぶ。

【 0 3 3 6 】

主変換は、複数の変換カーネル候補のうちのいずれか 1 つを用いて実行することができる。例えば、D C T 2 、 D C T 8 又は D C T 7 のうちのいずれか 1 つを用いて主変換を実行することができる。

【 0 3 3 7 】

水平方向及び垂直方向に対して、異なる変換カーネルを用いることもできる。ビットストリームを介して、信号で、水平方向の変換カーネルと垂直方向の変換カーネルの組み合わせを表す情報を送信することもできる。

【 0 3 3 8 】

主変換及び二次変換の実行ユニットは、異なる。例えば、 8×8 ブロックに対して主変換を実行し、また、変換された 8×8 ブロックにおけるサイズが 4×4 であるサブブロックに対して二次変換を実行することができる。この場合、二次変換を実行しない余剰領域の変換係数を 0 とすることもできる。

【 0 3 3 9 】

あるいは、 4×4 ブロックに対して主変換を実行し、また、変換された 4×4 ブロックを含むサイズが 8×8 である領域に対して二次変換を実行することができる。

【 0 3 4 0 】

ビットストリームを介して、信号で、二次変換を実行するかどうかを表す情報を送信することができる。

【 0 3 4 1 】

デコーダにおいて、二次変換の逆変換（第 2 逆変換）を実行し、また、その結果に対して主変換の逆変換（第 1 逆変換）を実行することができる。第 2 逆変換及び第 1 逆変換の実行結果として、現在ブロックの残差信号を得ることができる。

【 0 3 4 2 】

量子化は、ブロックのエネルギーの減少に用いられ、また、量子化プロセスは、変換係数を特定の定数で除算するプロセスを含む。前記定数は、量子化パラメータにより導出されてもよく、また、量子化パラメータは、1 から至 6 3 の値と定義されてもよい。

【 0 3 4 3 】

エンコーダにおいて変換及び量子化を実行すると、デコーダは、逆量子化及び逆変換により、残差ブロックを得ることができる。デコーダは、予測ブロックと残差ブロックを加算することで、現在ブロックの再構成ブロックを得ることができる。

【 0 3 4 4 】

現在ブロックの再構成ブロックを得ると、インループフィルタリング (I n - l o o p f i l t e r i n g) により、量子化及び符号化プロセスにおいて発生した情報紛失を減少させることができる。インループフィルタは、デブロッキングフィルタ (D e b l o c k i n g f i l t e r) 、サンプル適応型オフセットフィルタ (S a m p l e A d a p

10

20

30

40

50

tive Offset filter: S A O) 又は適応型ループフィルタ (A d a p t i v e Loop Filter: A L F) のうちの少なくとも 1 つを含んでもよい。以下、インループフィルタを適用する前の再構成ブロックを第 1 再構成ブロックと呼び、また、インループフィルタを適用した後の再構成ブロックを第 2 再構成ブロックと呼ぶことができる。

【 0 3 4 5 】

第 1 再構成ブロックに対してデブロッキングフィルタ、 S A O 又は A L F のうちの少なくとも 1 つを適用することで、第 2 再構成ブロックを得ることができる。この場合、デブロッキングフィルタを適用した後に、 S A O 又は A L F を適用することができる。

【 0 3 4 6 】

デブロッキングフィルタは、ブロックを単位として量子化を実行する場合にブロックの境界において発生した画質劣化 (ブロッキングアーチファクト : B l o c k i n g A r t i f a c t) を軽減するために用いられる。デブロッキングフィルタを適用するために、第 1 再構成ブロックと隣接再構成ブロックとの間のブロック強度 (B l o c k i n g S t r e n g t h : B S) を決定することができる。

【 0 3 4 7 】

図 3 0 は、ブロック強度の決定プロセスを示すフロー チャートである。

【 0 3 4 8 】

図 3 0 に示す例において、 P は、第 1 再構成ブロックを表し、また、 Q は、隣接再構成ブロックを表す。ここで、隣接再構成ブロックは、現在ブロックの左側又は上方に隣接してもよい。

【 0 3 4 9 】

図 3 0 に示す例において、 P 及び Q の予測符号化モード、非ゼロ変換係数を含むかどうか、同じ参照画像を用いてインター予測を実行するかどうか、及び動きベクトルの差分値が閾値以上であるかどうかを考慮することで、ブロック強度を決定することができる。

【 0 3 5 0 】

ブロック強度に基づいて、デブロッキングフィルタを適用したかどうかを決定することができる。例えば、ブロック強度が 0 である場合、フィルタリングを実行しない可能性がある。

【 0 3 5 1 】

S A O は、周波数領域において量子化を実行する場合に発生したリングアーチファクト (R i n g i n g A r t i f a c t) を軽減するために用いられる。第 1 再構成画像を考慮したパターンを加算するか又は減算することで決定されたオフセットにより、 S A O を実行することができる。オフセットの決定方法は、エッジオフセット (E d g e O f f s e t : E O) 又はバンドオフセット (B a n d O f f s e t) を含む。 E O は、周辺画素のパターンに基づいて現在サンプルのオフセットを決定する方法を表す。 B O は、領域における類似した輝度値を有する画素の集合に対して共通のオフセットを適用する方法を表す。具体的には、画素輝度を 3 2 個の均等な区間に分割し、また、類似した輝度値を有する画素を 1 つの集合とすることができます。例えば、 3 2 個のバンドのうちの隣接する 4 個のバンドを一組とし、また、 4 個のバンドに属するサンプルに対して、同じオフセットを適用することができる。

【 0 3 5 2 】

A L F は、第 1 再構成画像又はデブロッキングフィルタを適用した再構成画像に対して、事前定義されたサイズ / 形状のフィルタを適用することで、第 2 再構成画像を生成する方法である。下記等式 2 1 は、 A L F の適用例を示す。

【 数 2 1 】

10

20

30

40

50

【等式 2 1】

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

10

【0 3 5 3】

画像、符号化ツリーユニット、符号化ブロック、予測ブロック又は変換ブロックを単位として、事前定義されたフィルタ候補のうちのいずれか 1 つを選択することができる。各フィルタ候補のサイズ又は形状のうちのいずれか 1 つは異なってもよい。

【0 3 5 4】

図 3 1 は、事前定義されたフィルタ候補を示す図である。

【0 3 5 5】

図 3 1 に示す例において、 5×5 、 7×7 及び 9×9 菱形のうちの少なくとも 1 つを選択することができる。

【0 3 5 6】

色度成分に対して、サイズが 5×5 である菱形のみを用いることができる。

20

【0 3 5 7】

パノラマビデオ、360 度ビデオ又は 4K / 8K UHD (超精密度) 等などの高解像度ビデオに対して、リアルタイム符号化又は低遅延符号化を実行するために、1 枚の画像を複数の領域に分割し、前記複数の領域に対して符号化 / 復号化を行うことができる。このために、画像をタイル (即ち、並行して符号化 / 復号化した基本ユニット) に分割し、またこれらのタイルを並行して処理することができる。

【0 3 5 8】

タイルを、矩形形状を有するものに制限することができる。タイルに対して符号化 / 復号化を行う場合、他のタイルのデータを使用しない。タイルを単位として、コンテキスト適応型二値算術符号化 (CABAC) コンテキストの確率テーブルを初期化し、また、それを、タイルの境界でループフィルタを適用しないように設定することができる。

30

【0 3 5 9】

図 3 2 は、画像を複数のタイルに分割する例を示す。

【0 3 6 0】

タイルは、少なくとも 1 つの符号化ツリーユニットを含み、また、タイルの境界は、符号化ツリーユニットの境界と重なり合う。

【0 3 6 1】

図 3 2 に示す例において、画像は、複数のタイル集合に分割されてもよい。ビットストリームを介して、信号で、画像を複数のタイル集合に分割するための情報を送信することができる。

40

【0 3 6 2】

画像の分割タイプによれば、タイルは、画像境界以外の全ての領域において、同一のサイズを有してもよい。

【0 3 6 3】

あるいは、画像を、水平方向に互いに隣接するタイルが同一の高さを有するように分割することができ、又は、画像を、垂直方向に互いに隣接するタイルが同一の幅を有するように分割することができる。

【0 3 6 4】

画像と交わる垂直線又は水平線のうちの少なくとも 1 本の線を用いて画像を分割する場

50

合、各タイルはいずれも、異なる及び／又は行に属する。後述する例示的な実施例において、タイルが属する列は、タイル列と呼ばれ、また、タイルが属する行は、タイル行と呼ばれる。

【0365】

ビットストリームを介して、信号で、画像をタイルに分割する形状を決定するための情報を送信することができる。画像パラメータ集合又はシーケンスパラメータ集合により。情報に対して符号化を行い、信号で送信することができる。該情報は、画像におけるタイルの数を決定するために用いられ、また、タイル行の数を示す情報及びタイル列の数を示す情報を含んでもよい。例えば、シンタックス要素 `num_tile_columns_minus_1` は、タイル列の数から 1 を減算することで得られた値を示し、また、シンタックス要素 `num_tile_rows_minus_1` は、タイル行の数から 1 を減算することで得られた値を示す。

【0366】

図32に示す例において、タイル列の数が 4 であり、且つタイル行の数が 3 であるため、`num_tile_columns_minus_1` は 3 であってもよく、また `num_tile_rows_minus_1` は、2 であってもよい。

【0367】

画像を複数のタイルに分割する場合、ビットストリームを介して、信号で、タイルのサイズを示す情報を送信することができる。例えば、画像を複数のタイル列に分割する場合、ビットストリームを介して、信号で、各タイル列の幅を示す情報を送信する。また、画像を複数のタイル行に分割する場合、ビットストリームを介して、信号で、各タイル行の高さを示す情報を送信する。例えば、各タイル列に対して、タイル列の幅を示すシンタックス要素 `column_width_minus_1` を符号化し、信号で送信することができる。また、各タイル行に対して、タイル行の高さを示すシンタックス要素 `row_height_minus_1` を符号化し、信号で送信することができる。

【0368】

`column_width_minus_1` は、タイル列の幅から 1 を減算することで得られた値を示すことができる。また、`row_height_minus_1` は、タイル行の高さから 1 を減算することで得られた値を示すことができる。

【0369】

最後のタイル列に対して、`column_width_minus_1` に対する符号化を省略してもよく、また、最後のタイル行に対して、`row_height_minus_1` に対する符号化を省略してもよい。画像のサイズを考慮して最後のタイル列の幅及び最後の一行の高さを導出することができる。

【0370】

デコーダは、`column_width_minus_1` 和 `row_height_minus_1` に基づいてタイルのサイズを決定することができる。

【0371】

表9は、画像をタイルに分割するシンタックステーブルを示す。

10

20

30

40

50

【表 9】

...	
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)
}	
...	

10

20

30

【0372】

表9を参照すると、信号で、タイル列の数を示すシンタックス要素num_tile_columns_minus1及びタイル行の数を示すシンタックス要素num_tile_rows_minus1を送信することができる。

【0373】

続いて、信号で、画像が、等しいサイズのタイルに分割されたかどうかを示すシンタックス要素uniform_spacing_flagを送信することができる。uniform_spacing_flagが真である場合、画像境界以外の他の領域におけるタイルを等しいサイズのものに分割することができる。

【0374】

uniform_spacing_flagが偽である場合、信号で、各タイル列の幅を示すシンタックス要素column_width_minus1及び各タイル行の高さを示すシンタックス要素row_height_minus1を送信することができる。

【0375】

シンタックス要素loop_filter_across_tiles_enabled_flagは、タイル境界でループフィルタの適用を許容するかどうかを示す。

【0376】

タイル列のうちの最小幅を有するタイル列は、最小幅タイルと呼ばれてもよく、また、タイル行のうちの最小高さを有するタイル行は、最小高さタイルと呼ばれてもよい。ビットストリームを介して、信号で、最小幅タイルの幅を示す情報及び最小高さタイルの高さを示す情報を送信することができる。例えば、シンタックス要素min_column_width_minus1は、最小幅タイルの幅から1を減算することで得られた値を示し、また、シンタックス要素min_row_height_minus1は、最小高さタイルの高さから1を減算することで得られた値を示す。

40

50

【0377】

各タイル列に対して、信号で、最小タイル幅との差分値を示す情報を送信することができる。例えば、シンタックス要素 `diff_column_width` は、現在タイル列と最小タイル列との幅差分値を示す。幅差分値は、符号化ツリーユニット列の数の差分値として表されてもよい。デコーダは、`min_column_width_minus1` に基づいて導出された最小幅タイルの幅と `diff_column_width` に基づいて導出された幅差分値を加算することで、現在タイルの幅を導出することができる。

【0378】

なお、各タイル行に対して、信号で、最小タイル高さとの差分値を示す情報を送信することができる。例えば、シンタックス要素 `diff_row_height` は、現在タイル行と最小タイル行との高さ差分値を示す。高さ差分値は、符号化ツリーユニット行の数の差分値として表されてもよい。デコーダは、`min_row_height_minus1` に基づいて導出された最小高さタイルの高さと `diff_row_height` に基づいて導出された高さ差分値を加算することで、現在タイルの高さを導出することができる。

10

【0379】

表10は、サイズ差異に関わる情報を含むシンタックステーブルを示す。

【表10】

...		20
<code>if (tile_enabled_flag) {</code>		
<code>num_tile_columns_minus1</code>	<code>ue (v)</code>	
<code>num_tile_rows_minus1</code>	<code>ue (v)</code>	
<code>uniform_spacing_flag</code>	<code>u (1)</code>	
<code>if (!uniform_spacing_flag) {</code>		
<code>min_column_width_minus1</code>		30
<code>min_row_height_minus1</code>		
<code>for (i=0, i < num_tile_columns_minus1: i++)</code>		
<code>diff_column_width[i]</code>	<code>ue (v)</code>	
<code>for (i=0, i < num_tile_rows_minus1: i++)</code>		
<code>diff_row_height[i]</code>	<code>ue (v)</code>	
<code>}</code>		40
<code>loop_filter_across_tiles_enabled_flag</code>	<code>u (1)</code>	
<code>}</code>		
...		

【0380】

画像を、水平に隣接するタイルと異なる高さを有するように分割することができ、又は、画像を、垂直に隣接するタイルと異なる幅を有するように分割することができる。上記

50

画像分割方法は、柔軟なタイル (Flexible Tile) 分割方法と呼ばれてもよく、柔軟なタイル分割方法で分割されたタイルは、柔軟なタイルと呼ばれてもよい。

【0381】

図33は、柔軟なタイル技術による画像分割モードを示す図である。

【0382】

画像を分割することで生成されたタイルの探索順は、所定の走査順に従ってもよい。なお、所定の走査順に応じて、インデックスを各タイルに割り当てることができる。

【0383】

タイルの走査順は、ラスタ走査、対角線走査、垂直走査又は水平走査のうちのいずれか1つであってもよい。図33(a)から図33(d)は、それぞれラスタ走査、対角線走査、垂直走査及び水平走査に基づいて、インデックスを各タイルに割り当てる例を示す。

10

【0384】

現在タイルのサイズ又は位置に基づいて次の走査順を決定することができる。例えば、現在タイルの高さは、現在タイルの右側に隣接するタイルの高さと異なる場合(例えば、右隣接タイルの高さは、現在タイルの高さより大きい場合)、現在タイルの下方に隣接するタイルの垂直線と同じである垂直線でのタイルのうちの最左側に位置するタイルは、現在タイルの後の走査対象として決定されてもよい。

【0385】

画像又はシーケンスを単位としてタイルの走査順を決定することができる。

【0386】

あるいは、画像における第1タイルのサイズを考慮することで、タイルの走査順を決定することができる。例えば、第1タイルの幅が高さより大きい場合、タイルの走査順を水平走査とすることができます。第1タイルの高さが幅より大きい場合、タイルの走査順を垂直走査とすることができます。第1タイルの幅が高さと同じである場合、タイルの走査順をラスタ走査又は対角線走査とすることができます。

20

【0387】

ビットストリームを介して、信号で、タイルの総数を示す情報を送信することができる。例えば、柔軟なタイル技術を適用する場合、信号で、画像におけるタイルの総数から2を減算することで導出されたシンタックス要素 `number_of_tiles_in_picture_minus_2` を送信することができる。デコーダは、`number_of_tiles_in_picture_minus_2` に基づいて、現在画像に含まれるタイルの数を認識することができる。

30

【0388】

表11は、タイルの数に関する情報を含むシンタックステーブルを示す。

40

50

【表 11】

pic_parameter_set_rbsp() {	記述子
...	
tiles_enabled_flag	u(1)
...	
if(tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i=0; i < number_of_tiles_in_picture_minus2 +2; i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0) {	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
...	

【0389】

タイルのサイズの符号化に必要なビット数を減少させるために、サブタイルのサイズを示す情報を符号化し、信号で送信することができる。サブタイルは、タイルを構成する基本ユニットであり、また、各タイルは、少なくとも1つのサブタイルを含むように設定されてもよい。サブタイルは、1つ又は複数の符号化ツリーユニットを含んでもよい。

【0390】

例えば、シンタックス要素 subtile_width_minus1 は、サブタイルの幅から1を減算することで得られた値を示す。シンタックス要素 subtile_height_minus1 は、サブタイルの高さから1を減算することで得られた値を示す。

【0391】

第1タイル以外の他のタイルが直前のタイルと同一のサイズを有するかどうかを示す情報を符号化し、信号で送信することができる。例えば、シンタックス要素 use_previous_tile_size_flag は、現在タイルのサイズが直前のタイルのサイズと同じであるかどうかを示す。use_previous_tile_size_f

10

20

30

40

50

`flag` が真である場合、現在タイルのサイズが直前のタイルのサイズと同じであることを示す。`use_previous_tile_size_flag` が偽である場合、現在タイルのサイズを示す情報を符号化し、信号で送信することができる。1番目のタイルに対して、`use_previous_tile_size_flag` の符号化を省略してもよく、また、フラグの値を偽とすることができます。

【0392】

タイルのサイズを示す情報は、`i` 番目のタイルの幅を示すシンタックス要素 `tile_width_minus1[i]` 及び `i` 番目のタイルの高さを示すシンタックス要素 `tile_height_minus1[i]` を含んでもよい。

【0393】

タイルのサイズを示す情報は、子タイルのサイズとの差分値を示すことができる。サブタイルのサイズ情報を用いる場合、各タイルのサイズの符号化に必要なビット数を減少することで、符号化 / 復号化効率を向上させることができる。例えば、下記等式 2-2 に基づいて、`i` 番目のタイルの幅 `tileWidth` を導出することができ、また、下記等式 2-3 に基づいて、`i` 番目のタイルの高さ `tileHeight` を導出することができる。

【数 2-2】

$$tileWidth = (subtileWidth_{minus1} + 1) * (tileWidth_{minus1}[i] + 1)$$

10

20

【数 2-3】

【等式 2-3】

$$tileHeight = (subtileHeight_{minus1} + 1) * (tileHeight_{minus1}[i] + 1)$$

【0394】

あるいは、サブタイルのサイズ情報の符号化を省略してもよく、また、`i` 番目のタイルのサイズをそのままタイルサイズ情報に符号化することができる。サブタイルのサイズ情報を選択的に符号化することができる。ビデオパラメータ集合、シーケンスパラメータ集合又は画像パラメータ集合を介して、信号で、サブタイルのサイズ情報が符号化されたかどうかを示す情報を送信することができる。

30

【0395】

タイルのサイズに関する情報は、符号化ツリーユニットの数を示すものに符号化されて、信号で送信されてもよい。例えば、`columnWidth_minus1`、`min_columnWidth_minus1`、`subtileWidth_minus1`、`tileWidth_minus1` 等は、タイルに含まれる符号化ツリーユニット列の数を示すことができる。なお、`diffColumnWidth` は、最小幅タイルに含まれる符号化ツリーユニット列の数と現在タイルに含まれる符号化ツリーユニット列の数との差分値を示すことができる。

40

【0396】

なお、`rowHeight_minus1`、`minRowHeight_minus1`、`subtileHeight_minus1`、`tileHeight_minus1` 等は、タイルに含まれる符号化ツリーユニット行の数を示すことができる。なお、`diffRowHeight` は、最小高さタイルに含まれる符号化ツリーユニット行の数と現在タイルに含まれる符号化ツリーユニット行の数との差分値を示すことができる。

【0397】

デコーダは、シンタックス要素に基づいて導出された符号化ツリーユニット列の数及び

50

/ 又は符号化ツリーユニット行の数、及び符号化ツリーユニットのサイズによって、タイルのサイズを決定することができる。例えば、*i* 番目のタイルの幅は、(*tile_width_minus_1* [*i*] + 1) * (符号化ツリーユニットの幅) とされてもよく、また、*i* 番目のタイルの高さは、(*tile_height_minus_1* [*i*] + 1) * (符号化ツリーユニットの高さ) とされてもよい。

【 0 3 9 8 】

それと同時に、シーケンスパラメータ集合又は画像パラメータ集合を介して、信号で、符号化ツリーユニットのサイズを示す情報を送信することができる。

【 0 3 9 9 】

表 1 1 において、シンタックス要素 *use_previous_tile_size_flag* を用いて、現在タイルのサイズが直前のタイルのサイズと同じであるかどうかを示すことを説明する。もう 1 つの例として、現在タイルの幅が直前のタイルの幅と同じであるかどうかを示す情報又は現在タイルの高さが直前のタイルの高さと同じであるかどうかを示す情報を符号化し、信号で送信することができる。

10

【 0 4 0 0 】

表 1 2 は、現在タイルの幅が直前のタイルの幅と同じであるかどうかを示す情報を含むシンタックステーブルを示す。

20

30

40

50

【表 1 2】

記述子	pic_parameter_set_rbsp() {
	...
u (1)	tiles_enabled_flag
	...
	CoveredWidthByTile = 0;
	CoveredHeightByTile = 0;
	PrevWidthByTile = 0;
	PrevHeightByTile = 0;
	if (tiles_enabled_flag) {
ue (v)	number_of_tiles_in_picture_minus2
	for (i=0 , i < number_of_tiles_in_picture_minus2+2 , i++) {
	if (i > 0)
u (1)	use_previous_tile_size_flag
	if (use_previous_tile_size_flag == 0)
	[
ue (v)	use_previous_tile_width_flag
ue (v)	if (use_previous_tile_width_flag == 0) {
ue (v)	tile_width_minus1[i]
ue (v)	tile_height_minus1[i]
	}
	}
ue (v)	tile_height_minus1[i]
	}
u (1)	loop_filter_across_tiles_enabled_flag
	}
	...

10

20

30

40

【0 4 0 1】

シンタックス要素 `use_previous_tile_width_flag` は、現在タイルの幅が直前のタイルの幅と同じであるかどうかを示す。`use_previous_tile_width_flag` が真である場合、現在タイルの幅を、直前のタイルの幅に等しいように設定することができる。この場合、現在タイルの幅を示す情報の符号化を省略してもよく、また、直前のタイルの幅から、現在タイルの幅を導出することができる。

【0 4 0 2】

`use_previous_tile_width_flag` が偽である場合、信号で

50

、現在タイルの幅を示す情報を送信することができる。例えば、`tile_width_minus1[i]`は、`i`番目のタイルの幅から1を減算することで得られた値を示すことができる。

【0403】

現在タイルのサイズが直前のタイルのサイズと異なると判定した場合（例えば、`use_previous_tile_size_flag`の値が0である場合）のみに、シンタックス要素`use_previous_tile_width_flag`を符号化して信号で送信することができる。

【0404】

`tile_width_minus1[i]`は、`i`番目のタイルに含まれる符号化ツリーユニット列の列から1を減算することで得られた値を有してもよい。デコーダは、`tile_width_minus1[i]`に1を加算することで、`i`番目のタイルに属する符号化ツリーユニットの列の数を導出し、導出された値に符号化ツリーユニットの幅を乗算することで、タイルの幅を算出することができる。

10

【0405】

表13は、現在タイルの高さが直前のタイルの高さと同じであるかどうかを示す情報を更に含むシンタックステーブルを示す。

20

30

40

50

【表 1 3】

記述子	
...	
tiles_enabled_flag	u (1)
...	
CoveredWidthByTile = 0,	
CoveredHeightByTile = 0 ,	
PrevWidthByTile = 0,	
PrevHeightByTile = 0 ,	
if (tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue (v)
for (i=0 , i < number_of_tiles_in_picture_minus2+2 , i++) {	
if (i > 0)	
use_previous_tile_size_flag	u (1)
if (use_previous_tile_size_flag == 0)	
{	
use_previous_tile_width_flag	ue (v)
if (use_previous_tile_width_flag == 0) {	
tile_width_minus1[i]	ue (v)
tile_width_minus1[i] +=1,	
use_previous_tile_height_flag	
if (use_previous_tile_height_flag ==0) {	
tile_height_minus1[i]	ue (v)
tile_height_minus1[i] +=1,	
}	
} else {	

10

20

30

40

【0 4 0 6】

use_previous_tile_height_flag	
if (use_previous_tile_height_flag	
==0) {	
tile_height_minus1[i]	ue (v)
tile_height +=1,	
}	
}	
}	
loop_filter_across_tiles_enabled_flag	u (1)
}	
...	

10

20

【0407】

シンタックス要素 `use_previous_tile_height_flag` は、現在タイルの高さが直前のタイルの高さと同じであるかどうかを示す。当 `use_previous_tile_height_flag` が真である場合、現在タイルの高さを、直前のタイルの高さに等しいように設定することができる。この場合、現在タイルの高さを示す情報の符号化を省略してもよく、また、直前のタイルの高さから、現在タイルの高さを導出することができる。

【0408】

`use_previous_tile_height_flag` が偽である場合、信号で、現在タイルの高さを示す情報を送信することができる。例えば、`tile_height_minus1[i]` は、`i` 番目のタイルの高さから 1 を減算することで得られた値を示すことができる。

30

【0409】

現在タイルのサイズが直前のタイルのサイズと異なると判定した場合（例如、当 `use_previous_tile_size_flag` の値が 0 である場合）のみに、シンタックス要素 `use_previous_tile_height_flag` を符号化して信号で送信することができる。なお、`use_previous_tile_width_flag` が偽である場合のみに、信号でシンタックス要素 `use_previous_tile_height_flag` を送信する。

40

【0410】

表 12 は、`use_previous_tile_width_flag` を用いる例を示し、また、表 13 は、`use_previous_tile_width_flag` 及び `use_previous_tile_height_flag` を用いる例を示す。上記表に示されていないが、`use_previous_tile_width_flag` の符号化を省略してもよく、また、`use_previous_tile_width_flag` 及び `use_previous_tile_height_flag` のみを用いることができる。

【0411】

タイル走査順、第 1 タイルの幅及び高さ、直前のタイルの幅及び高さのうちの少なくとも 1 つに基づいて、`use_previous_tile_height_flag` 及び

50

`use_previous_tile_size_flag` のうちのどちらを用いるかを決定することができる。例えば、タイルの走査順が垂直方向である場合、`use_previous_tile_height_flag` を用いることができ、タイルの走査順が水平方向である場合、`use_previous_tile_width_flag` を用いることができる。あるいは、第1タイル又は直前のタイルは、幅が高さより大きい非正方形形状である場合、`use_previous_tile_width_flag` を用いることができる。第1タイル又は直前のタイルは、高さが幅より大きい非正方形形状である場合、`use_previous_tile_height_flag` を用いることができる。

【0412】

信号で、画像に含まれるタイルの数を送信する場合、最後のタイルに対して、タイルのサイズに関わる情報の符号化を省略してもよい。

【0413】

表14は、最後のタイルに対して、タイルのサイズ情報の符号化を省略する例を示す。

10

20

30

40

50

【表 1 4】

記述子	
...	
tiles_enabled_flag	u (1)
...	
if (tiles_enabled_flag) {	10
number_of_tiles_in_picture_minus2	ue (v)
subtile_width_minus1	ue (v)
subtile_height_minus1	ue (v)
for (i=0 , i <	
number_of_tiles_in_picture_minus2+1 , i++) {	
if (i > 0)	20
use_previous_tile_size_flag	u (1)
if (use_previous_tile_size_flag == 0)	
{	
tile_width_minus1[i]	ue (v)
tile_height_minus1[i]	ue (v)
}	
}	30
loop_filter_across_tiles_enabled_flag	u (1)
}	
...	

【0 4 1 4】

最後のタイル以外のタイルのサイズを指定した場合、画像における余剰領域を最後のタイルとすることができます。

【0 4 1 5】

各符号化ツリーユニットに対して、符号化ツリーユニットが属するタイルを認識するための識別子（以下、タイル ID、Tile ID と呼ばれる）を割り当てることができる。

【0 4 1 6】

図 3 4 は、タイル ID を各符号化ツリーユニットに割り当てる例を示す図である。

【0 4 1 7】

同一のタイル ID を同一のタイルに属する符号化ツリーユニットに割り当てることができる。具体的には、N 番目の Tile ID をタイル N に属する符号化ツリーユニットに割り当てることができる。

【0418】

各符号化ツリーユニットに割り当てられるタイルIDを決定するために、画像における符号化ツリーユニットの位置を示す変数x及びyを決定することができる。ここで、xは、符号化ツリーユニットの左上サンプルの位置(x_0, y_0)のx軸座標を符号化ツリーユニットの幅で割ることで得られた値を表し、yは、符号化ツリーユニットの左上サンプルの位置(x_0, y_0)のy軸座標を符号化ツリーユニットの高さで割ることで得られた値を表す。具体的には、下記等式24及び25により、x及びyを導出することができる。

【数24】

【等式24】

10

$$x = (x_0 / (CTU \ width))$$

【数25】

【等式25】

20

$$y = (y_0 / (CTU \ height))$$

【0419】

以下説明されるプロセスにより、タイルIDを各符号化ツリーユニットに割り当てる操作を実行することができる。

【0420】

i) タイルIDの初期化

各符号化ツリーユニットのタイルIDは、画像におけるタイルの数から1を減算することで得られた値に初期化されてもよい。

30

40

50

【表 1 5】

tile_enabled_flag が 1 に等しいと、以下のように、変数 tile_id の値及び二次元アレイ TileId の値を指定する。

```

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

```

10

20

【0 4 2 1】

i i) タイル ID の導出

【表 1 6】

以下のように、変数 tile_id の値及び二次元アレイ CTU_tile に割り当てられた値を導出する。

```

for (ctu_y=0 , ctu_y <
     (tile_height_minus1 [i]+1) * (subtile_height_minus1+1) && 30
     (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++

```

40

40

【0 4 2 2】

上記実施例において、画像パラメータ集合を介して、信号で、タイルの境界でループフィルタを適用することを許容するかどうかを示すフラグを送信することを説明した。しかしながら、全てのタイルの境界でいずれもループフィルタを使用しないと設定すれば、主観画像品質の低下及び符号化効率の低下という問題を発生することがある。

【0 4 2 3】

50

従って、各タイルがループフィルタを適用することを許容するかどうかを示す情報を符号化し、信号で送信することができる。

【0424】

図35は、各タイルに対してループフィルタを適用するかどうかを選択的に決定する例を示す図である。

【0425】

図35に示す例において、各タイルに対して、水平又は垂直境界でループフィルタ（例えば、デブロッキングフィルタ、SAO及び/又はALF）の適用を許容するかどうかを決定することができる。

【0426】

表17は、各タイルに対して、ループフィルタの適用を許容するかどうかを示す情報を符号化する例を示す。

10

20

30

40

50

【表 1 7】

pic_parameter_set_rbsp() {	記述子
...	
tiles_enabled_flag	u (1)
...	
if(tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue (v)
subtile_width_minus1	ue (v)
subtile_height_minus1	ue (v)
for (i=0 , i < number_of_tiles_in_picture_minus2+1 , i++) {	
if (i > 0)	
use_previous_tile_size_flag	u (1)
if (use_previous_tile_size_flag == 0)	
{	
tile_width_minus1[i]	ue (v)
tile_height_minus1[i]	ue (v)
}	
loop_filter_across_tiles_flag[i]	u (1)
}	
}	
...	

【0 4 2 7】

表 1 7 の例において、シンタックス要素 `loop_filter_across_tile_flag[i]` は、ループフィルタを `i` 番目のタイルに適用することを許容するかどうかを示す。`loop_filter_across_tile_flag[i]` の値が 1 であると、タイル ID が `i` であるタイルの水平境界及び垂直境界でループフィルタを使用できることを示す。`loop_filter_across_tile_flag[i]` の値が 0 であると、タイル ID が `i` であるタイルの水平境界及び垂直境界でループフィルタを使用しないことを示す。

【0 4 2 8】

水平方向及び垂直方向の各方向でループフィルタの適用を許容するかどうかを示す情報を符号化することができる。

【0 4 2 9】

10

20

30

40

50

表18は、それぞれ、水平方向及び垂直方向で、ループフィルタの適用を許容するかどうかを示す情報を符号化する例を示す。

【表18】

記述子	
...	
tiles_enabled_flag	u(1)
...	
if(tiles_enabled_flag) {	
number_of_tiles_in_picture_minus2	ue(v)
subtile_width_minus1	ue(v)
subtile_height_minus1	ue(v)
for (i=0 , i < number_of_tiles_in_picture_minus2+1 , i++) {	
if (i > 0)	
use_previous_tile_size_flag	u(1)
if (use_previous_tile_size_flag == 0)	
{	
tile_width_minus1[i]	ue(v)
tile_height_minus1[i]	ue(v)
}	
loop_filter_hor_across_tiles_flag[i]	u(1)
loop_filter_ver_across_tiles_flag[i]	u(1)
}	
}	
...	

【0430】

表18の例において、シンタックス要素loop_filter_hor_across_tiles_flag[i]は、水平方向にi番目のタイルと交差する位置でループフィルタを適用することを許容するかどうかを示す。シンタックス要素loop_filter_ver_across_tiles_flag[i]は、垂直方向にi番目のタイルと交差する位置でループフィルタを適用することを許容するかどうかを示す。

【0431】

loop_filter_hor_across_tiles_flag[i]の値が1

10

20

30

40

50

であると、タイル ID が i であるタイルの水平境界でループフィルタを使用できることを示す。`loop_filter_hor_across_tile_flag[i]` の値が 0 であると、タイル ID が i であるタイルの垂直境界でループフィルタを使用しないことを示す。

【0432】

`loop_filter_ver_across_tile_flag[i]` の値が 1 であると、タイル ID が i であるタイルの垂直境界でループフィルタを使用できることを示す。`loop_filter_ver_across_tile_flag[i]` の値が 0 であると、タイル ID が i であるタイルの垂直境界でループフィルタを使用しないことを示す。

10

【0433】

複数のタイルを含むタイル群がループフィルタの適用を許容するかどうかを示す情報を符号化し、信号で送信することができる。該情報に基づいて、タイル群に含まれる複数のタイルがループフィルタの適用を許容するかどうかを決定することができる。

【0434】

タイル群を決定するために、ビットストリームを介して、信号で、前記タイル群に属するタイルの数、タイル群のサイズ及び画像の分割情報のうちの少なくとも 1 つを送信することができる。あるいは、エンコーダ及びデコーダにおいて事前定義されたサイズの領域をタイル群とすることができます。

20

【0435】

ループフィルタの適用を許容するかどうかを示す情報の符号化を省略してもよく、また、タイルに含まれる符号化ツリーユニットの数、タイルの幅及びタイルの高さのうちの少なくとも 1 つに基づいて、ループフィルタの適用を許容するかどうかを決定することができる。例えば、タイルの幅が参照値未満である場合、水平方向にループフィルタを適用することを許容し、タイルの高さが参照値未満である場合、垂直方向にループフィルタを適用することができる。

【0436】

タイルの境界でループフィルタを用いる場合、タイルに含まれるデータに基づいて、タイル外部の再構成データを生成することができる。この場合、タイルに含まれるデータに対してパディング又は補間を行うことで、タイル外部の再構成ビデオを得ることができる。続いて、タイル外部の再構成データを用いることで、ループフィルタを適用することができる。

30

【0437】

復号化プロセス又は符号化プロセスに重点を置いて説明する実施例を復号化プロセス又は符号化プロセスに用いられたものも本発明の範囲内に含まれる。説明された順番と異なる順番で、所定の順番で説明された複数の実施例を変更したものも本発明の範囲内に含まれる。

【0438】

一連のステップ又はフローチャートに基づいて実施例を説明したが、これは、発明の時間的順番を限定しない。また、必要に応じて、同時に実行してもよいか又は他の順番に応じて実行してもよい。なお、上記実施例において、ブロック図を構成する構成要素（例えば、ユニット、モジュール等）は、それぞれ、ハードウェア機器又はソフトウェアとして実現されてもよい。また、複数の構成要素を組み合わせて单一のハードウェア機器又はソフトウェアとして実行されてもよい。前記実施例は、プログラム命令の形態で実行してもよい。前記プログラム命令は、種々のコンピュータ部材により実行されてコンピュータ可読記憶媒体に記録されてもよい。前記コンピュータ可読記憶媒体は、プログラム命令、データファイル、データ構造などを単独で含んでもよいか又はその組み合わせを含んでもよい。コンピュータ可読記憶媒体の例として、ハードディスク、フレキシブルディスク及び磁気テープなどの磁気媒体、CD-ROM、DVD 等の光学記録媒体、フロップティーカールディスク (floptical disk) のような磁気・光学媒体 (magnet

40

50

o - o p t i c a l m e d i a) 及び R O M 、 R A M 、 フラッシュメモリ等のような、プログラム命令を記憶して該命令を実行する方式で特に配置されたハードウェア機器を含んでもよい。前記ハードウェア機器は、1つ又は複数のソフトウェアモジュールとして操作を行い、本発明による処理を実行するように構成されてもよく、その逆も同様である。

【産業上の利用可能性】

【 0 4 3 9 】

本発明は、ビデオに対して符号化 / 復号化を行う電子機器に適用可能である。

10

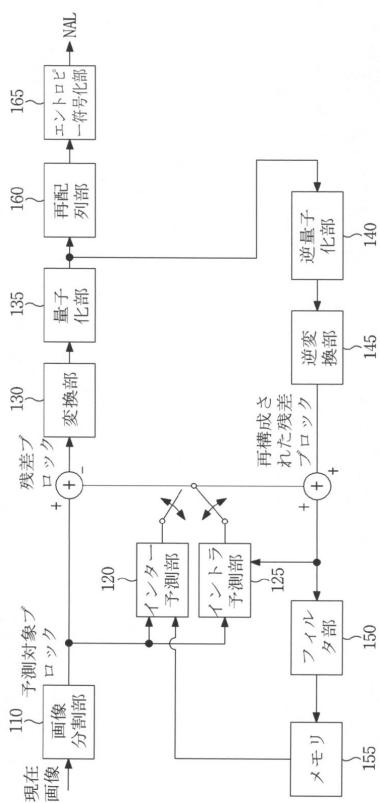
20

30

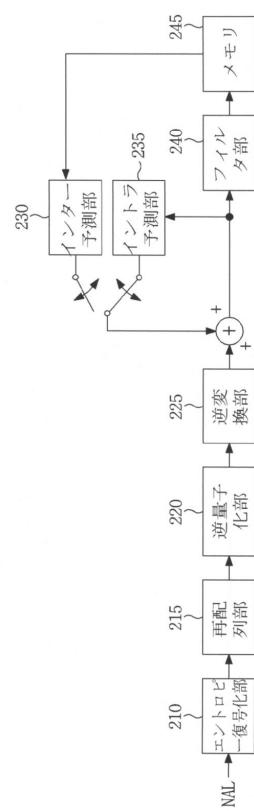
40

50

【図面】
【図 1】

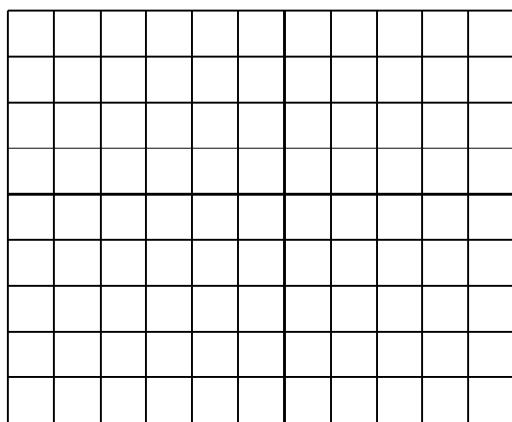


【 四 2 】



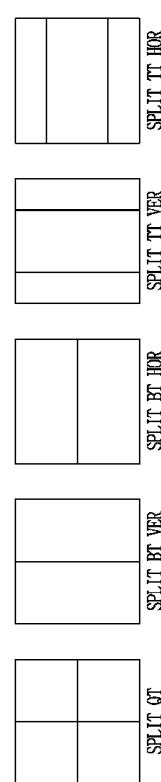
【図3】

[五三]



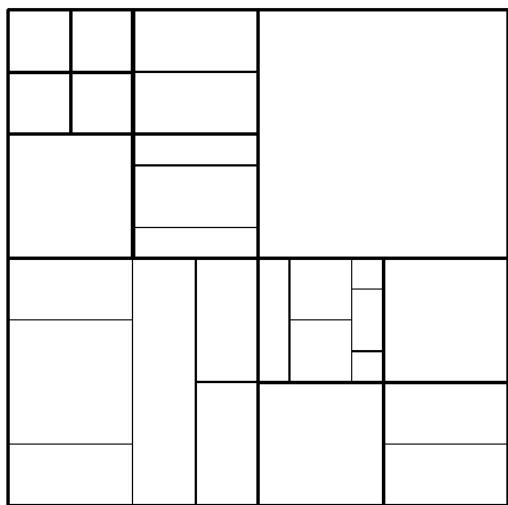
【図4】

[도4]

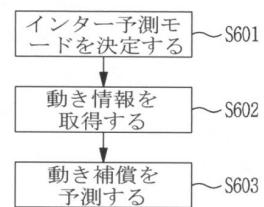


【図 5】

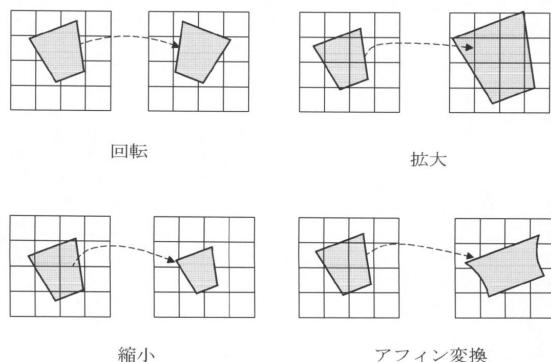
[図5]



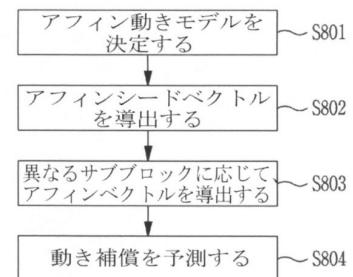
【図 6】



【図 7】



【図 8】

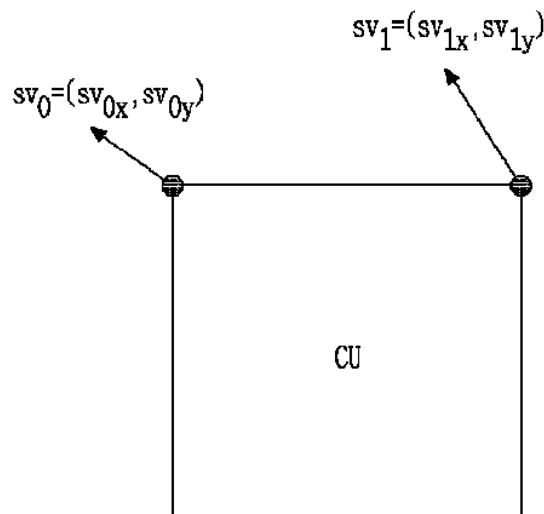


30

40

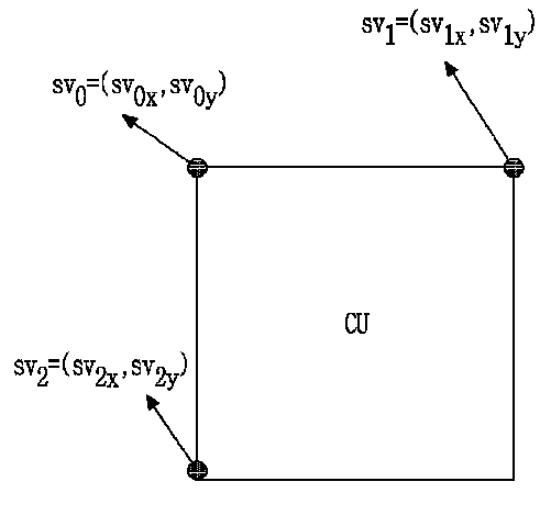
50

【図 9 (a)】



(a)

【図 9 (b)】

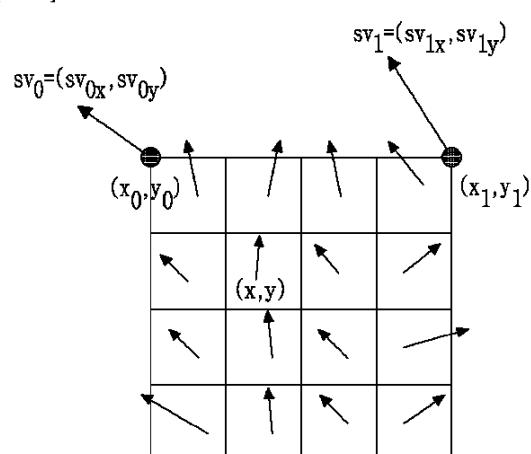


(b)

10

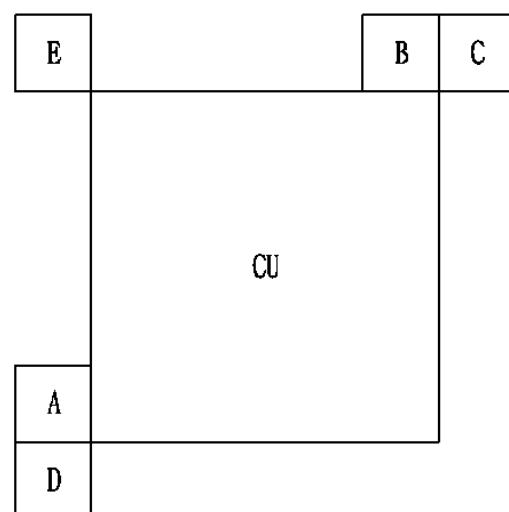
【図 10】

[도10]



【図 11】

[도11]



20

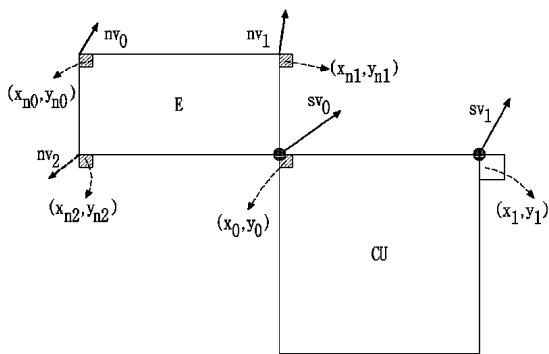
30

40

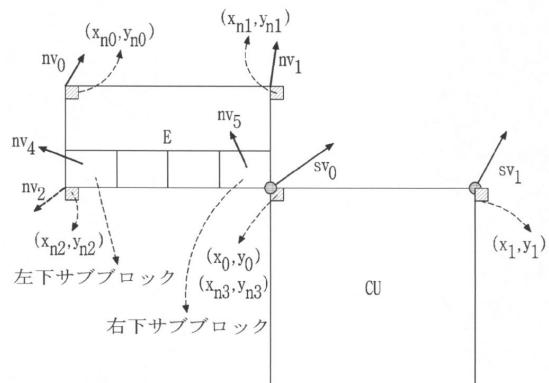
50

【図 1 2】

[E12]

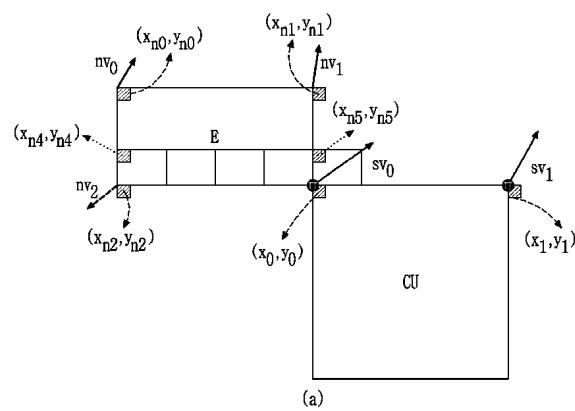


【図 1 3】

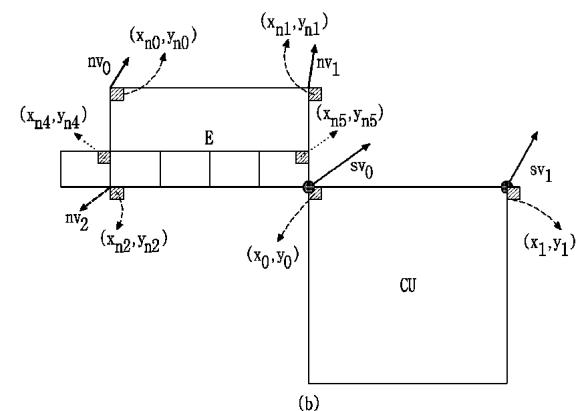


10

【図 1 4 (a)】



【図 1 4 (b)】



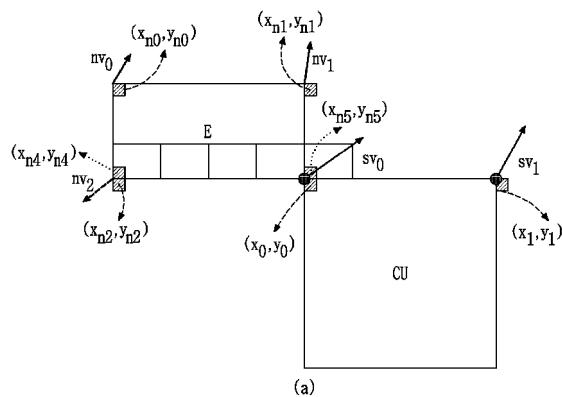
20

30

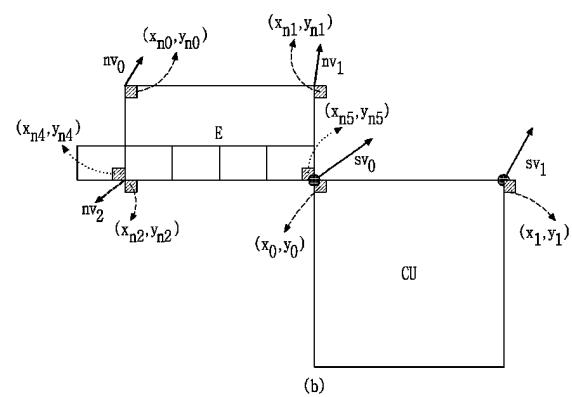
40

50

【図 1 5 (a)】

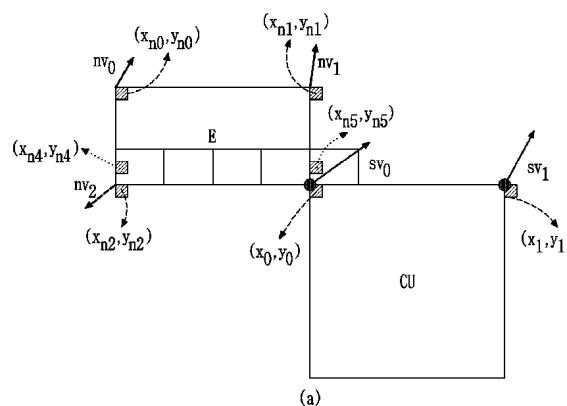


【図 1 5 (b)】

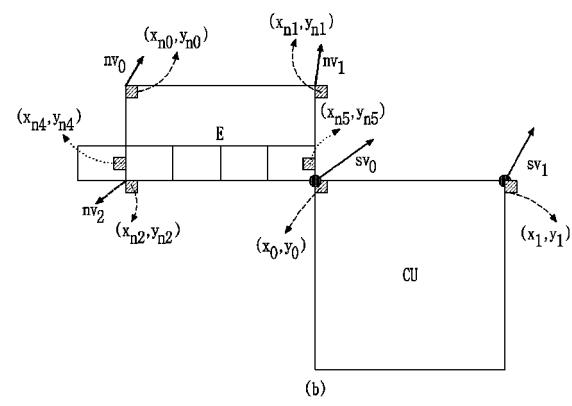


10

【図 1 6 (a)】



【図 1 6 (b)】



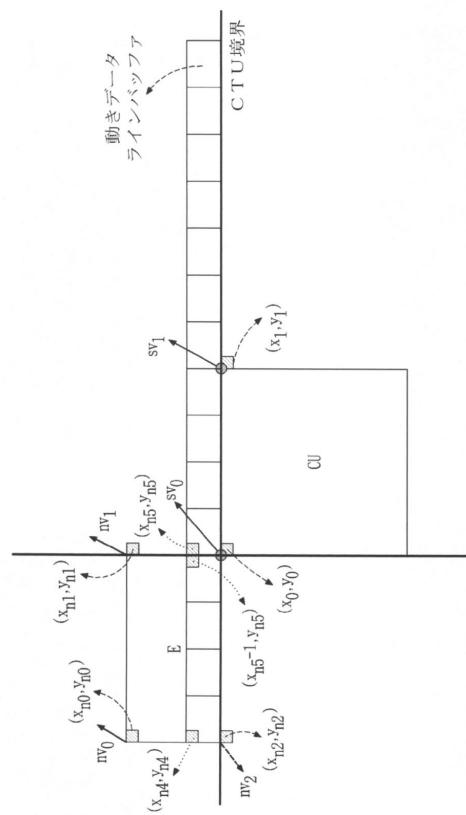
20

30

40

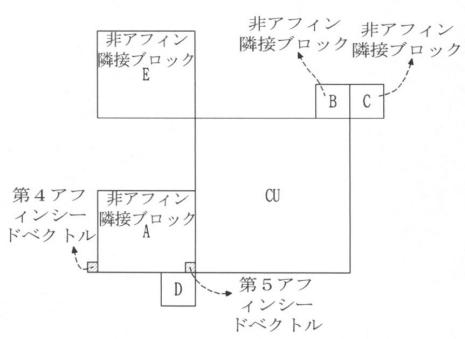
50

【図 17】



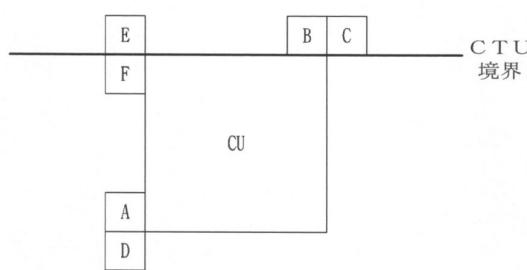
10

【図 18】



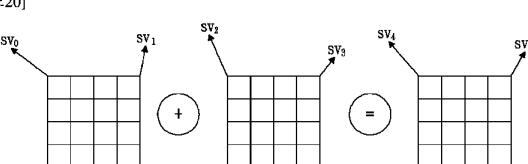
20

【図 19】



30

【図 20】

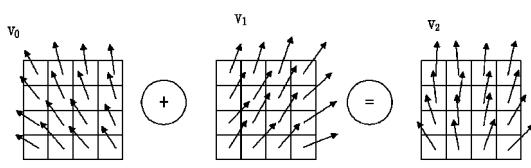


40

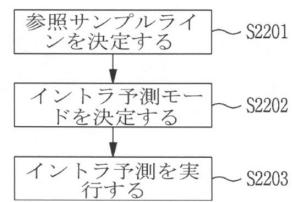
50

【図 2 1】

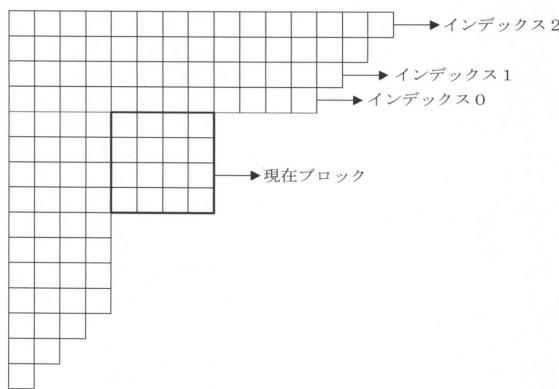
[E21]



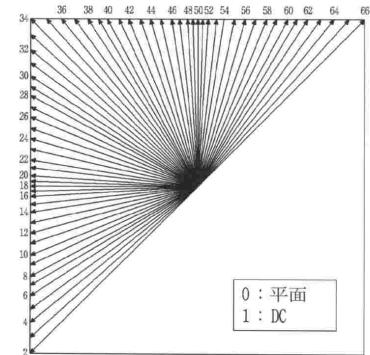
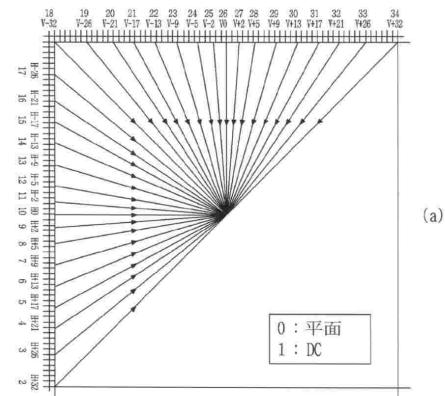
【図 2 2】



【図 2 3】



【図 2 4】



10

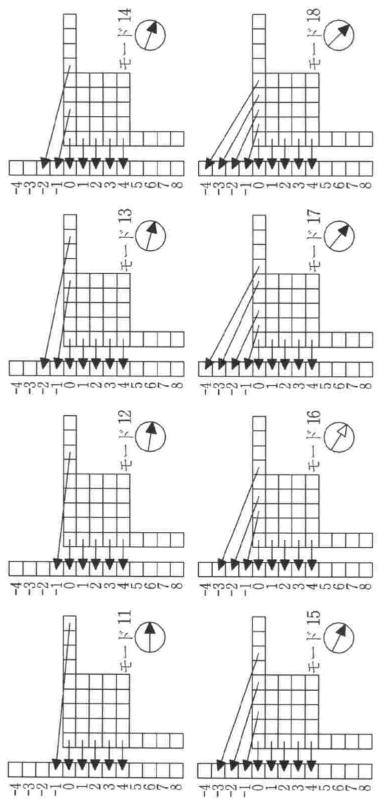
20

30

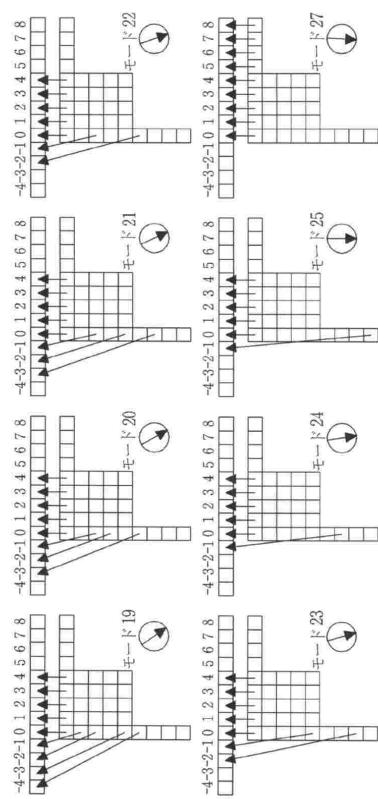
40

50

【図 2 5】



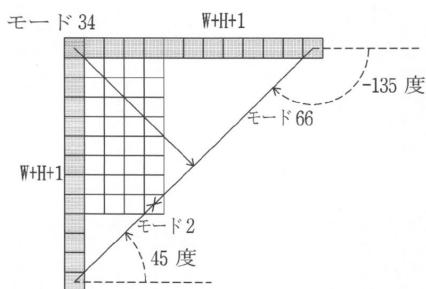
【図 2 6】



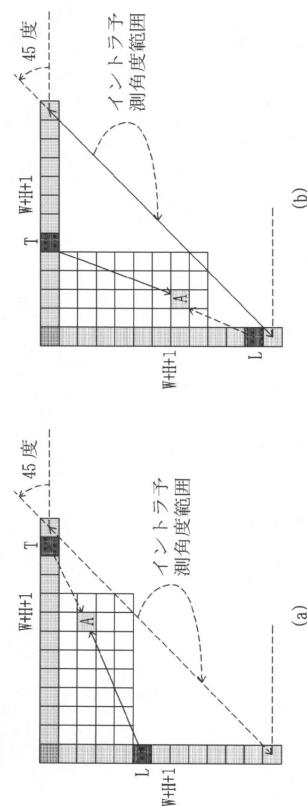
10

20

【図 2 7】



【図 2 8】

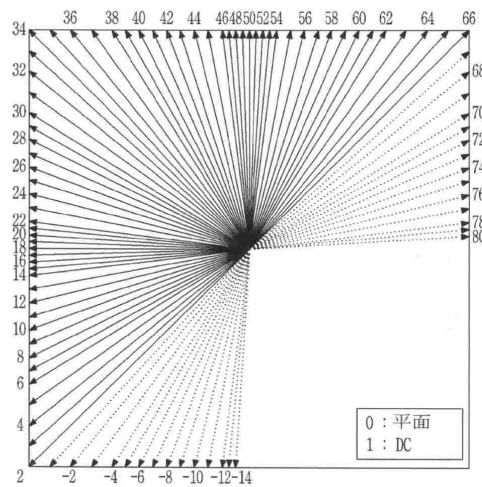


30

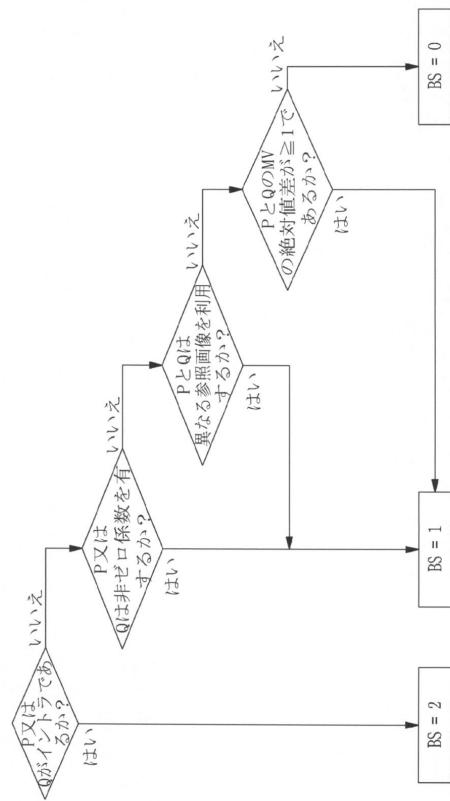
40

50

【図 29】



【図 30】



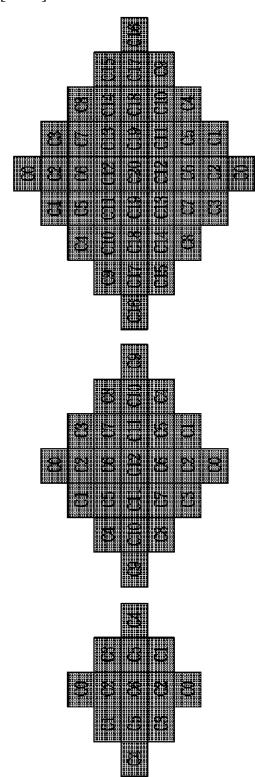
10

20

30

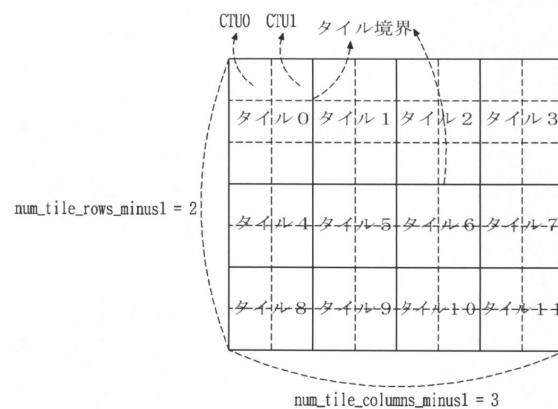
40

【図 31】



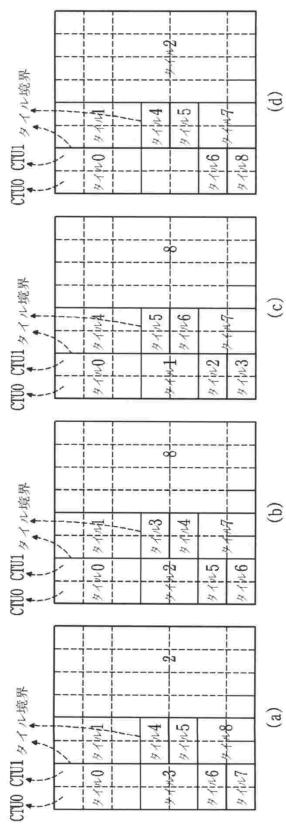
[図31]

【図 32】

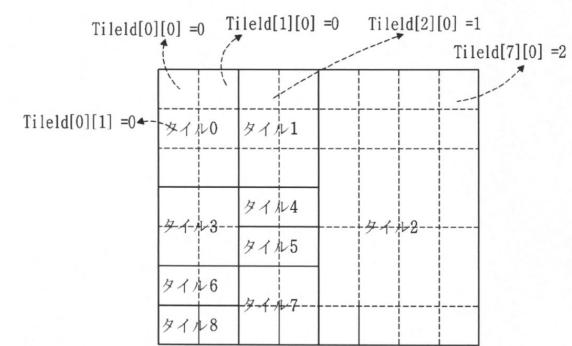


50

【図 3 3】



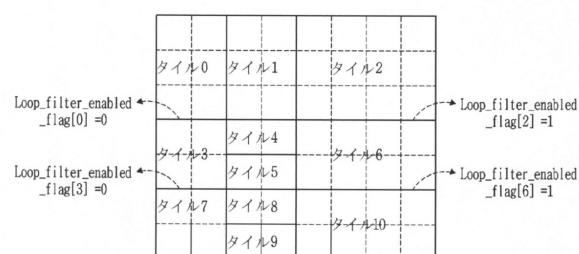
【図 3 4】



10

20

【図 3 5】



30

40

50

フロントページの続き

(33) 優先権主張国・地域又は機関

韓国(KR)

(31) 優先権主張番号 10-2018-0114344

(32) 優先日 平成30年9月21日(2018.9.21)

(33) 優先権主張国・地域又は機関

韓国(KR)

(74) 代理人 100107582

弁理士 關根 賀

(74) 代理人 100152205

弁理士 吉田 昌司

(74) 代理人 100137523

弁理士 出口 智也

(74) 代理人 100120385

弁理士 鈴木 健之

(72) 発明者 リー、ベグン

大韓民国キヨンギ - ド、ソンナム - シ、ブンダン - グ、パンギヨ - ロ、393、214 - 1202

審査官 田中 純一

(56) 参考文献 米国特許出願公開第2018/0098063(US, A1)

特表2019-535192(JP, A)

国際公開第2018/061563(WO, A1)

国際公開第2017/130696(WO, A1)

米国特許出願公開第2019/0037231(US, A1)

(58) 調査した分野 (Int.Cl., DB名)

H04N 7/12

H04N 19/00 - 19/98