

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6817425号
(P6817425)

(45) 発行日 令和3年1月20日(2021.1.20)

(24) 登録日 令和2年12月28日(2020.12.28)

(51) Int.Cl.	F 1
HO4N 19/52 (2014.01)	HO4N 19/52
HO4N 19/105 (2014.01)	HO4N 19/105
HO4N 19/139 (2014.01)	HO4N 19/139
HO4N 19/176 (2014.01)	HO4N 19/176

請求項の数 36 (全 68 頁)

(21) 出願番号	特願2019-517341 (P2019-517341)	(73) 特許権者	507364838 クアルコム、インコーポレイテッド
(86) (22) 出願日	平成29年10月5日(2017.10.5)		アメリカ合衆国 カリフォルニア 921 21 サンディエゴ モアハウス ドラ イブ 5775
(65) 公表番号	特表2019-535192 (P2019-535192A)	(74) 代理人	100108453 弁理士 村山 靖彦
(43) 公表日	令和1年12月5日(2019.12.5)	(74) 代理人	100163522 弁理士 黒田 晋平
(86) 國際出願番号	PCT/US2017/055350	(72) 発明者	イーウェン・チェン アメリカ合衆国・カリフォルニア・921 21-1714・サン・ディエゴ・モアハ ウス・ドライブ・5775
(87) 國際公開番号	W02018/067823		
(87) 國際公開日	平成30年4月12日(2018.4.12)		
審査請求日	令和1年12月11日(2019.12.11)		
(31) 優先権主張番号	62/404,719		
(32) 優先日	平成28年10月5日(2016.10.5)		
(33) 優先権主張国・地域又は機関	米国(US)		
(31) 優先権主張番号	15/725,052		
(32) 優先日	平成29年10月4日(2017.10.4)		
(33) 優先権主張国・地域又は機関	米国(US)		

最終頁に続く

(54) 【発明の名称】ビデオコーディングにおけるアフィン動きモデルのための動きベクトル予測

(57) 【特許請求の範囲】

【請求項1】

ビデオデータを復号する方法であって、ソースアフィンブロックを選択するステップであって、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択するステップが、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると、決定するステップを備える、ステップと。

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿して、前記現在のブロックの複数の制御点の動きベクトル予測子を決定するステップであって、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用される、ステップと。

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィン動きベクトル予測子(MVP)セットを、アフィンMVPセット候補リストに挿入するステップと、

ビットストリームにおいてシグナリングされるインデックスに基づいて、前記アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定するステップと、

前記ピットストリームから、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)を取得するステップと、

前記選択されたアフィンMVPセットおよび前記MVDに含まれる前記動きベクトル予測子に基づいて、前記現在のブロックの前記制御点の動きベクトルを決定するステップと、

前記現在のブロックの前記制御点の前記動きベクトルに基づいて、予測ブロックを生成するステップと、

残差データおよび前記予測ブロックに基づいて前記現在のブロックを再構築するステップとを備える、方法。

【請求項 2】

10

前記現在のブロックの前記制御点が、前記現在のブロックの第1の制御点および前記現在のブロックの第2の制御点を含み、前記方法が、

前記現在のブロックの前記第1の制御点の隣のブロックの動きベクトルとして、第1の動きベクトル予測子を決定するステップと、

前記現在のブロックの前記第2の制御点の隣のブロックの動きベクトルとして、第2の動きベクトル予測子を決定するステップと、

前記アフィンMVPセット候補リストに、前記第1の動きベクトル予測子および前記第2の動きベクトル予測子を含むアフィンMVPセットを挿入するステップとをさらに備える、請求項1に記載の方法。

【請求項 3】

20

前記現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択するステップが、

前記ソースアフィンブロックが、事前に定義された訪問順序で訪問される前記複数の隣接ブロックの最初に生じるアフィンコーディングされたブロックであると、決定するステップを備える、請求項1に記載の方法。

【請求項 4】

アフィンコーディングされたブロックが前記事前に定義された優先度セットのうちの1つの中にはない場合、前記アフィンコーディングされたブロックが利用可能であると見なされない、請求項1に記載の方法。

【請求項 5】

30

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各自的の隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと同じであるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1であり、

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと同じであるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項4に記載の方法。

【請求項 6】

40

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各自的の隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと異なるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1であり、

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと

50

異なるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項4に記載の方法。

【請求項7】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックがアフィンインターモードでコーディングされる場合、
前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、

10

前記それぞれの隣接ブロックがアフィンマージモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にある

ように定義される、請求項4に記載の方法。

【請求項8】

アフィンソースブロックが第1のアフィンソースブロックであり、前記方法が、

第2のソースアフィンブロックを選択するステップであって、前記第2のソースアフィンブロックが、前記現在のブロックに空間的に隣接する異なるアフィンコーディングされたブロックである、ステップと、

前記第2のソースアフィンブロックの制御点の動きベクトルを外挿して、前記現在のブロックの前記制御点の第2の動きベクトル予測子を決定するステップと、

20

前記現在のブロックの前記制御点の前記第2の動きベクトル予測子を含む第2のアフィンMVPセットを、前記アフィンMVPセット候補リストに挿入するステップとをさらに備える、請求項1に記載の方法。

【請求項9】

ビデオデータを符号化する方法であって、

ソースアフィンブロックを選択するステップであって、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、前記現在のブロックが複数の隣接ブロックの間にあり、前記ソースアフィンブロックを選択するステップが、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると、決定するステップを備える、ステップと、

30

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿して、前記現在のブロックの複数の制御点の動きベクトル予測子を決定するステップであって、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用される、ステップと、

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィンMVPセットを、アフィン動きベクトル予測子(MVP)セット候補リストに挿入するステップと、

40

前記アフィンMVPセット候補リストの中のアフィンMVPセットを選択するステップと、

ビットストリームにおいて、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)をシグナリングするステップと、

前記ビットストリームにおいて、前記選択されたアフィンMVPセットの前記アフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングするステップとを備える、方法。

【請求項10】

前記現在のブロックの前記制御点が、前記現在のブロックの第1の制御点および前記現在のブロックの第2の制御点を含み、前記方法が、

50

前記現在のブロックの前記第1の制御点の隣のブロックの動きベクトルとして、第1の動きベクトル予測子を決定するステップと、

前記現在のブロックの前記第2の制御点の隣のブロックの動きベクトルとして、第2の動きベクトル予測子を決定するステップと、

前記アフィンMVPセット候補リストに、前記第1の動きベクトル予測子および前記第2の動きベクトル予測子を含むアフィンMVPセットを挿入するステップとをさらに備える、請求項9に記載の方法。

【請求項11】

前記現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択するステップが、

前記ソースアフィンブロックが、事前に定義された訪問順序で訪問される前記複数の隣接ブロックの最初に生じるアフィンコーディングされたブロックであると、決定するステップを備える、請求項9に記載の方法。

【請求項12】

アフィンコーディングされたブロックが前記事前に定義された優先度セットのうちの1つの中にはない場合、前記アフィンコーディングされたブロックが利用可能であると見なされない、請求項9に記載の方法。

【請求項13】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと同じであるリストX参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1に等しく、

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと同じであるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項12に記載の方法。

【請求項14】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと異なるリストX参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1に等しく、

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと異なるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項12に記載の方法。

【請求項15】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックがアフィンインターモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、

前記それぞれの隣接ブロックがアフィンマージモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にある

10

20

30

40

50

ように定義される、請求項12に記載の方法。

【請求項16】

アフィンソースブロックが第1のアフィンソースブロックであり、前記方法が、

第2のソースアフィンブロックを選択するステップであって、前記第2のソースアフィンブロックが、前記現在のブロックに空間的に隣接する異なるアフィンコーディングされたブロックである、ステップと、

前記第2のソースアフィンブロックの制御点の動きベクトルを外挿して、前記現在のブロックの前記制御点の第2の動きベクトル予測子を決定するステップと、

前記現在のブロックの前記制御点の前記第2の動きベクトル予測子を含む第2のアフィンMVPセットを、前記アフィンMVPセット候補リストに挿入するステップとをさらに備える、10請求項9に記載の方法。

【請求項17】

ビデオデータを復号するためのデバイスであって、

前記ビデオデータを記憶するように構成されるメモリと、

1つまたは複数の処理回路とを備え、前記1つまたは複数の処理回路が、

ソースアフィンブロックを選択し、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、前記現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択することの一部として、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると決定し、20

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿して、前記現在のブロックの複数の制御点の動きベクトル予測子を決定し、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用され、

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィン動きベクトル予測子(MVP)セットを、アフィンMVPセット候補リストに挿入し、

ビットストリームにおいてシグナリングされるインデックスに基づいて、前記アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定し、30

前記ビットストリームから、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)を取得し、

前記選択されたアフィンMVPセットおよび前記MVDに含まれる前記動きベクトル予測子に基づいて、前記現在のブロックの前記制御点の動きベクトルを決定し、

前記現在のブロックの前記制御点の前記動きベクトルに基づいて、予測ブロックを生成し、

残差データおよび前記予測ブロックに基づいて前記現在のブロックを再構築するよう構成される、デバイス。40

【請求項18】

前記現在のブロックの前記制御点が、前記現在のブロックの第1の制御点および前記現在のブロックの第2の制御点を含み、前記1つまたは複数の処理回路がさらに、

前記現在のブロックの前記第1の制御点の隣のブロックの動きベクトルとして、第1の動きベクトル予測子を決定し、

前記現在のブロックの前記第2の制御点の隣のブロックの動きベクトルとして、第2の動きベクトル予測子を決定し、

前記アフィンMVPセット候補リストに、前記第1の動きベクトル予測子および前記第2の動きベクトル予測子を含むアフィンMVPセットを挿入するように構成される、請求項17に記載のデバイス。50

【請求項 19】

前記現在のブロックが複数の隣接ブロックの隣にあり、前記1つまたは複数の処理回路が、前記ソースアフィンブロックを選択することの一部として、

前記ソースアフィンブロックが、事前に定義された訪問順序で訪問される前記複数の隣接ブロックの最初に生じるアフィンコーディングされたブロックであると、前記1つまたは複数の処理回路が決定する

ように構成される、請求項17に記載のデバイス。

【請求項 20】

アフィンコーディングされたブロックが前記事前に定義された優先度セットのうちの1つの中にはない場合、前記アフィンコーディングされたブロックが利用可能であると見なされない、請求項17に記載のデバイス。10

【請求項 21】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと同じであるリストX参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1に等しく、20

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと同じであるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項20に記載のデバイス。

【請求項 22】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと異なるリストX参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1に等しく、30

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと異なるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項20に記載のデバイス。

【請求項 23】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、40

前記それぞれの隣接ブロックがアフィンインターモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、

前記それぞれの隣接ブロックがアフィンマージモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にある

ように定義される、請求項20に記載のデバイス。

【請求項 24】

アフィンソースブロックが第1のアフィンソースブロックであり、前記1つまたは複数の処理回路がさらに、

第2のソースアフィンブロックを選択し、前記第2のソースアフィンブロックが、前記現在のブロックに空間的に隣接する異なるアフィンコーディングされたブロックであり、50

前記第2のソースアフィンブロックの制御点の動きベクトルを外挿して、前記現在のブロックの前記制御点の第2の動きベクトル予測子を決定し、

前記現在のブロックの前記制御点の前記第2の動きベクトル予測子を含む第2のアフィンMVPセットを、前記アフィンMVPセット候補リストに挿入するように構成される、請求項17に記載のデバイス。

【請求項 2 5】

ビデオデータを符号化するためのデバイスであって、

前記ビデオデータを記憶するように構成されるメモリと、

1つまたは複数の処理回路とを備え、前記1つまたは複数の処理回路が、

ソースアフィンブロックを選択し、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、前記現在のブロックが複数の隣接ブロックの間にあり、前記ソースアフィンブロックを選択することの一部として、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると決定し、

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿して、前記現在のブロックの複数の制御点の動きベクトル予測子を決定し、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用され、

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィン動きベクトル予測子(MVP)セットを、アフィンMVPセット候補リストに挿入し、

前記アフィンMVPセット候補リストの中のアフィンMVPセットを選択し、

ビットストリームにおいて、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)をシグナリングし、

前記ビットストリームにおいて、前記選択されたアフィンMVPセットの前記アフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングする

ように構成される、デバイス。

【請求項 2 6】

前記現在のブロックの前記制御点が、前記現在のブロックの第1の制御点および前記現在のブロックの第2の制御点を含み、前記1つまたは複数の処理回路がさらに、

前記現在のブロックの前記第1の制御点の隣のブロックの動きベクトルとして、第1の動きベクトル予測子を決定し、

前記現在のブロックの前記第2の制御点の隣のブロックの動きベクトルとして、第2の動きベクトル予測子を決定し、

前記アフィンMVPセット候補リストに、前記第1の動きベクトル予測子および前記第2の動きベクトル予測子を含むアフィンMVPセットを挿入するように構成される、請求項25に記載のデバイス。

【請求項 2 7】

前記現在のブロックが複数の隣接ブロックの間にあり、前記1つまたは複数の処理回路が、前記ソースアフィンブロックを選択することの一部として、

前記ソースアフィンブロックが、事前に定義された訪問順序で訪問される前記複数の隣接ブロックの最初に生じるアフィンコーディングされたブロックであると、前記1つまたは複数の処理回路が決定する

ように構成される、請求項25に記載のデバイス。

【請求項 2 8】

アフィンコーディングされたブロックが前記事前に定義された優先度セットのうちの1つの中にはない場合、前記アフィンコーディングされたブロックが利用可能であると見なされ

10

20

30

40

50

ない、請求項25に記載のデバイス。

【請求項 29】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと同じであるリストX参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1であり、

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと同じであるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項28に記載のデバイス。

【請求項 30】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックが、前記現在のブロックのリストX参照ピクチャと異なるリストX参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、Xが0または1であり、

前記それぞれの隣接ブロックが、前記現在のブロックの前記リストX参照ピクチャと異なるリストY参照ピクチャを有する場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にあり、Yが1-Xに等しい

ように定義される、請求項28に記載のデバイス。

【請求項 31】

前記複数の事前に定義された優先度セットが第1の優先度セットおよび第2の優先度セットを含み、

前記第1の優先度セットおよび前記第2の優先度セットが、前記複数の隣接ブロックの各々のそれぞれの隣接ブロックに対して、

前記それぞれの隣接ブロックがアフィンインターモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第1の優先度セットの中にあり、

前記それぞれの隣接ブロックがアフィンマージモードでコーディングされる場合、前記それぞれの隣接ブロックが前記第2の優先度セットの中にある

ように定義される、請求項28に記載のデバイス。

【請求項 32】

アフィンソースブロックが第1のアフィンソースブロックであり、前記1つまたは複数の処理回路がさらに、

第2のソースアフィンブロックを選択し、前記第2のソースアフィンブロックが、前記現在のブロックに空間的に隣接する異なるアフィンコーディングされたブロックであり、

前記第2のソースアフィンブロックの制御点の動きベクトルを外挿して、前記現在のブロックの前記制御点の第2の動きベクトル予測子を決定し、

前記現在のブロックの前記制御点の前記第2の動きベクトル予測子を含む第2のアフィンMVPセットを、前記アフィンMVPセット候補リストに挿入するように構成される、請求項25に記載のデバイス。

【請求項 33】

ビデオデータを復号するためのデバイスであって、

ソースアフィンブロックを選択するための手段であって、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択する

10

20

30

40

50

ための手段が、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると、決定するための手段を備える、手段と、

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿して、前記現在のブロックの複数の制御点の動きベクトル予測子を決定するための手段であって、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用される、手段と、10

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィン動きベクトル予測子(MVP)セットを、アフィンMVPセット候補リストに挿入するための手段と、

ビットストリームにおいてシグナリングされるインデックスに基づいて、前記アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定するための手段と、

前記ビットストリームから、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)を取得するための手段と、

前記選択されたアフィンMVPセットおよび前記MVDに含まれる前記動きベクトル予測子に基づいて、前記現在のブロックの前記制御点の動きベクトルを決定するための手段と、

前記現在のブロックの前記制御点の前記動きベクトルに基づいて、予測ブロックを生成するための手段と、20

残差データおよび前記予測ブロックに基づいて前記現在のブロックを再構築するための手段とを備える、デバイス。

【請求項 3 4】

ビデオデータを符号化するためのデバイスであって、

ソースアフィンブロックを選択するための手段であって、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、前記現在のブロックが複数の隣接ブロックの間にあり、前記ソースアフィンブロックを選択するための手段が、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると、決定するための手段を備える、手段と、30

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿して、前記現在のブロックの複数の制御点の動きベクトル予測子を決定するための手段であって、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用される、手段と、

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィンMVPセットを、アフィン動きベクトル予測子(MVP)セット候補リストに挿入するための手段と、40

前記アフィンMVPセット候補リストの中のアフィンMVPセットを選択するための手段と、

ビットストリームにおいて、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)をシグナリングするための手段と、

前記ビットストリームにおいて、前記選択されたアフィンMVPセットの前記アフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングするための手段とを備える、デバイス。

【請求項 3 5】

実行されると、ビデオ復号のためのデバイスの1つまたは複数の処理回路に、

ソースアフィンブロックを選択させ、前記ソースアフィンブロックが、現在のブロック50

に空間的に隣接するアフィンコーディングされたブロックであり、現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択させることの一部として、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると決定させ、

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿させて、前記現在のブロックの複数の制御点の動きベクトル予測子を決定させ、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用され、10

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィン動きベクトル予測子(MVP)セットを、アフィンMVPセット候補リストへ挿入させ、

ビットストリームにおいてシグナリングされるインデックスに基づいて、前記アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定させ、

前記ビットストリームから、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)を取得させ、

前記選択されたアフィンMVPセットおよび前記MVDに含まれる前記動きベクトル予測子に基づいて、前記現在のブロックの前記制御点の動きベクトルを決定させ、20

前記現在のブロックの前記制御点の前記動きベクトルに基づいて、予測ブロックを生成させ、

残差データおよび前記予測ブロックに基づいて前記現在のブロックを再構築させる、命令を記憶したコンピュータ可読記憶媒体。

【請求項 3 6】

実行されると、ビデオデータを符号化するためのデバイスの1つまたは複数の処理回路に、

ソースアフィンブロックを選択させ、前記ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、現在のブロックが複数の隣接ブロックの隣にあり、前記ソースアフィンブロックを選択させることの一部として、前記ソースアフィンブロックが、複数の事前に定義された優先度セット内のブロックのうち、事前に定義された訪問順序に従って前記複数の隣接ブロックの最初に生じる利用可能なアフィンコーディングされたブロックであると決定させ、30

前記ソースアフィンブロックの複数の制御点の動きベクトルを外挿させて、前記現在のブロックの複数の制御点の動きベクトル予測子を決定させ、前記ソースアフィンブロックの前記複数の制御点が、前記ソースアフィンブロックの左上の第1の制御点と、前記ソースアフィンブロックの右上の第2の制御点とを含み、前記ソースアフィンブロックの前記第1および第2の制御点の前記動きベクトルが、ソースアフィンブロックのアフィン予測のために使用され、

前記現在のブロックの前記制御点の前記動きベクトル予測子を含むアフィン動きベクトル予測子(MVP)セットを、アフィンMVPセット候補リストへ挿入させ、40

前記アフィンMVPセット候補リストの中のアフィンMVPセットを選択させ、

ビットストリームにおいて、前記現在のブロックの前記制御点の動きベクトルと前記選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示す動きベクトル差分(MVD)をシグナリングさせ、

前記ビットストリームにおいて、前記選択されたアフィンMVPセットの前記アフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングさせる、命令を記憶したコンピュータ可読記憶媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本出願は、その内容全体が参考として本明細書に組み込まれる、2016年10月5日に出願された米国仮特許出願第62/404,719号の利益を主張する。

【0002】

本開示は、ビデオコーディングのためのデバイス、システム、および方法に関する。

【背景技術】**【0003】**

デジタルビデオ能力は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップまたはデスクトップコンピュータ、タブレットコンピュータ、電子ブックリーダー、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレーヤ、ビデオゲーミングデバイス、ビデオゲームコンソール、セルラーまたは衛星無線電話、いわゆる「スマートフォン」、ビデオ遠隔会議デバイス、ビデオストリーミングデバイスなどを含む、広範囲のデバイスに組み込まれ得る。デジタルビデオデバイスは、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、Part 10、Advanced Video Coding (AVC)、ITU-T H.265、High Efficiency Video Coding (HEVC)規格によって定義された規格、およびそのような規格の拡張に記載されているビデオ圧縮技法などのビデオ圧縮技法を実施する。ビデオデバイスは、そのようなビデオ圧縮技法を実施することによって、デジタルビデオ情報をより効率的に送信し、受信し、符号化し、復号し、かつ/または記憶し得る。10

【0004】

ビデオ圧縮技法は、ビデオシーケンスに固有の冗長性を低減または除去するために、空間(イントラピクチャ)予測および/または時間(インターピクチャ)予測を実行する。ブロックベースのビデオコーディングの場合、ビデオスライス(すなわち、ビデオフレームまたはビデオフレームの一部分)がビデオブロックに区分されることがあり、ビデオブロックは、ツリーブロック、コーディング単位(CU)、および/またはコーディングノードと呼ばれることもある。ピクチャのイントラコーディングされた(I)スライスの中のビデオブロックは、同じピクチャの中の隣接ブロックの中の参照サンプルに対する空間的予測を使用して符号化される。ピクチャのインターフォーマットコーディングされた(PまたはB)スライスの中のビデオブロックは、同じピクチャの中の隣接ブロックの中の参照サンプルに対する空間的予測または他の参照ピクチャの中の参照サンプルに対する時間的予測を使用し得る。空間的予測または時間的予測は、コーディングされるべきブロックの予測ブロックをもたらす。残差データは、コーディングされるべき元のブロックと予測ブロックとの間の画素差分を表す。インターフォーマットコーディングされたブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトル、およびコーディングされたブロックと予測ブロックとの間の差分を示す残差データに従って符号化される。イントラコーディングされたブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データは、画素領域から変換領域に変換されて残差変換係数をもたらすことがあり、残差変換係数は次いで、量子化されることがある。30

【発明の概要】**【課題を解決するための手段】****【0005】**

全般に、本開示は、アフィン動き予測モードのための動きベクトル予測および動きベクトル再構築に関する技法を説明する。本技法は、HEVC(High Efficiency Video Coding)などの既存のビデオコーデックのいずれにも適用されることが可能であり、またはあらゆる未来のビデオコーディング規格における効率的なコーディングツールになり得る。

【0006】

一例では、本開示は、ビデオデータを復号する方法を説明し、この方法は、ソースアフィンブロックを選択するステップであって、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックである、ステップと、ソースアフィンブロックの制御点の動きベクトルを外挿して現在のブロックの制御点の動きベクト50

ル予測子を決定するステップと、アフィン動きベクトル予測子(MVP)セット候補リストに、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入するステップと、ビットストリームにおいてシグナリングされるインデックスに基づいて、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定するステップと、ビットストリームから、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)を取得するステップと、選択されたアフィンMVPセットに含まれる動きベクトル予測子およびMVDに基づいて、現在のブロックの制御点の動きベクトルを決定するステップと、現在のブロックの制御点の動きベクトルに基づいて、予測ブロックを生成するステップと、残差データおよび予測ブロックに基づいて現在のブロックを再構築するステップとを備える。

10

【0007】

別の例では、本開示は、ビデオデータを符号化する方法を説明し、この方法は、ソースアフィンブロックを選択するステップであって、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックである、ステップと、ソースアフィンブロックの制御点の動きベクトルを外挿して現在のブロックの制御点の動きベクトル予測子を決定するステップと、アフィン動きベクトル予測子(MVP)セット候補リストに、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入するステップと、アフィンMVPセット候補リストの中のアフィンMVPセットを選択するステップと、ビットストリームにおいて、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)をシグナリングするステップと、ビットストリームにおいて、選択されたアフィンMVPセットのアフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングするステップとを備える。

20

【0008】

別の例では、本開示は、ビデオデータを復号するためのデバイスを説明し、このデバイスは、ビデオデータを記憶するように構成されるメモリと、1つまたは複数の処理回路とを備え、1つまたは複数の処理回路は、ソースアフィンブロックを選択し、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、ソースアフィンブロックの制御点の動きベクトルを外挿して現在のブロックの制御点の動きベクトル予測子を決定し、アフィン動きベクトル予測子(MVP)セット候補リストに、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入し、ビットストリームにおいてシグナリングされるインデックスに基づいて、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定し、ビットストリームから、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)を取得し、選択されたアフィンMVPセットに含まれる動きベクトル予測子およびMVDに基づいて、現在のブロックの制御点の動きベクトルを決定し、現在のブロックの制御点の動きベクトルに基づいて、予測ブロックを生成し、残差データおよび予測ブロックに基づいて現在のブロックを再構築するように構成される。

30

【0009】

別の例では、本開示は、ビデオデータを符号化するためのデバイスを説明し、このデバイスは、ビデオデータを記憶するように構成されるメモリと、1つまたは複数の処理回路とを備え、1つまたは複数の処理回路は、ソースアフィンブロックを選択し、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、ソースアフィンブロックの制御点の動きベクトルを外挿して現在のブロックの制御点の動きベクトル予測子を決定し、アフィン動きベクトル予測子(MVP)セット候補リストに、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入し、アフィンMVPセット候補リストの中のアフィンMVPセットを選択し、ビットストリームにおいて、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)をシグナリングし、ビット

40

50

ストリームにおいて、選択されたアフィンMVPセットのアフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングするように構成される。

【0010】

別の例では、本開示は、ビデオデータを復号するためのデバイスを説明し、このデバイスは、ソースアフィンブロックを選択するための手段であって、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックである、手段と、ソースアフィンブロックの制御点の動きベクトルを外挿して現在のブロックの制御点の動きベクトル予測子を決定するための手段と、アフィン動きベクトル予測子(MVP)セット候補リストに、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入するための手段と、ビットストリームにおいてシグナリングされるインデックスに基づいて、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定するための手段と、ビットストリームから、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)を取得するための手段と、選択されたアフィンMVPセットに含まれる動きベクトル予測子およびMVDに基づいて、現在のブロックの制御点の動きベクトルを決定するための手段と、現在のブロックの制御点の動きベクトルに基づいて、予測ブロックを生成するための手段と、残差データおよび予測ブロックに基づいて現在のブロックを再構築するための手段とを備える。10

【0011】

別の例では、本開示は、ビデオデータを符号化するためのデバイスを説明し、このデバイスは、ソースアフィンブロックを選択するための手段であって、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックである、手段と、ソースアフィンブロックの制御点の動きベクトルを外挿して現在のブロックの制御点の動きベクトル予測子を決定するための手段と、アフィン動きベクトル予測子(MVP)セット候補リストに、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入するための手段と、アフィンMVPセット候補リストの中のアフィンMVPセットを選択するための手段と、ビットストリームにおいて、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)をシグナリングするための手段と、ビットストリームにおいて、選択されたアフィンMVPセットのアフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングするための手段とを備える。20

【0012】

別の例では、本開示は、命令を記憶したコンピュータ可読記憶媒体を説明し、この命令は、実行されると、ビデオ復号のためのデバイスの1つまたは複数の処理回路に、ソースアフィンブロックを選択させ、ソースアフィンブロックが、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり、ソースアフィンブロックの制御点の動きベクトルを外挿させて現在のブロックの制御点の動きベクトル予測子を決定させ、アフィン動きベクトル予測子(MVP)セット候補リストへ、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入させ、ビットストリームにおいてシグナリングされるインデックスに基づいて、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定させ、ビットストリームから、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)を取得させ、選択されたアフィンMVPセットに含まれる動きベクトル予測子およびMVDに基づいて、現在のブロックの制御点の動きベクトルを決定させ、現在のブロックの制御点の動きベクトルに基づいて、予測ブロックを生成させ、残差データおよび予測ブロックに基づいて現在のブロックを再構築させる。30

【0013】

別の例では、本開示は、命令を記憶したコンピュータ可読記憶媒体を説明し、この命令は、実行されると、ビデオデータを符号化するためのデバイスの1つまたは複数の処理回路に、ソースアフィンブロックを選択させ、ソースアフィンブロックが、現在のブロック40

に空間的に隣接するアフィンコーディングされたブロックであり、ソースアフィンブロックの制御点の動きベクトルを外挿させて現在のブロックの制御点の動きベクトル予測子を決定させ、アフィン動きベクトル予測子(MVP)セット候補リストへ、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを挿入させ、アフィンMVPセット候補リストの中のアフィンMVPセットを選択させ、ビットストリームにおいて、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との差分を示す動きベクトル差分(MVD)をシグナリングさせ、ビットストリームにおいて、選択されたアフィンMVPセットのアフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングさせる。

【0014】

10

本開示の1つまたは複数の態様の詳細が、添付の図面および下記の説明に記載される。本開示において説明される技法の他の特徴、目的、および利点は、これらの説明および図面、ならびに特許請求の範囲から明らかになろう。

【図面の簡単な説明】

【0015】

【図1】本開示において説明される1つまたは複数の技法を利用することができる例示的なビデオ符号化および復号システムを示すブロック図である。

【図2A】マージモードのための空間隣接動きベクトル(MV)候補を示す図である。

【図2B】高度動きベクトル予測(AMVP)モードのための空間隣接MV候補を示す図である。

【図3A】時間動きベクトル予測子(TMVP)候補を導出するための例示的な技法を示す概念図である。

20

【図3B】動きベクトルのスケーリングを示す図である。

【図4】現在のブロックのための簡略化されたアフィン動きモデルを示す図である。

【図5】サブブロックごとのアフィン動きベクトルフィールド(MVF)を示す図である。

【図6A】AF_INTERモードにおいて使用されるような現在のブロックおよび隣接ブロックを示すブロック図である。

【図6B】4パラメータのアフィン動きモデルにおいて使用される例示的なアフィンMVPセット候補リストを示す図である。

【図7A】AF_MERGEモードで現在のブロックをコーディングするときに使用される隣接ブロックを示す図である。

30

【図7B】AF_MERGEのための候補を示す図である。

【図8A】6パラメータのアフィン動きモデルにおいて使用される例示的なブロックを示す図である。

【図8B】6パラメータのアフィン動きモデルにおいて使用される例示的なアフィンMVPセット候補リストを示す図である。

【図9】本開示の技法に従って、外挿されたMVPセットを含む例示的なアフィンMVPセット候補リストを示す図である。

【図10】現在のブロックの各サブブロックのアフィン動きがそれ自体の隣接ブロックの外挿された動きから予測され、または直接継承され得る、本開示の技法に従ったサブブロック動き予測またはパラメータ予測を示す図である。

40

【図11A】本開示の技法に従って、4パラメータのアフィン動きモデルのための例示的なアフィンMVPセット候補リストを示す図である。

【図11B】本開示の技法に従って、6パラメータのアフィン動きモデルのための例示的なアフィンMVPセット候補リストを示す図である。

【図12】本開示において説明される1つまたは複数の技法を実施することができる例示的なビデオエンコーダを示すブロック図である。

【図13】本開示において説明される1つまたは複数の技法を実施することができる例示的なビデオデコーダを示すブロック図である。

【図14A】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

50

【図14B】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。

【図15A】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

【図15B】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。

【図16A】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

【図16B】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。 10

【図17】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

【図18】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。

【図19A】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

【図19B】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。 20

【図20A】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

【図20B】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。 20

【図21A】本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。

【図21B】本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。

【発明を実施するための形態】

【0016】

ビデオデータのさらなる圧縮を実現するために、アフィン動きモデルの使用が提案されている。ブロックのためのアフィン動きモデルは、一連のピクチャの中のブロックの回転を表す。ブロックのアフィン動きモデルは、ブロックの制御点の動きベクトルに基づいて決定され得る。いくつかの実装形態では、ブロックの制御点はブロックの左上および右上の角である。いくつかの実装形態では、ブロックの制御点はさらにブロックの左下の角を含む。ビデオコーダ(すなわち、ビデオエンコーダまたはビデオデコーダ)は、ブロックの制御点の動きベクトルに基づいて、ブロックのサブブロックの動きベクトルを計算し得る。 30

【0017】

ブロックの制御点の動きベクトルをシグナリングするために、2つの主要な技法が提案されている。第1の技法はアフィンインターモードである。第2の技法はアフィンマージモードである。アフィンインターモードでは、ビデオエンコーダは、現在のブロックのためのアフィン動きベクトル予測子(MVP)セット候補リストを生成する。アフィンMVPセット候補リストはアフィンMVPセットのリストである。各アフィンMVPセットは、現在のブロックの異なる制御点に対応するMVPのセットである。ビデオエンコーダは、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットをビデオデコーダに対して特定するインデックスをシグナリングする。加えて、ビデオエンコーダは、現在のブロックの制御点の各自の動きベクトル差分(MVD)をシグナリングする。制御点の動きベクトルは、制御点のMVDと選択されたアフィンMVPセットの中の制御点の動きベクトル予測子とを足したものに等しいことがある。ビデオエンコーダはまた、ビデオデコーダが現在のブロックとともに使用する参照ピクチャを特定する参照インデックスをシグナリングする。ビデオデコーダは、同じアフィンMVPセット候補リストを生成し、シグナリングされたインデックスを使 40

用して選択されるアフィンMVPセットを決定する。ビデオデコーダは、MVDを選択されたアフィンMVPセットの動きベクトルに加算して、現在のブロックの制御点の動きベクトルを決定し得る。

【0018】

アフィンマージモードでは、ビデオエンコーダおよびビデオデコーダは、現在のブロックのための同じアフィンソースブロックを特定する。アフィンソースブロックは、現在のブロックに空間的に隣接するアフィンコーディングされたブロックであり得る。ビデオエンコーダおよびビデオデコーダは、アフィンソースブロックの制御点の動きベクトルから、現在のブロックの制御点の動きベクトルを外挿する。たとえば、ビデオエンコーダおよびビデオデコーダは、現在のブロック内の位置の動きベクトルを記述するアフィン動きモデルを構築し得る。アフィン動きモデルは、アフィンパラメータのセットによって定義される。ビデオエンコーダおよびビデオデコーダは、現在のブロックの制御点の動きベクトルに基づいてアフィンパラメータを決定し得る。ビデオエンコーダおよびビデオデコーダは、アフィンソースブロックの制御点の動きベクトルに基づいて、現在のブロックの制御点の動きベクトルを決定し得る。10

【0019】

本開示の1つの例示的な技法によれば、アフィンインター モードでアフィンMVPセット候補リストを生成するとき、ビデオエンコーダは、現在のブロックのためのアフィンMVPセット候補リストに、アフィンソースブロックの制御点の外挿された動きベクトルを指定するアフィンMVPセットを含め得る。この例では、ビデオエンコーダは、アフィンMVPセット候補リストへのインデックス、現在のブロックの各制御点のMVD、および参照インデックスをシグナリングし得る。ビデオデコーダは、現在のブロックのために同じアフィンMVPセット候補リストを生成し得る。加えて、ビデオデコーダは、アフィンMVPセット候補リストへのインデックスを使用して、選択されるアフィンMVP候補セットを特定する。ビデオデコーダは次いで、MVDおよび選択されたアフィンMVP候補セットの動きベクトル予測子を使用して、現在のブロックの制御点の動きベクトルを決定し得る。さらに、ビデオデコーダは、その動きベクトルと参照インデックスにより示される参照ピクチャとを使用して、現在のブロックのための予測ブロックを生成し得る。ビデオデコーダは、現在のブロックの予測ブロックを使用して現在のブロックを再構築し得る。アフィンソースブロックの制御点の外挿された動きベクトルを指定するアフィンMVPセットを、現在のブロックのアフィンMVPセット候補リストに含めることで、コーディング効率を上げることができる。20

【0020】

図1は、本開示の技法を使用し得る例示的なビデオ符号化および復号システム10を示すブロック図である。図1に示されるように、システム10は、宛先デバイス14によって後で復号されるべき符号化されたビデオデータを提供するソースデバイス12を含む。具体的には、ソースデバイス12は、コンピュータ可読媒体16を介して宛先デバイス14にビデオデータを提供する。ソースデバイス12および宛先デバイス14は、デスクトップコンピュータ、ノートブック(すなわち、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、タブレットコンピュータ、テレビジョン、カメラ、ディスプレイデバイス、デジタルメディアプレーヤー、ビデオゲーミングコンソール、ビデオストリーミングデバイスなどを含む、広範囲のデバイスのうちのいずれかを備え得る。いくつかの場合、ソースデバイス12および宛先デバイス14は、ワイヤレス通信に対応し得る。したがって、ソースデバイス12および宛先デバイス14は、ワイヤレス通信デバイスであり得る。ソースデバイス12は、例示的なビデオ符号化デバイス(すなわち、ビデオデータを符号化するためのデバイス)である。宛先デバイス14は例示的なビデオ復号デバイス(すなわち、ビデオデータを復号するためのデバイス)である。40

【0021】

図1の例では、ソースデバイス12は、ビデオソース18、ビデオデータを記憶するように構成された記憶媒体19、ビデオエンコーダ20、および出力インターフェース22を含む。宛50

先デバイス14は、入力インターフェース26、符号化されたビデオデータを記憶するように構成された記憶媒体28、ビデオデコーダ30、およびディスプレイデバイス32を含む。他の例では、ソースデバイス12および宛先デバイス14は、他の構成要素または構成を含む。たとえば、ソースデバイス12は、外部カメラなどの外部ビデオソースからビデオデータを受信し得る。同様に、宛先デバイス14は、統合されたディスプレイデバイスを含むのではなく、外部ディスプレイデバイスとインターフェースしてもよい。

【 0 0 2 2 】

図1の図示されるシステム10は一例にすぎない。ビデオデータを処理するための技法は、任意のデジタルビデオ符号化および/または復号デバイスによって実行され得る。一般に、本開示の技法はビデオ符号化デバイスによって実行されるが、技法はまた、通常は「コーデック」と呼ばれるビデオエンコーダ/デコーダによって実行され得る。ソースデバイス12および宛先デバイス14は、単に、ソースデバイス12が宛先デバイス14に送信するためのコーディングされたビデオデータを生成するようなコーディングデバイスの例である。いくつかの例では、ソースデバイス12および宛先デバイス14は、ソースデバイス12および宛先デバイス14の各々がビデオ符号化構成要素および復号構成要素を含むように実質的に対称的な方式で動作し得る。したがって、システム10は、たとえば、ビデオストリーミング、ビデオ再生、ビデオ放送、またはビデオ電話のための、ソースデバイス12と宛先デバイス14との間の一方向または双方向のビデオ送信をサポートし得る。

【 0 0 2 3 】

ソースデバイス12のビデオソース18は、ビデオカメラ、以前にキャプチャされたビデオを含むビデオアーカイブ、および/またはビデオコンテンツプロバイダからビデオデータを受信するためのビデオフィードインターフェースなどの、ビデオキャプチャデバイスを含み得る。さらなる代替として、ビデオソース18は、ソースビデオとしてのコンピュータグラフィックスベースのデータ、またはライブビデオとアーカイブされたビデオとコンピュータで生成されたビデオとの組合せを生成し得る。ソースデバイス12は、ビデオデータを記憶するように構成された1つまたは複数のデータ記憶媒体(たとえば、記憶媒体19)を備え得る。本開示において説明される技法は、ビデオコーディング全般に適用可能であることがあり、ワイヤレスおよび/または有線の適用例において適用されることがある。各々の場合において、キャプチャされた、事前にキャプチャされた、またはコンピュータで生成されたビデオは、ビデオエンコーダ20によって符号化され得る。出力インターフェース22は、符号化されたビデオ情報をコンピュータ可読媒体16に出力し得る。

【 0 0 2 4 】

出力インターフェース22は、様々なタイプの構成要素またはデバイスを備え得る。たとえば、出力インターフェース22は、ワイヤレス送信機、モ뎀、有線ネットワーキング構成要素(たとえば、イーサネット(登録商標)カード)、または別の物理的な構成要素を備え得る。出力インターフェース22がワイヤレス受信機を備える例では、出力インターフェース22は、4G、4G-LTE、LTE Advanced、5Gなどのセルラー通信規格に従って変調される、ビットストリームなどのデータを受信するように構成され得る。出力インターフェース22がワイヤレス受信機を備えるいくつかの例では、出力インターフェース22は、IEEE 802.11仕様、IEEE 802.15仕様(たとえば、ZigBee(商標))、Bluetooth(登録商標)規格などの他のワイヤレス規格に従って変調される、ビットストリームなどのデータを受信するように構成され得る。いくつかの例では、出力インターフェース22の回路は、ビデオエンコーダ20の回路および/またはソースデバイス12の他の構成要素に組み込まれ得る。たとえば、ビデオエンコーダ20および出力インターフェース22は、システムオンチップ(SoC)の一部であり得る。SoCは、汎用マイクロプロセッサ、グラフィックス処理装置などの、他の構成要素も含み得る。

【 0 0 2 5 】

宛先デバイス14は、コンピュータ可読媒体16を介して、復号されるべき符号化されたビデオデータを受信し得る。コンピュータ可読媒体16は、ソースデバイス12から宛先デバイス14に符号化されたビデオデータを移動することが可能な任意のタイプの媒体またはデバ

10

20

30

40

50

イスを備え得る。いくつかの例では、コンピュータ可読媒体16は、ソースデバイス12がリアルタイムで宛先デバイス14に符号化されたビデオデータを直接送信することを可能にする通信媒体を含む。符号化されたビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、宛先デバイス14に送信され得る。通信媒体は、高周波(RF)スペクトラルまたは1つまたは複数の物理伝送線路などの、任意のワイヤレスまたは有線通信媒体を備え得る。通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークなどの、パケットベースネットワークの一部を形成し得る。通信媒体は、ルータ、スイッチ、基地局、またはソースデバイス12から宛先デバイス14への通信を容易にするのに有用であり得る任意の他の機器を含み得る。宛先デバイス14は、符号化されたビデオデータおよび復号されたビデオデータを記憶するように構成された1つまたは複数のデータ記憶媒体を備え得る。

10

【0026】

いくつかの例では、符号化されたデータは、出力インターフェース22から記憶デバイスに出力され得る。同様に、符号化されたデータは、入力インターフェースによって記憶デバイスからアクセスされ得る。記憶デバイスは、ハードドライブ、Blu-ray(登録商標)ディスク、DVD、CD-ROM、フラッシュメモリ、揮発性メモリもしくは不揮発性メモリ、または符号化されたビデオデータを記憶するための任意の他の好適なデジタル記憶媒体などの、分散されるかまたはローカルにアクセスされる様々なデータ記憶媒体のいずれかを含み得る。さらなる例では、記憶デバイスは、ソースデバイス12によって生成された符号化されたビデオを記憶し得るファイルサーバまたは別の中間記憶デバイスに相当し得る。宛先デバイス14は、ストリーミングまたはダウンロードを介して記憶デバイスからの記憶されたビデオデータにアクセスし得る。ファイルサーバは、符号化されたビデオデータを記憶するとともにその符号化されたビデオデータを宛先デバイス14へ送信することが可能な、任意のタイプのサーバであり得る。例示的なファイルサーバは、(たとえば、ウェブサイトのための)ウェブサーバ、FTPサーバ、ネットワークアタッチストレージ(NAS)デバイス、またはローカルディスクドライブを含む。宛先デバイス14は、インターネット接続を含む任意の標準的なデータ接続を通じて符号化されたビデオデータにアクセスし得る。これは、ワイヤレスチャネル(たとえば、Wi-Fi接続)、有線接続(たとえば、DSL、ケーブルモ뎀など)、またはファイルサーバに記憶された符号化されたビデオデータにアクセスするのに適した両方の組合せを含み得る。記憶デバイスからの符号化されたビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはそれらの組合せであり得る。

20

【0027】

技法は、オーバージエアレビジョン放送、ケーブルテレビジョン送信、有線送信、衛星テレビジョン送信、動的適応ストリーミングオーバーHTTP(DASH:dynamic adaptive streaming over HTTP)などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されているデジタルビデオ、データ記憶媒体に記憶されたデジタルビデオの復号、または他の適用例もしくは上の例の組合せなどの、様々なマルチメディア用途のいずれかをサポートするビデオコーディングに適用され得る。いくつかの例では、システム10は、ビデオストリーミング、ビデオ再生、ビデオ放送、および/またはビデオ電話などの適用例をサポートするために、一方向または双方向のビデオ送信をサポートするように構成され得る。

30

【0028】

コンピュータ可読媒体16は、ワイヤレス放送もしくは有線ネットワーク送信などの一時的媒体、またはハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-ray(登録商標)ディスク、もしくは他のコンピュータ可読媒体などの記憶媒体(すなわち、非一時的記憶媒体)を含み得る。いくつかの例では、ネットワークサーバ(図示せず)が、たとえば、ネットワーク送信を介して、ソースデバイス12から符号化されたビデオデータを受信し、符号化されたビデオデータを宛先デバイス14に提供し得る。同様に、ディスクスタンピング設備などの媒体製造設備のコンピューティングデバイスが、ソースデバイス12から符号化されたビデオデータを受信し、符号化されたビデオデータを記憶する。

40

50

タを含むディスクを製造し得る。したがって、コンピュータ可読媒体16は、様々な例において、様々な形態の1つまたは複数のコンピュータ可読媒体を含むものと理解され得る。

【0029】

宛先デバイス14の入力インターフェース26は、コンピュータ可読媒体16から情報を受信する。コンピュータ可読媒体16の情報は、ビデオエンコーダ20によって定義され、ビデオデコーダ30によっても使用される、シンタックス情報を含むことがあり、シンタックス情報は、ブロックおよび他のコーディングされたユニット、たとえばピクチャグループ(GOP)の特性および/または処理を記述するシンタックス要素を含む。入力インターフェース26は、様々なタイプの構成要素またはデバイスを備え得る。たとえば、入力インターフェース26は、ワイヤレス受信機、モ뎀、有線ネットワーキング構成要素(たとえば、イーサネット(登録商標)カード)、または別の物理的な構成要素を備え得る。入力インターフェース26がワイヤレス受信機を備える例では、入力インターフェース26は、4G、4G-LTE、LTE Advanced、5Gなどのセルラー通信規格に従って変調される、ビットストリームなどのデータを受信するように構成され得る。入力インターフェース26がワイヤレス受信機を備えるいくつかの例では、入力インターフェース26は、IEEE 802.11仕様、IEEE 802.15仕様(たとえば、ZigBee(商標))、Bluetooth(登録商標)規格などの他のワイヤレス規格に従って変調される、ビットストリームなどのデータを受信するように構成され得る。いくつかの例では、入力インターフェース26の回路は、ビデオデコーダ30の回路および/または宛先デバイス14の他の構成要素に組み込まれ得る。たとえば、ビデオデコーダ30および入力インターフェース26は、SoCの一部であり得る。SoCは、汎用マイクロプロセッサ、グラフィックス処理装置などの、他の構成要素も含み得る。10

【0030】

記憶媒体28は、入力インターフェース26によって受け取られた符号化されたビデオデータ(たとえば、ビットストリーム)などの、符号化されたビデオデータを記憶するように構成され得る。ディスプレイデバイス32は、復号されたビデオデータをユーザに表示し、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)ディスプレイ、または別のタイプのディスプレイデバイスなどの、様々なディスプレイデバイスのいずれかを備え得る。20

【0031】

ビデオエンコーダ20およびビデオデコーダ30は各々、1つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、ディスクリート論理、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組合せなどの、様々な適切なエンコーダ回路のいずれかとして実装され得る。技法が部分的にソフトウェアで実装されると、デバイスは、適切な非一時的コンピュータ可読媒体の中にソフトウェア用の命令を記憶し、本開示の技法を実行するために1つまたは複数のプロセッサを使用してハードウェアにおいて命令を実行し得る。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つまたは複数のエンコーダまたはデコーダに含まれることがあり、それらのいずれもが、それぞれのデバイスの中で複合エンコーダ/デコーダ(コーデック)の一部として統合されることがある。30

【0032】

いくつかの例では、ビデオエンコーダ20およびビデオデコーダ30は、現行はたは将来の規格などのビデオコーディング規格に従って動作し得る。例示的なビデオコーディング規格は、限定はされないが、ITU-T H.261、ISO/IEC MPEG-1 Visual、ITU-T H.262またはISO/IEC MPEG-2 Visual、ITU-T H.263、ISO/IEC MPEG-4 VisualおよびITU-T H.264(ISO/IEC MPEG-4 AVCとしても知られる)(そのスケーラブルビデオコーディング(SVC)拡張およびマルチビュービデオコーディング(MVC)拡張を含む)を含む。加えて、新しいビデオコーディング規格、すなわちHigh Efficiency Video Coding(HEVC)またはITU-T H.265が、その範囲拡張、スクリーンコンテンツコーディング拡張、3Dビデオコーディング拡張(3D-HEVC)、マルチビュー拡張(MV-HEVC)およびスケーラブル拡張(SHVC)を含めて、Joint Collaboration Team on Video Coding(JCT-VC)、ならびに、ITU-T Video Coding Experts Group(VC4050

EG)およびISO/IEC Motion Picture Experts Group(MPEG)のJoint Collaboration Team on 3D Video Coding Extension Development(JCT-3V)によって最近開発された。以後HEVC WDと呼ばれる、HEVC暫定仕様書は、http://phenix.int-evry.fr/jct/doc_end_user/documents/14_Vienna/wg11/JCTVC-N1003-v1.zipから入手可能な、Wang他、「High Efficiency Video Coding (HEVC) Defect Report」、Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11、第14回会合、ウィーン、オーストリア、2013年7月25日～8月2日、文書JCTVC-N1003_v1から入手可能である。HEVCはまた、Recommendation ITU-T H.265, Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services - Coding of moving video、High efficiency video coding、2016年12月として公開されている。

10

【 0 0 3 3 】

ITU-T VCEG(Q6/16)およびISO/IEC MPEG(JTC 1/SC 29/WG 11)は、現在のHEVC規格(その現在の拡張ならびにスクリーンコンテンツコーディングおよびハイダイナミックレンジコーディングのための目先の拡張を含む)の圧縮能力を大きく超える圧縮能力を有する、将来のビデオコーディング技術の標準化の潜在的な必要性について現在研究している。このグループは、この分野の専門家により提案されている圧縮技術の設計を評価するために、Joint Video Exploration Team(JVET)として知られている共同研究においてこの調査活動に一緒に取り組んでいる。JVETは、2015年10月19～21日に最初の会合を開いた。参照ソフトウェアの最新バージョン、すなわちJoint Exploration Model 3 (JEM 3)は、https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-3.0/からダウンロードすることができる。J. Chen、E. Alshina、G. J. Sullivan、J.-R. Ohm、J. Boyce、「Algorithm Description of Joint Exploration Test Model 3」、JVET-C1001、2016年5月、(以後「JVET-C1001」)は、Joint Exploration Test Model 3 (JEM3.0)のアルゴリズム記述を含む。

20

【 0 0 3 4 】

HEVCおよび他のビデオコーディング規格では、ビデオデータは一連のピクチャを含む。ピクチャは、「フレーム」と呼ばれることもある。ピクチャは、1つまたは複数のサンプルアレイを含み得る。ピクチャの各々のそれぞれのサンプルアレイは、それぞれの色成分に対するサンプルのアレイを備え得る。HEVCでは、ピクチャは、 S_L 、 S_{Cb} 、および S_{Cr} と表記される3つのサンプルアレイを含み得る。 S_L は、輝度サンプルの2次元アレイ(すなわち、ブロック)である。 S_{Cb} は、Cb色差サンプルの2次元アレイである。 S_{Cr} は、Cr色差サンプルの2次元アレイである。他の事例では、ピクチャはモノクロームであってよく、輝度サンプルのアレイしか含まないことがある。

30

【 0 0 3 5 】

ビデオデータを符号化することの一部として、ビデオエンコーダ20はビデオデータのピクチャを符号化し得る。言い換えると、ビデオエンコーダ20は、ビデオデータのピクチャの符号化された表現を生成し得る。ピクチャの符号化された表現は、「コーディングされたピクチャ」または「符号化されたピクチャ」と本明細書で呼ばれ得る。

【 0 0 3 6 】

ピクチャの符号化された表現を生成するために、ビデオエンコーダ20は、ピクチャのブロックを符号化し得る。ビデオエンコーダ20は、ビットストリームに、ビデオブロックの符号化された表現を含め得る。たとえば、ピクチャの符号化された表現を生成するために、ビデオエンコーダ20は、ピクチャの各サンプルアレイをコーディングツリーブロック(CTB)へと区分し、CTBを符号化し得る。CTBは、ピクチャのサンプルアレイにおけるサンプルの $N \times N$ ブロックであり得る。HEVCメインプロファイルでは、CTBのサイズは 16×16 から 64×64 にわたり得るが、技術的には 8×8 のCTBサイズがサポートされ得る。

40

【 0 0 3 7 】

ピクチャのコーディングツリー単位(CTU)は、1つまたは複数のCTBを備えることがあり、1つまたは複数のCTBのサンプルを符号化するために使用されるシンタックス構造を備えることがある。たとえば、各CTUは、輝度サンプルのCTBと、色差サンプルの2つの対応す

50

るCTBと、CTBのサンプルを符号化するために使用されるシンタックス構造とを備え得る。モノクロームピクチャまたは3つの別々の色平面を有するピクチャでは、CTUは、単一のCTBと、そのCTBのサンプルを符号化するために使用されるシンタックス構造とを備え得る。CTUは、「ツリーブロック」または「最大コーディング単位」(LCU)と呼ばれることがある。本開示では、「シンタックス構造」は、指定された順序でビットストリームに一緒に現れる0個以上のシンタックス要素として定義され得る。いくつかのコーデックでは、符号化されたピクチャは、ピクチャのすべてのCTUを含む符号化された表現である。

【 0 0 3 8 】

ピクチャのCTUを符号化するために、ビデオエンコーダ20は、CTUのCTBを1つまたは複数のコーディングブロックへと区分し得る。コーディングブロックは、サンプルの $N \times N$ ブロックである。いくつかのコーデックでは、ピクチャのCTUを符号化するために、ビデオエンコーダ20は、CTUのコーディングツリーブロック上で四分木区分を再帰的に実行してCTBをコーディングブロックへと区分することができ、したがって、「コーディングツリー単位」という名前である。コーディング単位(CU)は、1つまたは複数のコーディングブロックと、1つまたは複数のコーディングブロックのサンプルを符号化するために使用されるシンタックス構造とを備え得る。たとえば、CUは、輝度サンプルアレイ、Cbサンプルアレイ、およびCrサンプルアレイを有するピクチャの輝度サンプルのコーディングブロックおよび色差サンプルの2つの対応するコーディングブロックと、コーディングブロックのサンプルを符号化するために使用されるシンタックス構造とを備え得る。モノクロームピクチャまたは3つの別個の色平面を有するピクチャでは、CUは、単一のコーディングブロック、およびコーディングブロックのサンプルをコーディングするために使用されるシンタックス構造を備え得る。10

【 0 0 3 9 】

さらに、ビデオエンコーダ20は、ビデオデータのピクチャのCUを符号化し得る。いくつかのコーデックでは、CUを符号化することの一部として、ビデオエンコーダ20が、CUのコーディングブロックを1つまたは複数の予測ブロックに区分し得る。予測ブロックは、同じ予測が適用されるサンプルの長方形(すなわち、正方形または非正方形)ブロックである。CUの予測単位(PU)は、CUの1つまたは複数の予測ブロックと、1つまたは複数の予測ブロックを予測するために使用されるシンタックス構造とを備え得る。たとえば、PUは、輝度サンプルの予測ブロック、色差サンプルの2つの対応する予測ブロック、および予測ブロックを予測するために使用されるシンタックス構造を備え得る。モノクロームピクチャまたは3つの別個の色平面を有するピクチャでは、PUは、単一の予測ブロック、および予測ブロックを予測するために使用されるシンタックス構造を備え得る。20

【 0 0 4 0 】

ビデオエンコーダ20は、CUの予測(prediction)ブロック(たとえば、輝度予測ブロック、Cb予測ブロック、およびCr予測ブロック)の予測(predictive)ブロック(たとえば、輝度予測ブロック、Cb予測ブロック、およびCr予測ブロック)を生成し得る。ビデオエンコーダ20は、予測ブロックを生成するためにイントラ予測またはインター予測を使用し得る。ビデオエンコーダ20が予測ブロックを生成するためにイントラ予測を使用する場合、ビデオエンコーダ20は、PUを含むピクチャの復号されたサンプルに基づいて、予測ブロックを生成し得る。ビデオエンコーダ20が現在のピクチャのCUの予測ブロックを生成するためにインター予測を使用する場合、ビデオエンコーダ20は、参照ピクチャ(すなわち、現在のピクチャ以外のピクチャ)の復号されたサンプルに基づいて、CUの予測ブロックを生成し得る。40

【 0 0 4 1 】

HEVCおよび特定の他のコーデックでは、ビデオエンコーダ20は、1つの予測モード(すなわち、イントラ予測またはインター予測)のみを使用してCUを符号化する。したがって、HEVCおよび特定の他のコーデックでは、ビデオエンコーダ20は、イントラ予測を使用してCUの予測ブロックを生成することができ、ビデオエンコーダ20は、インター予測を使用してCUの予測ブロックを生成することができる。ビデオエンコーダ20がCUを符号化するため50

にインター予測を使用するとき、ビデオエンコーダ20は、CUを2つまたは4つのPUへと区分することがあり、または1つのPUがCU全体に相当する。1つのCUの中に2つのPUが存在するとき、2つのPUは、CUの半分のサイズの長方形であることがあり、またはCUの1/4もしくは3/4のサイズを有する2つの長方形であることがある。HEVCでは、インター予測モードでコーディングされるCUに対して8つの予測モード、すなわち、PART_2N × 2N、PART_2N × N、PART_N × 2N、PART_N × N、PART_2N × nU、PART_2N × nD、PART_nL × 2N、およびPART_nR × 2Nがある。CUがイントラ予測されるとき、2N × 2NおよびN × Nのみが許されるPU形状であり、各PU内で、単一のイントラ予測モードがコーディングされる(一方で色差予測モードはCUレベルでシグナリングされる)。

【0042】

10

ビデオエンコーダ20は、CUの1つまたは複数のCr残差ブロックを生成し得る。たとえば、ビデオエンコーダ20は、CUの輝度残差ブロックを生成し得る。CUの輝度残差ブロックの中の各サンプルは、CUの予測輝度ブロックの1つの中の輝度サンプルとCUの元の輝度コーディングブロック中の対応するサンプルとの差分を示す。加えて、ビデオエンコーダ20は、CUのCb残差ブロックを生成し得る。CUのCb残差ブロックの中の各サンプルは、CUの予測Cbブロックのうちの1つの中のCbサンプルと、CUの元のCbコーディングブロックの中の対応するサンプルとの差分を示し得る。ビデオエンコーダ20はまた、CUのCr残差ブロックを生成し得る。CUのCr残差ブロックの中の各サンプルは、CUの予測Crブロックのうちの1つの中のCrサンプルと、CUの元のCrコーディングブロックの中の対応するサンプルとの差分を示し得る。

【0043】

20

さらに、ビデオエンコーダ20は、CUの残差ブロックを1つまたは複数の変換ブロックに分解し得る。たとえば、ビデオエンコーダ20は、四分木区分を使用してCUの残差ブロックを1つまたは複数の変換ブロックに分解し得る。変換ブロックとは、同じ変換が適用されるサンプルの長方形(たとえば、正方形または非正方形)ブロックである。CUの変換単位(TU)は、1つまたは複数の変換ブロックを備え得る。たとえば、TUは、輝度サンプルの変換ブロック、色差サンプルの2つの対応する変換ブロック、およびそれらの変換ブロックサンプルを変換するために使用されるシンタックス構造を備え得る。したがって、CUの各TUは、輝度変換ブロック、Cb変換ブロック、およびCr変換ブロックを有し得る。TUの輝度変換ブロックは、CUの輝度残差ブロックのサブブロックであり得る。Cb変換ブロックは、CUのCb残差ブロックのサブブロックであり得る。Cr変換ブロックは、CUのCr残差ブロックのサブブロックであり得る。モノクロームピクチャまたは3つの別個の色平面を有するピクチャでは、TUは、単一の変換ブロック、および変換ブロックのサンプルを変換するために使用されるシンタックス構造を含み得る。

【0044】

30

ビデオエンコーダ20は、1つまたは複数の変換をTUの変換ブロックに適用して、TUの係数ブロックを生成し得る。係数ブロックは変換係数の2次元アレイであり得る。変換係数は、スカラー量であり得る。いくつかの例では、1つまたは複数の変換は、画素領域から周波数領域に変換ブロックを変換する。したがって、そのような例では、変換係数は、周波数領域において考慮されるべきスカラー量であり得る。変換係数レベルは、変換係数値の計算のためのスケーリングの前の復号プロセスにおける特定の2次元周波数インデックスと関連付けられる値を表す整数量である。

【0045】

40

いくつかの例では、ビデオエンコーダ20は、変換ブロックへの変換の適用をスキップする。そのような例では、ビデオエンコーダ20は、変換係数と同じ方法で残差サンプル値を扱い得る。したがって、ビデオエンコーダ20が変換の適用をスキップする例では、変換係数および係数ブロックの以下の議論は、残差サンプルの変換ブロックに適用可能であり得る。

【0046】

係数ブロックを生成した後、ビデオエンコーダ20は、係数ブロックを量子化し得る。量

50

子化は一般に、変換係数を表すために使用されるデータの量をできるだけ減らすために変換係数が量子化され、さらなる圧縮をもたらすプロセスを指す。いくつかの例では、ビデオエンコーダ20は量子化をスキップする。ビデオエンコーダ20が係数ブロックを量子化した後、ビデオエンコーダ20は、量子化された変換係数を示すシンタックス要素を生成し得る。ビデオエンコーダ20は、量子化された変換係数を示すシンタックス要素のうちの1つまたは複数をエントロピー符号化し得る。たとえば、ビデオエンコーダ20は、量子化された変換係数を示すシンタックス要素に対してコンテキスト適応型バイナリ算術コーディング(CABAC)を実行することができる。したがって、符号化されたブロック(たとえば、符号化されたCU)は、量子化された変換係数を示すエントロピー符号化されたシンタックス要素を含み得る。

10

【0047】

ビデオエンコーダ20は、符号化されたビデオデータを含むビットストリームを出力し得る。言い換えると、ビデオエンコーダ20は、ビデオデータの符号化された表現を含むビットストリームを出力し得る。たとえば、ビットストリームは、ビデオデータの符号化されたピクチャの表現および関連するデータを形成するビットのシーケンスを備え得る。いくつかの例では、コーディングされたピクチャの表現は、ブロックの符号化された表現を含み得る。

【0048】

ビットストリームは、ネットワーク抽象化レイヤ(NAL)ユニットのシーケンスを備え得る。NALユニットは、NALユニットの中のデータのタイプの指示、および必要に応じてエミュレーション防止ビットが散在させられているローバイトシーケンスペイロード(RBSP: raw byte sequence payload)の形態でそのデータを含むバイトを含む、シンタックス構造である。NALユニットの各々は、NALユニットヘッダを含むことがあり、RBSPをカプセル化する。NALユニットヘッダは、NALユニットタイプコードを示すシンタックス要素を含み得る。NALユニットのNALユニットヘッダによって指定されるNALユニットタイプコードは、NALユニットのタイプを示す。RBSPは、NALユニット内にカプセル化されている整数個のバイトを含むシンタックス構造であり得る。いくつかの事例では、RBSPは0ビットを含む。

20

【0049】

ビデオデコーダ30は、ビデオエンコーダ20によって生成されたビットストリームを受信し得る。上で述べられたように、ビットストリームは、ビデオデータの符号化された表現を備え得る。ビデオデコーダ30は、ビットストリームを復号してビデオデータのピクチャを再構築し得る。ビットストリームを復号することの一部として、ビデオデコーダ30は、ビットストリームを構文解析してビットストリームからシンタックス要素を取得し得る。ビデオデコーダ30は、ビットストリームから取得されたシンタックス要素に少なくとも一部に基づいて、ビデオデータのピクチャを再構築し得る。ビデオデータのピクチャを再構築するためのプロセスは、全般に、ピクチャを符号化するためにビデオエンコーダ20によって実行されるプロセスの逆であり得る。たとえば、ビデオデコーダ30は、インター予測またはイントラ予測を使用して、現在のCUの各PUの1つまたは複数の予測ブロックを生成することができ、PUの動きベクトルを使用して現在のCUのPUの予測ブロックを決定することができる。加えて、ビデオデコーダ30は、現在のCUのTUの係数ブロックを逆量子化し得る。ビデオデコーダ30は、係数ブロックに対して逆変換を実行して、現在のCUのTUの変換ブロックを再構築し得る。いくつかの例では、ビデオデコーダ30は、現在のCUのPUの予測ブロックのサンプルを現在のCUのTUの変換ブロックの対応する復号されたサンプルに加算することによって、現在のCUのコーディングブロックを再構築し得る。ピクチャのCUごとにコーディングブロックを再構築することによって、ビデオデコーダ30はピクチャを再構築することができる。

30

【0050】

ピクチャのスライスは、ピクチャの整数個のCTUを含み得る。スライスのCTUは、ラスター走査順序などの走査順序で連続的に順序付けられ得る。HEVCでは、スライスは、1つの独立スライスセグメントに含まれ、同じアクセス単位内に次の独立スライスセグメントが

40

50

(もしあれば)それに先行するすべての後続の従属スライスセグメントが(もしあれば)それらに含まれる、整数個のCTUとして定義される。さらに、HEVCでは、スライスセグメントは、タイル走査において連続的に順序付けられ单一のNALユニットに含まれている、整数個のコーディングツリー単位として定義される。タイル走査は、CTBがタイルにおいてCTBラスター走査で連続的に順序付けられる一方で、ピクチャの中のタイルがピクチャのタイルのラスター走査で連続的に順序付けられるような、ピクチャを区分するCTBの特定の逐次的な順序付けである。タイルは、ピクチャにおける特定のタイル列および特定のタイル行の中のCTUの長方形領域である。スライスセグメントヘッダは、スライスセグメントで表される第1またはすべてのコーディングツリー単位に関係するデータ要素を含むコーディングされたスライスセグメントの一部である。「スライスヘッダ」という用語は、現在のスライスセグメントである独立スライスセグメントの、または復号順序において現在の従属スライスセグメントに先行する直近の独立スライスセグメントの、スライスセグメントヘッダに適用される。

【0051】

上で簡単に言及されたように、HEVCでは、スライスの中の最大のコーディング単位は、コーディングツリーブロック(CTB)またはコーディングツリー単位(CTU)と呼ばれる。CTBは、そのノードがコーディング単位である四分木を含む。(技術的には 8×8 のCTBサイズがサポートされ得るが)CTBのサイズは、HEVCメインプロファイルでは 16×16 から 64×64 にわたり得る。コーディング単位(CU)は、同じサイズのCTBであり得るが、 8×8 程度に小さくてもよい。各コーディング単位は、1つのモードを用いてコーディングされる。CUがインターフォーマンスされるとき、CUは、2つもしくは4つの予測単位(PU)へとさらに区分されることがあり、またはさらなる区分が適用されないとには1つだけのPUになることがある。1つのCUの中に2つのPUが存在するとき、PUは、CUの半分のサイズの長方形であることがあり、またはCUの1/4もしくは3/4のサイズを有する2つの長方形であることがある。CUがインターフォーマンスされるとき、動き情報の1つのセットが各PUに対して存在する。加えて、各PUは、動き情報のセットを導出するために固有のインターフォーマンスモードでコーディングされる。いくつかのビデオコーディング規格では、CUは複数のPUに分割されない。したがって、そのようなビデオコーディング規格では、PUとCUの区別はない。したがって、本開示の技法がそのような規格において適用されるとき、PUの議論はCUに適用可能であり得る。

【0052】

ビデオコーダは、現在のブロック(たとえば、CUまたはPU)のための单方向インター予測または双方向インター予測を実行し得る。現在のブロックのための单方向インター予測を実行するとき、ビデオコーダは、動きベクトルを使用して参照ピクチャにおける位置を決定する。ビデオコーダは次いで、現在のブロックの予測ブロックを生成し得る。予測ブロックは、動きベクトルによって示される位置における参照ピクチャの中のサンプルのブロック、または参照ピクチャのサンプルから補間されるサンプルのブロックを備え得る。双方向インター予測を実行するとき、ビデオコーダは、第2の参照ピクチャおよび第2の動きベクトルを用いてこのプロセスを実行し、それにより、現在のブロックの第2の予測ブロックを生成し得る。双方向インター予測では、單一の参照ピクチャから生成された予測ブロックは、予備予測ブロックと本明細書で呼ばれることがある。さらに、双方向インター予測では、ビデオコーダは、2つの予備ブロックに基づいて、現在のブロックの最終予測ブロックを生成し得る。いくつかの例では、ビデオコーダは、最終予測ブロックの中の各サンプルが予備予測ブロックの中の対応するサンプルの加重平均となるように、最終予測ブロックを生成し得る。

【0053】

ピクチャにおけるインター予測をサポートするために、ビデオコーダは、ピクチャのために2つの参照ピクチャリストを生成する。ピクチャの参照ピクチャリストは、そのピクチャの中のブロックのインター予測を実行する際に使用可能な参照ピクチャを含む。2つの参照ピクチャリストは、リスト0およびリスト1と一般に呼ばれる。一例では、ピクチャ

10

20

30

40

50

のリスト0の中の各参照ピクチャは、出力順序においてそのピクチャの前にある。この例では、ピクチャのリスト1の中の各参照ピクチャは、出力順序においてそのピクチャの後にある。したがって、リスト0の中の参照ピクチャの使用は第1のインター予測方向であると見なされることがあり、リスト1の中の参照ピクチャの使用は第2のインター予測方向であると見なされることがある。ビデオエンコーダ20およびビデオデコーダ30は、同じ順序の参照ピクチャを伴うピクチャのリスト0を生成する。同様に、ビデオエンコーダ20およびビデオデコーダ30は、同じ順序の参照ピクチャを伴うピクチャのリスト1を生成する。したがって、ビデオエンコーダ20は、参照ピクチャの参照ピクチャリストにおける位置を示す参照インデックスをシグナリングすることによって、参照ピクチャリストの中のある参照ピクチャをビデオデコーダ30に示し得る。

10

【0054】

HEVC規格は、マージモードおよび高度動きベクトル予測(AMVP)モードを含む、複数のインター予測モードを提供する。マージモードでは、ビデオエンコーダ20およびビデオデコーダ30は、PUのための一一致するマージ動きベクトル(MV)候補リストを生成する。PUのためのマージMV候補リストは1つまたは複数のマージ候補を含み、これらのマージ候補は動きベクトル予測子(MVP)とも呼ばれることがある。HEVCでは、マージMV候補リストは最大で5つのマージ候補を含む。マージMV候補リストの中の各々のそれぞれのマージ候補は、1つまたは複数の動きベクトルおよび1つまたは複数の参照インデックスを指定する。たとえば、マージ候補はリスト0動きベクトルおよび/またはリスト1動きベクトルを指定することがあり、リスト0参照インデックスおよび/またはリスト1参照インデックスを指定することがある。リスト0動きベクトルは、リスト0の中の参照ピクチャにおける位置を示す動きベクトルである。リスト1動きベクトルは、リスト1の中の参照ピクチャにおける位置を示す動きベクトルである。ビデオエンコーダ20は、PUのための選択されたマージ候補のマージMV候補リストにおける位置を示すマージインデックスをシグナリングし得る。ビデオデコーダ30は、マージインデックスを使用して選択されたマージ候補を特定し得る。ビデオデコーダ30は次いで、選択されたマージ候補の動きベクトルおよび参照インデックスを、PUの動きベクトルおよび参照インデックスとして使用し得る。

20

【0055】

AMVPモードでは、ビデオエンコーダ20は、PUのためのリスト0AMVP候補リストおよび/またはリスト1AMVP候補リストを生成し、これらのいずれもがAMVP候補リストと呼ばれ得る。ビデオデコーダ30は、ビデオエンコーダ20によって生成されるAMVP候補リストと一致するAMVP候補リストを生成する。HEVCでは、AMVP候補リストは2つのAMVP候補を含む。リスト0AMVP候補リストの中の各々のそれぞれのAMVP候補が、それぞれのリスト0動きベクトルを指定する。リスト1AMVP候補リストの中の各々のそれぞれのAMVP候補が、それぞれのリスト1動きベクトルを指定する。AMVPモードにおいて、PUがリスト0から单方向インター予測され、または双方向インター予測される場合、ビデオエンコーダ20は、リスト0MVPインデックス、リスト0参照インデックス、およびリスト0動きベクトル差分(MVD)をシグナリングする。リスト0MVPインデックスは、リスト0AMVP候補リストにおける選択されたAMVP候補の位置を指定する。リスト0参照インデックスは、選択されたリスト0参照ピクチャの位置を指定する。リスト0MVDは、PUのリスト0動きベクトルと、リスト0AMVP候補リストの中の選択されたAMVP候補によって指定されるリスト0動きベクトルとの間の差分を指定する。したがって、ビデオデコーダ30は、リスト0MVPインデックスおよびリスト0MVDを使用して、PUのリスト0動きベクトルを決定し得る。ビデオデコーダ30は次いで、PUのリスト0動きベクトルによって特定される選択されたリスト0参照ピクチャにおける位置に対応するサンプルを備える、PUの予備予測ブロックまたは最終予測ブロックを決定し得る。ビデオエンコーダ20はリスト1のための同様のシンタックス要素をシグナリングすることができ、ビデオデコーダ30はリスト1のためのシンタックス要素を同様の方法で使用することができます。

30

【0056】

以上のように、マージ候補は、動き情報のフルセットに対応するが、AMVP候補は、特定

40

50

の予測方向のための1つだけの動きベクトルを含む。マージモードとAMVPモードの両方に対する候補は、同じ空間的隣接ブロックおよび時間的隣接ブロックから同様に導出され得る。

【 0 0 5 7 】

空間的MV候補が、特定のPU(PU_0)に対して図2Aおよび図2Bに示される隣接ブロックから導出されるが、ブロックから候補を生成する方法は、マージモードとAMVPモードとで異なる。図2Aは、マージモードのための空間的隣接MV候補を示す。マージモードでは、最高で4つの空間的MV候補を、図2Aにおいて番号で示される順序で導出することができ、その順序は、左(0)、上(1)、右上(2)、左下(3)、および左上(4)である。

【 0 0 5 8 】

図2Bは、AMVPモードのための空間的隣接MV候補を示す。図2Bに示されるように、AMVPモードでは、隣接ブロックは2つのグループ、すなわちブロック0および1からなる左のグループと、ブロック2、3および4からなる上のグループに分割される。各グループに対して、シグナリングされた参照インデックスによって示される参照ピクチャと同じ参照ピクチャを参照する隣接ブロックの中の潜在的な候補は、グループの最終候補を形成するために選択される際の優先順位が最高である。たとえば、リスト0AMVP候補リストを生成するとの一部として、ビデオコーダは、ブロック0がリスト0から予測されるかどうかを確認し、そうである場合、ブロック0のリスト0参照ピクチャが現在のPUのリスト0参照ピクチャと同じであるかどうかを確認する。ブロック0がリスト0から予測され、ブロック0のリスト0参照ピクチャが現在のPUのリスト0参照ピクチャと同じである場合、ビデオコーダは、リスト0AMVP候補リストにブロック0のリスト0動きベクトルを含める。そうではない場合、ビデオコーダは、ブロック0がリスト1から予測されるかどうかを確認し、そうである場合、ブロック0のリスト1参照ピクチャが現在のPUのリスト0参照ピクチャと同じであるかどうかを確認する。ブロック0がリスト0から予測され、ブロック0のリスト1参照ピクチャが現在のPUのリスト0参照ピクチャと同じである場合、ビデオコーダは、リスト0AMVP候補リストにブロック0のリスト1動きベクトルを含める。ブロック0のリスト1参照ピクチャが現在のPUのリスト0参照ピクチャと同じではない場合、ビデオコーダはブロック0の代わりにブロック1についてこのプロセスを繰り返す。

【 0 0 5 9 】

しかしながら、ブロック1がリスト1から予測されない場合、またはブロック1のリスト1参照ピクチャが現在のPUのリスト0参照ピクチャと同じではない場合、ビデオコーダは、ブロック0がリスト0から予測されるかどうかを決定し、そうである場合、ブロック0のリスト0参照ピクチャと現在のPUのリスト0参照ピクチャがともに長期参照ピクチャであるか、またはともに短期ピクチャであるかを決定する。ブロック0のリスト0参照ピクチャおよび現在のPUのリスト0参照ピクチャがともに長期参照ピクチャである場合、またはブロック0のリスト0参照ピクチャおよび現在のPUのリスト0参照ピクチャがともに短期参照ピクチャである場合、ビデオコーダは、ブロック0のリスト0参照ピクチャと現在のPUのリスト0参照ピクチャとの間の時間的な差に基づいて、ブロック0のリスト0動きベクトルをスケーリングし得る。ビデオコーダは、スケーリングされたリスト0動きベクトルをリスト0AMVP候補リストへと含める。ブロック0のリスト0参照ピクチャが長期参照ピクチャであり、現在のPUのリスト0参照ピクチャが短期参照ピクチャである場合、またはその逆である場合、ビデオコーダは、ブロック0がリスト1から予測されるかどうかを決定し、そうである場合、ブロック0のリスト1参照ピクチャと現在のPUのリスト0参照ピクチャがともに長期参照ピクチャであるか、またはともに短期ピクチャであるかを決定する。ブロック0のリスト1参照ピクチャおよび現在のPUのリスト0参照ピクチャがともに長期参照ピクチャである場合、またはブロック0のリスト1参照ピクチャおよび現在のPUのリスト0参照ピクチャがともに短期参照ピクチャである場合、ビデオコーダは、ブロック0のリスト1参照ピクチャと現在のPUのリスト0参照ピクチャとの間の時間的な差に基づいて、ブロック0のリスト1動きベクトルをスケーリングし得る。ビデオコーダは、スケーリングされたリスト0動きベクトルをリスト0AMVP候補リストへと含める。ブロック0のリスト1参照ピクチャが長期

10

20

30

40

50

参照ピクチャであり、現在のPUのリスト0参照ピクチャが短期参照ピクチャである場合、またはその逆である場合、ビデオコーダはブロック0の代わりにブロック1についてこのプロセスを繰り返す。

【 0 0 6 0 】

ビデオコーダは、現在のPUのリスト0AMVP候補リストに第2の候補を含めるように、ブロック2、3、および4について同様のプロセスを実行し得る。加えて、ビデオコーダは、リスト0への言及をリスト1に変え、リスト1への言及をリスト0に変えた上で、このプロセス全体を繰り返して、現在のPUのリスト1AMVP候補リストを生成し得る。

【 0 0 6 1 】

したがって、図2Bに示されるように、AMVPモードでは、隣接ブロックは2つのグループ、すなわちブロック0および1からなる左のグループと、ブロック2、3および4からなる上のグループに分割される。各グループに対して、シグナリングされた参照インデックスによって示される参照ピクチャと同じ参照ピクチャを参照する隣接ブロックの中の潜在的な候補は、グループの最終候補を形成するために選択される際の優先順位が最高である。すべての隣接ブロックが、同じ参照ピクチャを指す動きベクトルを含まないことがあり得る。したがって、そのような候補を見出すことができない場合、第1の利用可能な候補が、最終候補を形成するようにスケーリングされ得るので、時間的距離の差が補償され得る。

【 0 0 6 2 】

ビデオコーダは、時間的動きベクトル予測子(TMVP)候補を、有効であり利用可能である場合、空間的動きベクトル候補の後にマージMV候補リストへ含め、またはAMVP候補リストへ含め得る。たとえば、AMVPの場合、ビデオコーダは、空間的隣接ブロックが利用可能ではない(たとえば、空間的隣接ブロックがピクチャ、スライス、またはタイルの境界の外側にあるので、空間的隣接ブロックがイントラ予測されるので、など)場合、AMVP候補リストにTMVP候補を含め得る。マージモードでは、TMVP候補は、時間的隣接ブロックのリスト0および/またはリスト1動きベクトルを指定し得る。マージモードにおけるTMVP候補のための参照インデックスは常に0に設定される。AMVPモードでは、TMVP候補は、時間的隣接ブロックのリスト0動きベクトルまたは時間的隣接ブロックのリスト1動きベクトルのいずれかを指定する。時間的隣接ブロックは参照ピクチャの中のブロックである。TMVP候補のための動きベクトル導出のプロセスは、マージモードとAMVPモードの両方に対して同じであり得る。

【 0 0 6 3 】

図3Aは、TMVP候補を導出するための例示的な技法を示す概念図である。図3Aに示されるように、TMVP候補導出のための主要なブロック位置は、併置された(collocated)PUの外側の右下のブロック300である。ビデオコーダがTMVP候補をそれから導出する時間的隣接ブロックは、右下のブロック300と併置される。図3Aの例では、右下のブロック300は「時間的」であることを示すためにブロック「T」としてマークされる。ビデオコーダは、空間的隣接候補を生成するために使用される上と左のブロックに対する偏りを補償するために、上または左のブロックの代わりに右下のブロック300を使用する。右下のブロック300が現在のCTB行の外側に位置する場合、または動き情報が利用可能ではない場合(たとえば、右下のブロック300と併置された時間的参照ブロックがイントラ予測されるので)、右下のブロック300はPUの中心のブロック302で置き換えられる。

【 0 0 6 4 】

TMVP候補の動きベクトルは、いわゆる「併置されたピクチャ」の併置されたPUから導出される。併置されたピクチャは、スライスレベルで(たとえば、`collocated_ref_idx`シンタックス要素を使用して)示され得る。併置されたPUの動きベクトルは併置されたMVと呼ばれる。H.264/AVCにおける時間的直接モードと同様に、TMVP候補動きベクトルを導出するために、併置されたMVは、図3Bに示されるように、時間的距離差分を補償するためにスケーリングされ得る。具体的には、図3Bにおいて、現在のピクチャ322の現在のブロック320をコーディングするとき、ビデオコーダは、併置されたピクチャ324の中の併置されたブロック323を決定する。併置されたブロック323の動きベクトル326(すなわち、併置され

10

20

30

40

50

た動きベクトル)は、併置された参照ピクチャ328の中の位置を示す。ビデオコーダは、併置された時間的距離と現在の時間的距離との差分に基づいて動きベクトル326をスケーリングすることによって、TMVP330を生成する。併置された時間的距離は、併置されたピクチャ324と併置された参照ピクチャ328との間の時間的距離である。現在の時間的距離は、現在のピクチャ322と現在の参照ピクチャ332との間の時間的距離である。

【0065】

上で述べられたように、ビデオコーダは動きベクトルをスケーリングし得る。動きベクトルをスケーリングするとき、動きベクトルの値は提示の時間におけるピクチャの距離に比例すると仮定される。動きベクトルは、2つのピクチャ、すなわち、参照ピクチャと、動きベクトルを含むピクチャ(すなわち、格納ピクチャ)とを関連付ける。他の動きベクトルを予測するためにある動きベクトルが使用されるとき、格納ピクチャと参照ピクチャの距離は、参照ピクチャおよび格納ピクチャのピクチャ順序カウント(POC)値に基づいて計算される。10

【0066】

予測されるべき動きベクトルに対して、その関連する格納ピクチャと参照ピクチャの両方が異なることがある。したがって、新しい距離(POCに基づく)が計算される。動きベクトルは、これらの2つのPOC距離に基づいてスケーリングされる。空間的隣接候補について、2つの動きベクトルの格納ピクチャは同じであるが、参照ピクチャは異なる。HEVCでは、動きベクトルのスケーリングは、空間的隣接候補および時間的隣接候補のためのTMVPとAMVPの両方に適用される。20

【0067】

さらに、いくつかの実装形態では、MV候補リスト(たとえば、マージMV候補リストまたはAMVP候補リスト)が完全ではない場合、ビデオコーダは、MV候補リストが必要とされる数の候補を有するまで、人工的な動きベクトル候補を生成してそれをMV候補リストの終わりに挿入することができる。マージモードでは、Bスライスのためだけに導出された合成候補およびゼロ候補という、2つのタイプの人工的なMV候補がある。合成候補は、1つのマージ候補からのリスト0動きベクトルと、異なるマージ候補のためのリスト1動きベクトルとの合成を指定する。ゼロ候補は、第1のタイプ(すなわち、合成候補)が十分な人工的な候補を提供しない場合にのみ、動きベクトル予測に使用される。ゼロ候補は、その水平成分と垂直成分が各々0に等しいMVを指定する候補である。30

【0068】

候補リストの中にすでにあり、必要な動き情報を有する候補の各ペアに対して、リスト0の中のピクチャを参照する第1の候補の動きベクトルと、リスト1の中のピクチャを参照する第2の候補の動きベクトルの合成によって、双方向合成動きベクトル候補が導出される。

【0069】

加えて、ビデオコーダは、候補の挿入のために剪定(pruning)プロセスを適用し得る。異なるブロックからの候補が偶然同じであることがあり、これは、マージ/AMVP候補リストの効率を下げることがある。この問題を解決するために剪定プロセスが適用される。剪定プロセスは、同一の候補を挿入するのをある程度避けるために、ある候補を現在の候補リストの中の他の候補と比較する。複雑さを下げるために、各々の可能性のある候補をすべての他の既存の候補と比較する代わりに、限られた数の剪定プロセスのみが適用される。40

【0070】

HEVCでは、変換動きモデルのみが、動き補償予測(MCP)のために適用される。しかしながら、現実の世界では、多くの種類の動き、たとえばズームイン/アウト、回転、遠近法の動き、および他の不規則な動きがある。JEMでは、簡略化されたアフィン変換動き補償予測が、コーディング効率を改善するために適用される。ブロックがアフィン動きモデルに従う場合、ブロックの中の場所(x,y)のMVは、以下のアフィン動きモデルによって決定され得る。50

【0071】

【数1】

$$\begin{cases} v_x = ax + by + c \\ v_y = dx + ey + f \end{cases} \quad (1)$$

【0072】

10

式(1)において、 v_x はブロック内の場所(x,y)のための動きベクトルの水平成分であり、 v_y はブロック内の場所(x,y)のための動きベクトルの垂直成分である。式(1)において、a、b、c、d、e、およびfはパラメータである。アフィン動きモデルでは、ブロック内の異なる場所が異なる動きベクトルを有することに留意されたい。

【0073】

JEM3.0において、アフィン動きモデルは、 $a=e$ かつ $b=-d$ を仮定することによって、4パラメータのアフィン動きモデルに簡略化される。したがって、式(1)は、以下の式(1')に示されるように簡略化され得る。

【0074】

【数2】

20

$$\begin{cases} v_x = ex + -dy + c \\ v_y = dx + ey + f \end{cases} \quad (1')$$

【0075】

30

4パラメータのアフィン動きモデルは、左上の制御点(V_0)の動きベクトルおよび右上の制御点(V_1)の動きベクトルによって表され得る。図4は、現在のブロック400のための簡略化されたアフィン動きモデルを示す。図4に示されるように、ブロックのアフィン動きフィールドは、2つの制御点動きベクトル

【0076】

【数3】

\widetilde{V}_0 および \widetilde{V}_1

40

【0077】

によって記述される。

【0078】

【数4】

\widetilde{V}_0

【0079】

は、現在のブロック400の左上の制御点402の制御点動きベクトルである。

50

【0080】

【数5】

$$\tilde{V}_1$$

【0081】

は、現在のブロック400の右上の制御点404の制御点動きベクトルである。

【0082】

ブロックの動きベクトルフィールド(MVF)は、以下の式によって記述される。

10

【0083】

【数6】

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w} x - \frac{(v_{1y} - v_{0y})}{w} y + v_{0x} \\ v_y = \frac{(v_{1y} - v_{0y})}{w} x + \frac{(v_{1x} - v_{0x})}{w} y + v_{0y} \end{cases} \quad (2)$$

20

【0084】

式(2)において、 v_x はブロックの中の場所(x, y)の動きベクトルの水平成分であり、 v_y はブロックの中の場所(x, y)の動きベクトルの垂直成分であり、 (v_{0x}, v_{0y}) は左上の角の制御点(たとえば、左上の制御点402)の動きベクトルであり、 (v_{1x}, v_{1y}) は右上の角の制御点(たとえば、右上の制御点404)の動きベクトルであり、 w はブロックの幅である。したがって、ビデオコーダは、式(2)を使用して、ブロックの制御点の動きベクトルに基づいて場所(x, y)の動きベクトルを「外挿する」ことができる。

【0085】

動き補償予測をさらに簡略化するために、ブロックベースのアフィン変換予測が適用される。したがって、ブロックの中の各位置の動きベクトルを導出するのではなく、ビデオコーダは、ブロックのサブブロックの動きベクトルを導出し得る。JEMでは、サブブロックは 4×4 のブロックである。サブブロックの動きベクトルを導出するために、ビデオコーダは、式(2)に従ってサブブロックの中心サンプルの動きベクトルを計算し得る。ビデオコーダは次いで、計算された動きベクトルを $1/16$ 分数精度に丸め得る。丸められた動きベクトルは、高精度動きベクトルと本明細書で呼ばれ得る。次いで、ビデオコーダは、動き補償補間フィルタを適用して、導出された動きベクトルを用いてサブブロックの各々の予測(すなわち、予測ブロック)を生成し得る。

30

【0086】

図5は、サブブロックごとの例示的なアフィン動きベクトルフィールド(MVF)を示す。図5の例に示されるように、現在のブロック500は、左上の制御点502および右上の制御点504を有する。ビデオコーダは、左上の制御点502の動きベクトル506および右上の制御点504の動きベクトル508に基づいて、現在のブロック500のサブブロックの動きベクトルを計算し得る。図5は、サブブロックの動きベクトルを小さい矢印として示す。

40

【0087】

MCPの後で、各サブブロックの高精度動きベクトルは、通常の動きベクトルと同じ精度に丸められて保存される。いくつかの例では、高精度動きベクトルの丸めは、記憶された動きベクトルの精度が高精度動きベクトル未満であるときだけ実行される。

【0088】

JEMには、AF_INTERモードおよびAF_MERGEモードという2つのアフィン動きモードがある

50

。JEMでは、AF_INTERモードは、幅と高さの両方が8を超えるCUのために適用され得る。AF_INTERモードが使用されるかどうかを示すために、ビットストリームの中のCUレベルにおいてアフィンフラグがシグナリングされる。AF_INTERモードでは、ビデオエンコーダ20は、リスト0参照ピクチャおよび/またはリスト1参照ピクチャを示すために、現在のプロックのためのリスト0参照インデックスおよび/またはリスト1参照インデックスをシグナリングする。

【0089】

AF_INTERモードでは、ビデオエンコーダ20およびビデオデコーダ30は各々、現在のプロックのための1つまたは複数の候補リスト(すなわち、アフィンMVPセット候補リスト)を構築する。たとえば、ビデオエンコーダ20およびビデオデコーダ30は各々、リスト0アフィンMVPセット候補リストおよび/またはリスト1アフィンMVPセット候補リストを構築し得る。アフィンMVPセット候補リストの各々は、アフィンMVPセットのそれぞれのセットを含む。4パラメータのアフィン動きモデルでは、リスト0アフィンMVPセット候補リストの中のアフィンMVPセットは、2つのリスト0動きベクトル(すなわち、動きベクトルペア)を指定する。4パラメータのアフィン動きモデルでは、リスト1アフィンMVPセット候補リストの中のアフィンMVPセットは、2つのリスト1動きベクトルを指定する。

【0090】

最初に、ビデオコーダ(たとえば、ビデオエンコーダ20またはビデオデコーダ30)は、隣接プロックを使用するタイプの動きベクトルペア $\{(v_0, v_1) | v_0 = \{v_A, v_B, v_C\}, v_1 = \{v_D, v_E\}\}$ でアフィンMVPセット候補リストを埋めることを試みる。図6Aは、AF_INTERモードにおいて使用されるような現在のプロック600および隣接プロックを示すプロック図である。図6Aに示されるように、 v_0 は、プロックA、B、またはCの動きベクトルから選択される。隣接プロックからの動きベクトルは、参照リストと、隣接プロックに対する参照のPOC、現在のCUに対する参照のPOC、および現在のCUのPOCの間の関係とに従ってスケーリングされる。たとえば、ビデオコーダが隣接プロック(たとえば、プロックA、B、またはC)のリスト0動きベクトルを v_0 として選択すると仮定する。この例では、隣接プロックのリスト0動きベクトルは、隣接プロックの参照ピクチャ(すなわち、隣接プロックに対する参照)の中の場所を示す。さらに、この例では、ビデオエンコーダ20は、現在のCUに対する参照ピクチャ(すなわち、現在のCUに対する参照)を示すリスト0参照インデックスを選択してシグナリングし得る。隣接プロックの参照ピクチャが現在のCUに対する参照ピクチャと同じではない場合、ビデオコーダは、参照の時間的距離と現在の時間的距離との間の差分に基づいて、隣接プロックのリスト0動きベクトルをスケーリングし得る。参照の時間的距離は、隣接プロックの参照ピクチャのPOCと現在のCUのPOCとの間の時間的距離である。現在の時間的距離は、現在のCUのPOCと現在のCUに対する参照ピクチャのPOCとの間の時間的距離である。ビデオコーダは、リスト1動きベクトルのために同様のプロセスを実行し得る。隣接プロックDおよびEから v_1 を選択するための手法は同様である。

【0091】

候補リストの中の候補の数が2より少ない場合、候補リストは、AMVP候補 $\{\text{AMVP}0, \text{AMVP}0\}$ および $\{\text{AMVP}1, \text{AMVP}1\}$ の各々を複製することによって組み立てられる動きベクトルペアによってパディングされる。言い換えると、ビデオコーダは、上で説明された方式で2つのAMVP候補を生成し得る。2つのAMVP候補はAMVP0およびAMVP1と表記される。ビデオコーダは次いで、図6Bの候補リスト620に、AMVP0を第1の制御点の動きベクトルとして指定しAMVP0を第2の制御点の動きベクトルとして指定する第1のアフィン動きベクトル予測子候補を含める。候補リストの中の候補の数が、候補リスト620に第1のアフィンMVP候補を含めた後でも2より少ない場合、ビデオコーダは候補リスト620に第2のアフィンMVP候補を含め、ここで、第2のアフィンMVP候補は、第1の制御点の動きベクトルとしてAMVP1を指定し、第2の制御点の動きベクトルとしてAMVP1を指定する。

【0092】

候補リスト620が2より大きいとき、ビデオコーダはまず、隣接動きベクトルの一貫性(候補ペアの中の2つの動きベクトルの類似性)に従って、候補リスト620の中の候補をソー

10

20

30

40

50

トする。ビデオコーダは、図6Bにおいて線の標識「サイズ=2」で示されるように、最初の2つの候補を保つ。ビデオエンコーダ20は、レート-ひずみコストの確認を使用して、どの動きベクトルセット候補が現在のCUの制御点動きベクトル予測(CPMVP)として選択されるかを決定し得る。ビデオエンコーダ20は、候補リスト620におけるCPMVPの場所を示すインデックスをピットストリームにおいてシグナリングし得る。ビデオデコーダ30は、ピットストリームからインデックスを取得し、インデックスを使用して候補リスト620の中の候補のいずれがCPMVPであるかを決定し得る。現在のアフィンCUのCPMVPが決定された後で、アフィン動き推定が適用され、制御点動きベクトル(CPMV)が発見される。ビデオエンコーダ20は、CPMVとCPMVPとの間の差分をピットストリームにおいてシグナリングする。言い換えると、ビデオエンコーダ20は、ピットストリームにおいて動きベクトル差分(MVD)をシグナリングする。10

【0093】

その上、HEVCとJEMの両方において、インターピクチャ予測シンタックス要素inter_predit_idxは、ブロック(たとえば、CUまたはPU)のためにリスト0が使用されるか、リスト1が使用されるか、または両方が使用されるかをシグナリングする。1つの参照ピクチャリストから取得される各MVPに対して、対応する参照ピクチャは参照ピクチャリストに対するインデックスref_idx_10/1によってシグナリングされ、MV(_x;_y)はMVPに対するインデックスmvp_10/1_flagおよびそのMVD差分(MVD)によって表される。MVDシンタックスはまた、MVDがデコーダ側で再構築され得るように、ピットストリームにおいてシグナリングされる。言い換えると、ブロックがリスト0から単方向予測され、または双方向予測される場合20、ビデオエンコーダ20は、リスト0の中の参照ピクチャの位置を示すためのref_idx_10_flagをシグナリングし、選択された動きベクトル予測子のリスト0AMVP候補リストにおける位置を示すためのmvp_10_flagをシグナリングし、リスト0MVDをシグナリングする。ブロックがリスト1から単方向予測され、または双方向予測される場合、ビデオエンコーダ20は、リスト1の中の参照ピクチャの位置を示すためのref_idx_11_flagをシグナリングし、選択された動きベクトル予測子のリスト1AMVP候補リストにおける位置を示すためのmvp_11_flagをシグナリングし、リスト1MVDをシグナリングする。

【0094】

加えて、ビデオエンコーダ20は、スライスヘッダにおいてフラグ(たとえば、mvd_11_zero_flag)をシグナリングし得る。このフラグは、第2の参照ピクチャリスト(たとえば、リスト1)に対するMVDが0に等しく、したがってピットストリームにおいてシグナリングされないかどうかを示す。第2の参照ピクチャリストに対するMVDをシグナリングしないことで、いくつかの状況においてコーディング効率をさらに改善することができる。30

【0095】

CUがAF_MERGEモードでコーディングされるとき、ビデオコーダは、訪問順序A B C D Eにおいて最初に生じる有効な隣接する再構築されたブロックからアフィンモードでコーディングされるブロックのアフィン動きモデルを、CUに割り当てる。図7Aは、AF_MERGEモードで現在のブロック700をコーディングするときに使用される隣接ブロックを示す。隣接ブロックに対する訪問順序(すなわち、選択順序)は、図7Aに示されるように左、上、右上、左下から左上である。たとえば、隣接ブロックBがアフィンモードを使用してコーディングされる順序A B C D Eにおいて最初の隣接ブロックである場合、ビデオコーダは、隣接ブロックBのアフィン動きモデルを現在のブロックのアフィン動きモデルとして使用し得る。たとえば、この例では、X=0および/またはX=1に対して、ビデオコーダは、隣接ブロックBの左上の制御点のリストX動きベクトルを外挿して現在のブロック700の左上の制御点のリストX動きベクトルを生成し、隣接ブロックBの左上の制御点のリストX参照インデックスを現在のブロック700の左上の制御点のリストX参照インデックスとして使用し、隣接ブロックBの右上の制御点のリストX動きベクトルを外挿して現在のブロック700の右上の制御点のリストX動きベクトルを生成し、隣接ブロックBの右上の制御点のリストX参照インデックスを現在のブロック700の右上の制御点のリストX参照インデックスとして使用し得る。この例では、ビデオコーダは、上の式(2)を使用して隣接ブロックBの4050

制御点の動きベクトルを外挿し、現在のブロック700の制御点の(x, y)の場所を式(2)における x および y として使用して、現在のブロック700の制御点の動きベクトルを決定し得る。

【 0 0 9 6 】

隣の左下のブロックAが図7Bにおいて示されるようにアフィンモードでコーディングされる場合、ブロックAを含む現在のブロック720の左上の角、右上の角、および左下の角の動きベクトル v_2 、 v_3 、および v_4 が導出される。現在のブロック720の左上の角の動きベクトル v_0 は、 v_2 、 v_3 、および v_4 に従って計算される。第2に、現在のCUの右上の動きベクトル v_1 が計算される。具体的には、6パラメータのアフィン動きモデルがまず、以下の式(3)に示されるように動きベクトル v_2 、 v_3 、および v_4 によって構築され、 v_0 および v_1 の値は次いで、この6パラメータのアフィン動きモデルによって計算される。すなわち、現在のブロック720(すなわち、

【 0 0 9 7 】

【 数 7 】

\widetilde{v}_0

【 0 0 9 8 】

)の左上の制御点の(x, y)の場所とともに使用されるとき、式(3)の中の v_x は、

20

【 0 0 9 9 】

【 数 8 】

\widetilde{v}_0

【 0 1 0 0 】

(すなわち、 v_{0x})の動きベクトルの x 成分であり、式(3)の中の x_y は、

【 0 1 0 1 】

【 数 9 】

30

\widetilde{v}_0

【 0 1 0 2 】

(すなわち、 v_{0y})の y 成分である。同様に、現在のブロック720(すなわち、

【 0 1 0 3 】

【 数 1 0 】

\widetilde{v}_1

40

【 0 1 0 4 】

)の右上の制御点の(x, y)の場所とともに使用されるとき、式(3)の中の v_x は、

【 0 1 0 5 】

【 数 1 1 】

\widetilde{v}_1

50

【0106】

(すなわち、 v_{1x})の動きベクトルのx成分であり、式(3)の中の x_y は、

【0107】

【数12】

 $\widetilde{v_1}$

【0108】

(すなわち、 v_{1y})のy成分である。

10

【0109】

【数13】

$$\begin{cases} v_x = \frac{(v_{3x} - v_{2x})}{w}x + \frac{(v_{4x} - v_{2x})}{h}y + v_{2x} \\ v_y = \frac{(v_{3y} - v_{2y})}{w}x + \frac{(v_{4y} - v_{2y})}{h}y + v_{2y} \end{cases} \quad (3)$$

20

【0110】

現在のCUの v_0 および v_1 のCPMVが導出された後で、式(1)の簡略化されたアフィン動きモデルに従って、現在のCUの動きベクトルフィールドが生成される。現在のCUがAF_MERGEモードでコーディングされるかどうかを特定するために、アフィンモードでコーディングされる少なくとも1つの隣接ブロックがあるとき、アフィンフラグがビットストリームにおいてシグナリングされる。

【0111】

JEMにおける4パラメータのアフィン動きモデルに加えて、6パラメータを伴うアフィン動きモデルがJVET-C0062において記述される。6パラメータのアフィンモデルでは、水平方向と垂直方向との間のスケーリング係数に制約はない。6パラメータのモデルを表すために、3つの角の動きベクトルが使用される。

30

【0112】

図8Aは、6パラメータのアフィン動きモデルにおいて使用される例示的なブロックを示す。図8Bは、6パラメータのアフィン動きモデルにおいて使用される例示的なアフィンMVPセット候補リスト820を示す。以下の6つの式は、3つの角における動きベクトル(図8Aに示されるような v_0 、 v_1 、および v_2)の水平(x)成分および垂直(y)成分を記述する。

【0113】

【数14】

$$\begin{cases} V0_x = c \\ V0_y = f \end{cases} \quad (4) \quad 10$$

$$\begin{cases} V1_x = a \times \text{幅} + c \\ V1_y = d \times \text{幅} + f \end{cases}$$

$$\begin{cases} V2_x = b \times \text{高さ} + c \\ V2_y = e \times \text{高さ} + f \end{cases}$$

【0114】

20

式(4)を解くことによって、6パラメータのアフィンモデルは、解を式(1)に代入することにより決定され得る。

【0115】

30

4パラメータのAF_INTERモードと同様に、6パラメータのAF_INTERモードに対する動きベクトルセット $\{(v_0, v_1, v_2) | v_0=\{v_A, v_B, v_c\}, v_1=\{v_D, v_E\}, v_2=\{v_F, v_G\}\}$ を伴う候補リストが、隣接ブロックを使用して構築される。したがって、図8Bの例では、アフィンMVPセット候補リスト820の中の各アフィンMVPセット(すなわち、各候補)は、3つの動きベクトルを含む。AF_INTERモードでは、ビデオコーダは、図8Bに示されるタイプの2つのアフィンMVPセット候補リストを生成し得る。アフィンMVPセット候補リストのうちの1つが、リスト0動きベクトルを指定するアフィンMVPセットを含む。他のアフィンMVPセット候補リストが、リスト1動きベクトルを指定するアフィンMVPセットを含む。

【0116】

HEVC/JEMにおけるアフィン動きの設計には以下の問題があり得る。たとえば、リスト0およびリスト1のアフィン動きの相関は、アフィンモードでコーディングされるブロックに対するMV予測のために利用されない。別の例では、現在のブロックおよび隣接ブロックのアフィン動きの相関は、アフィンモードでコーディングされるブロックに対するMV予測のために利用されない。さらに別の例では、リスト1ゼロMVDは、制御点の不正確なMVが原因で、アフィン動きモデルの性能を損なうことがある。本開示は、これらの問題を克服し、場合によってはコーディング効率を改善し得る技法を説明する。

【0117】

40

本開示のいくつかの例では、アフィン動きモデルは、式(1)において示されるような6パラメータの動きモデルである。JEM-3.0では、ブロックのためのアフィン動きモデルは、制御点($V0, V1$)の動きベクトルによって表される。JVET-C0062では、ブロックのためのアフィン動きモデルは、制御点($V0, V1, V2$)の動きベクトルによって表される。しかしながら、式(1)のパラメータa、b、c、d、e、fまたは簡略化された4つのパラメータをシグナリングすることによって、アフィン動きモデルを表すことが望ましいことがある。アフィン動きモデルはまた、式(5)としてさらに解釈されることが可能であり、ここで 0_x および 0_y は変換オフセットであり、 S_x および S_y はx方向とy方向におけるスケーリング比であり、 α および β は回転角である。

【0118】

50

【数15】

$$\begin{cases} v_x = s_x * \cos \theta_x * x - s_y * \sin \theta_y * y + O_x \\ v_y = s_x * \sin \theta_x * x + s_y * \cos \theta_y * y + O_y \end{cases} \quad (5)$$

【0119】

本開示は、アフィン動き予測の動きベクトル予測子(MVP)またはパラメータ予測を改善するためのいくつかの方法を提案する。ビデオコーダは、式(1)のa、b、c、d、e、fまたは式(5)の O_x 、 O_y 、 s_x 、 s_y 、 θ_x 、および θ_y の表現のためのパラメータ予測を実行し得ることに留意されたい。

10

【0120】

本開示のいくつかの技法が以下で列挙される。

1)リスト0とリスト1との間のアフィン動きベクトル予測およびアフィンパラメータ予測。

2)制御点間のアフィン動きベクトル予測およびパラメータのセット間のアフィンパラメータ予測。

3)隣接ブロックからのアフィン動きベクトル予測およびアフィンパラメータ予測。隣接ブロックは空間的隣接ブロックに限定されない。むしろ、いくつかの例では、時間的隣接ブロックが使用される。

20

4)サブブロックアフィン動き予測およびサブブロックアフィンパラメータ予測、各サブブロックは固有の制御点および/またはパラメータを有し得る。

5)追加のMVP候補の生成を精緻化する。

6)GPBスライスにおいてアフィンインター モードに対するゼロL1 MVDを無効にする。

【0121】

本開示の技法は個別に適用され得る。代わりに、技法の任意の組合せが適用され得る。本開示は、以下の技法の各々のさらなる詳細を詳述する。

30

【0122】

上で言及されたように、HEVCおよびJEMにおけるアフィン動きの設計の欠点の1つは、リスト0のアフィン動きとリスト1のアフィン動きとの間の相関が利用されないことである。言い換えると、既存の手法は、リスト0およびリスト1に対して独立にアフィン制御点をシグナリングする。リスト0のアフィン動きとリスト1のアフィン動きとの間の相関を活用することで、コーディング効率を上げる機会が生まれ得る。

【0123】

したがって、本開示の技法によれば、ビデオエンコーダ20およびビデオデコーダ30は、1つのインター予測方向においてアフィン動きモデルを使用して、別のインター予測方向におけるアフィン動きモデルのシグナリングを改善し得る。特定のインター予測方向におけるアフィン動きモデルは、インター予測方向に対応する特定の参照ピクチャリストの中の参照ピクチャにおける位置を指示する動きベクトルを指定する、アフィン動きモデルである。

40

【0124】

たとえば、リスト0のMVとリスト1のMVとの間の相関を利用するため、ビデオエンコーダ20およびビデオデコーダ30は、リスト0のMVをリスト1のMVのためのMV予測子として使用することができ、またはその逆を行うことができる。ビデオデコーダ30は、予測方向の指示に従って、リスト0からリスト1へのMV予測またはリスト1からリスト0へのMV予測を使用することを決め得る。予測方向の指示は、明示的にシグナリングされることがあり、または暗黙的に導出されることがある。

50

【 0 1 2 5 】

ビデオデコーダ30は、復号または再構築された情報を使用して、暗黙的な予測方向の指示を導出し得る。一例では、予測方向の指示は、各予測方向のアフィンMVPセット候補リスト(すなわち、MVPリスト)の中のMVPに依存する。1つの予測方向のMVPリストが、いくつかのより好ましくない方法から導出されたMVP(たとえば、スケーリングされた動きベクトル予測子、または局所照明補償モードでコーディングされたブロックから導出されたMVP)を含む場合、別の予測方向からのアフィンモデルが、現在の予測方向におけるアフィンモデルを予測するために使用される。代わりに、予測方向の指示は、どれだけの異なるMV予測子が各リストの中にあるかに依存し得る。

【 0 1 2 6 】

10

別の例では、1つの予測方向のMVPリストが、いくつかのより好ましい方法から導出されたMVP(たとえば、スケーリングされていない動きベクトル予測子、またはフレームレートアップコンバージョン(FRUC)モードでコーディングされたブロックからのMVP)を含む場合、現在の予測方向からのアフィンモデルが、他の予測方向におけるアフィンモデルを予測するために使用される。FRUCモードでは、ブロックの動き情報はシグナリングされず、デコーダ側で導出される。

【 0 1 2 7 】

一例では、リスト1に対するMVPセット候補リストの中に低優先度のMVPがある場合、予測方向の指示はリスト0からリスト1として設定され、MV予測プロセスにおいて、リスト0のMVがリスト1のMVのためのMVPとして使用される。そうではなく、リスト1に対するMVPセット候補リストの中に低優先度のMVPがない場合、予測方向の指示はリスト1からリスト0として設定され、MV予測プロセスにおいて、リスト1のMVがリスト0のMVのためのMVPとして使用される。

20

【 0 1 2 8 】

予測方向の指示が決定された後で、置き換えられるべきMVPセット候補(N個の候補を伴う)が、復号された情報または再構築情報に従ってデコーダ側で決定される。一例では、置き換えられるべきMVPセット候補は、少なくとも1つの低優先度のMVPを含む第1のMVPセット候補である。一例では、POC情報に従ってスケーリングされた空間的MVP、パディングされたAMVP MVP、および時間的MVPに対して、低い優先度が与えられ得る。

【 0 1 2 9 】

30

一例では、さらに、置き換えられるべきMVPセット候補がないとき、リスト0とリスト1との間でのアフィン動きベクトル予測は実行されない。いくつかの例では、置き換えられるべきMVPセット候補の数Nは1に設定される。一例では、予測方向の指示がリスト0からリスト1へのMV予測として設定される場合、リスト1の中の置き換えられるべきMVPセット候補のMVPが次いで、リスト0の中の対応する制御点のスケーリングされたMVによってそれぞれ置き換えられ、またはこの逆が行われる。置き換えられ、またはこの逆が行われる。

【 0 1 3 0 】

別の例では、予測方向の指示がリスト0からリスト1へのMV予測として設定される場合、リスト1の中の置き換えられるべきMVPセット候補の部分的なMVPのみが、リスト0の中の対応する制御点のスケーリングされたMVによってそれぞれ置き換えられ、またはこの逆が行われる。たとえば、第1のMVP(V0)のみが置き換えられる。

40

【 0 1 3 1 】

別の例では、予測方向の指示がリスト0からリスト1へのMV予測として設定される場合、リスト1の中の置き換えられるべきMVPセット候補の第1のMVP(V0)のみが、リスト0の中の対応する制御点のスケーリングされたMVによってそれぞれ置き換えられる。リスト1の中の置き換えられるべきMVPセット候補の第2のMVP(V1)は、リスト0の中の第1の制御点(V0)のスケーリングされたMVと、リスト0における第1の制御点のMVと第2の制御点のMVとの間の差分(V1-V0)とを足したものによって置き換えられる。6パラメータのアフィンモデルのための第3のMVP(V2)を置き換える手法は同様である。予測方向の指示がリスト1からリスト0へのMV予測として設定される場合、リスト0の中の置き換えられるべきMVPセット候補

50

の第1のMVP(V0)のみが、リスト1の中の対応する制御点のスケーリングされたMVによってそれぞれ置き換える。リスト0の中の置き換えるべきMVPセット候補の第2のMVP(V1)は、リスト1の中の第1の制御点(V0)のスケーリングされたMVと、リスト1における第1の制御点のMVと第2の制御点のMVとの間の差分(V1-V0)とを足したものによって置き換える。そして、6パラメータのアフィンモデルのための第3のMVP(V2)を置き換える手法は同様である。

【 0 1 3 2 】

いくつかの例では、制御点のMVは、他の制御点のMVのためのMVP候補として使用され得る。一例では、左上の制御点(V0)のMVは、右上の制御点(V1)のMVのためのMVP候補として使用される。同様に、6パラメータのアフィン動きモデルのもとでは、左上の制御点(V0)のMVは、右上の制御点(V1)のMVのためのMVP候補として使用される。その上、制御点間の動きベクトル予測は、選択的に適用され得る。代わりに、6パラメータのアフィンモデルのもとでは、左上の制御点(V0)の動きベクトルが右上の制御点(V1)(または左下の制御点(V2))の動きベクトルの予測子として使用され、V0およびV1のペア(またはV0およびV2のペア)が4パラメータのアフィン動きモデルを使用することによってV2(またはV1)のための動きベクトル予測子を導出するために使用される。一例では、第2のMVPセット候補のみが、制御点間のMV予測を適用される。10

【 0 1 3 3 】

別の例では、アフィンモデルは、1つの制御点の動きベクトルおよび追加のシグナリングされるモデルパラメータから導出され得る。シグナリングされるパラメータは、限定はされないが、制御点間またはアフィン回転角間などの動きベクトル差分を含む。一例では、現在のブロックの制御点の1つの動きベクトルが、回転角とともに、アフィン動きモードとしてコーディングされるブロックのためにシグナリングされる。各アフィンブロックに対して、アフィン動きモデルは、制御点の動きベクトルと回転角とを使用して構築される。20

【 0 1 3 4 】

本開示の特定の技法によれば、ビデオコーダは、現在のアフィンブロックの制御点のためのアフィン動きベクトル予測子として、隣接する外挿された動きベクトルを使用し得る。たとえば、現在のインターブロックに対して、ビデオコーダは、ソースアフィンブロックと名付けられる現在のインターブロックの隣接アフィンブロック(たとえば、AF_INTERモードまたはAF_MERGEモードで符号化される隣接ブロック)のうちの1つまたは複数の動きモデルを利用して、現在のインターブロック(すなわち、現在のアフィン動きモデル)のためのアフィン動きモデルを予測し得る。たとえば、現在のブロックの制御点のMVPは、隣接ブロックの制御点から外挿され得る。たとえば、現在のブロックの各々のそれぞれの制御点に対して、ビデオコーダは、上の式(2)を使用して、ソースアフィンブロックの制御点の動きベクトルから、現在のブロックのそれぞれの制御点に対するMVPを外挿し得る。ソースアフィンブロックは、1つまたは複数の空間的隣接ブロックまたは時間的隣接ブロックであり得る。

【 0 1 3 5 】

一例では、ソースアフィンブロックは、事前に定義された訪問順序(たとえば、A B C D EまたはB A D C Eまたは図7Aに示されるブロックの任意の他の訪問順序)に基づいて、有効な空間的隣接ブロックからアフィンモードでコーディングされる最初のブロックとして決定される。40

【 0 1 3 6 】

別の例では、ソースアフィンブロックは、事前に定義された訪問順序(たとえば、A B C D EまたはB A D C Eまたは図7Aおよび図7Bに示されるような任意の他の訪問順序)に基づいて、1つまたは複数の事前に定義された優先度セットに従って隣接ブロックからアフィンモードでコーディングされる最初のブロックとして決定される。優先度のいずれも満たさない隣接アフィンブロックは、利用可能ではないものと見なされる。

【 0 1 3 7 】

10

20

30

40

50

いくつかの例では、ソースアフィンプロックは、まず訪問順序に従って、次いで事前に定義された優先順序に従って決定される。たとえば、ソースアフィンプロックは、A (優先度1) B (優先度1) C (優先度1) D (優先度1) E (優先度1)
 A (優先度2) などの順序に従って決定され得る。この例では、ビデオコーダは次いで、プロックAが優先度セット1の中にあるかどうかを確認し、ない場合、ビデオコーダは、プロックBが優先度セット1の中にあるかどうかを確認し、ない場合、ビデオコーダは、プロックCが優先度セット1の中にあるかどうかを確認し、以下同様である。

【 0 1 3 8 】

別の例では、ビデオコーダは、まず事前に定義された優先順序に従って、次いで訪問順序に従ってソースアフィンプロックを決定し得る。たとえば、A (優先度1) A (優先度2) B (優先度1) B (優先度2) C (優先度1) C (優先度2) など。
 したがって、この例では、ビデオコーダは次いで、プロックAが優先度セット1の中にあるかどうかを確認し、ない場合、ビデオコーダは、プロックAが優先度セット2の中にあるかどうかを確認し、ない場合、ビデオコーダは、プロックBが優先度セット1の中にあるかどうかを確認し、以下同様である。

【 0 1 3 9 】

様々な例において、優先度セットは様々な方法で定義される。様々な例示的な優先度セットの定義が以下のように列挙される。以下の優先度セットの定義は、個別に適用され得る。代わりに、それらの任意の組合せが適用され得る。

【 0 1 4 0 】

第1の例示的な優先度セットは以下のように定義され、より小さい番号がより高い優先度を表す。

1. 隣接アフィンプロックのリストX参照ピクチャが現在のプロックのリストX参照ピクチャと同じ参照ピクチャである場合、隣接アフィンプロックは優先度セット1の中にある、リストXは現在評価されている現在のプロックの参照ピクチャリストであり、Xは0または1のいずれかである。

2. 隣接アフィンプロックのリストX参照ピクチャが現在のプロックのリストY参照ピクチャと同じ参照ピクチャである場合、隣接アフィンプロックは優先度セット2の中にある、リストYは現在評価されている現在のプロックの参照ピクチャリスト以外の参照ピクチャリストであり、Yは0または1のいずれかである。

【 0 1 4 1 】

別の例では、優先度セットは以下のように定義される。

1. 隣接アフィンプロックのリストX参照ピクチャが現在のプロックのリストX参照ピクチャと異なる場合、隣接アフィンプロックは優先度セット1の中にある、リストXは現在評価されている現在のプロックの参照ピクチャリストであり、Xは0または1のいずれかである。

2. 隣接アフィンプロックのリストY参照ピクチャが現在のプロックのリストY参照ピクチャと異なる場合、隣接アフィンプロックは優先度セット2の中にある、リストYは現在評価されている現在のプロックの参照ピクチャリスト以外の参照ピクチャリストであり、Yは0または1のいずれかである。

【 0 1 4 2 】

別の例では、優先度セットは以下のように定義される。

1. 隣接アフィンプロックのMVの差分が事前に定義された範囲内にある。
2. 隣接アフィンプロックのMVの差分が事前に定義された範囲内にない。

【 0 1 4 3 】

別の例では、優先度セットは以下のように定義される。この例および他の例では、より小さい番号がより高い優先度を表し得る。

1. 隣接アフィンプロックがAF_INTERモードでコーディングされる場合、その隣接アフィンプロックは優先度セット1の中にある。

2. 隣接アフィンプロックがAF_MERGEモードでコーディングされる場合、その隣接

10

20

30

40

50

アフィンブロックは優先度セット2の中にある。

【0144】

別の例では、隣接アフィンブロックの優先度は、隣接アフィンブロックが現在のアフィンブロックと同じインター予測方向を有するかどうかに依存する。別の例では、隣接アフィンブロックの優先度は、隣接アフィンブロックのサイズに依存する。たとえば、よりサイズの大きい隣接アフィンブロックがより高い優先度を有し得る。

【0145】

一例では、ビデオコーダは、リストXのためのソースアフィンブロックとして、現在のブロックと同じリストX参照ピクチャを有しB A D C Eという訪問順序において最初に生じる隣接アフィンブロックを選択する。この例では、隣接アフィンブロックが利用可能ではない(たとえば、隣接アフィンブロックのいずれもが現在のブロックと同じリストX参照ピクチャを有しない)場合、ビデオコーダは、ソースアフィンブロックとして、現在のブロックのリストX参照ピクチャをそのリストY参照ピクチャとして有しB A D C Eという訪問順序において最初に生じる隣接アフィンブロックを選択することができ、ここでXは0または1であり、Yは(1-X)である。10

【0146】

一例では、双予測アフィンモデルが現在のブロックにおいて使用される場合、リスト0およびリスト1のためのソースアフィンブロックは異なり得る。言い換えると、ビデオコーダは、現在のブロックのためのリスト0およびリスト1を評価するとき、異なるソースアフィンブロックを使用し得る。上で言及されたソースアフィンブロック選択プロセスは、各参照ピクチャリストのために個別に適用され得る。20

【0147】

ビデオコーダがソースアフィンブロックを選択した後で、ビデオコーダは、ソースアフィンブロックの制御点のMVを使用して、現在のブロックの制御点のためのMV予測子のセットを外挿する。たとえば、4パラメータのアフィン動きモデルでは、ビデオコーダは、ソースアフィンブロックの第1の制御点のリストX MVから、現在のブロックの第1の制御点のリストX MVを外挿し得る。加えて、この例では、ビデオコーダは、ソースアフィンブロックの第2の制御点のリストX MVから、現在のブロックの第2の制御点のリストX MVを外挿し得る。この例では、Xは0または1のいずれかであり、外挿されるリストX MVの得られるペアは、外挿された動きベクトル予測子(MVP)セットと名付けられ、 $\{V'_0, V'_1\}$ と表記され得る。本開示の他の箇所で説明されるように、ビデオコーダは式(2)を使用して補間を実行し得る。6パラメータのアフィン動きモデルでは、ビデオコーダはまた、ソースアフィンブロックの第3の制御点のリストX MVから、現在のブロックの第3の制御点のリストX MVを外挿し得る。現在のブロックの第3の制御点の外挿されたリストX MVは、動きベクトル予測子セットにも含まれることがあり、 $\{V'_0, V'_1, V'_2\}$ と表記されることがある。30

【0148】

ビデオコーダは次いで、外挿されたリストX MVPセット(たとえば、4パラメータのアフィン動きモデルに対して $\{V'_0, V'_1\}$ または6パラメータのアフィン動きモデルに対して $\{V'_0, V'_1, V'_2\}$)をリストXアフィンMVPセット候補リストへと挿入し得る。ビデオコーダが外挿されたMVPセットをリストXアフィンMVPセット候補リストへと挿入した後で、ビデオコーダは、従来のアフィンMVPセット候補をリストXアフィンMVPセット候補リストへと挿入する。従来のアフィンMVPセット候補は、本開示において提供される他の例に従って生成されるアフィンMVPセット候補であり得る。ビデオコーダは、補間されたMVPセットの後または前に、従来のMVPセット候補をリストXアフィンMVPセット候補リストへと挿入し得る。現在のブロックが双方向予測される場合、ビデオコーダはリストYのために同様のプロセスを実行することができ、Yは1-Xに等しい。40

【0149】

図9は、本開示の技法に従って、外挿されたMVPセット902を含む例示的なアフィンMVPセット候補リスト900を示す。図9の例では、補間されたMVPセット $\{V'_i, V'_j\}$ (902)は、従来のMVPセット候補が後に続くアフィンMVPセット候補リスト900の第1の場所へと挿入される50

。ビデオコーダは、候補リスト620(図6B)またはアフィンMVPセット候補リスト820(図8B)と同じ方式で、アフィンMVPセット候補リスト900の残りを構築し得る。

【0150】

一例では、外挿プロセスは、MV外挿を実行するために4パラメータのアフィン動きモデルが使用されるか6パラメータのアフィン動きモデルが使用されるかに応じて、式(1)または(2)を使用して上で説明されたAF_MERGEモードと同じであり得る。代わりに、他の外挿関数が適用され得る。たとえば、ビデオコーダは、双線形関数をアフィンソースブロックの制御点の動きベクトルに適用して外挿プロセスを実行し得る。

【0151】

いくつかの例では、ビデオコーダは、上で説明されたようにソースアフィンブロックを選択することに加えて第2のソースアフィンブロックを選択する。¹⁰ビデオコーダは、第1のソースアフィンブロックを選択した後でソースアフィンブロックを探し続けることによって、第2のソースアフィンブロックを選択し得る。ビデオコーダは、ソースアフィンブロックを選択するための上で説明された例のいずれかに従って探索を行い得る。ビデオコーダは、第2のソースアフィンブロックの制御点のMVを使用して、現在のブロックの制御点のための第2のMVPセットを外挿することができ、アフィンMVPセット候補リストに別のMVPセット候補として第2のMVPセットを挿入することができる。

【0152】

別の例では、ビデオコーダは、現在のブロックをコーディングするとき、2つ以上のソースアフィンブロックを選択する。この例では、ビデオコーダは、いくつかのMVを使用して現在のブロックの制御点のためのMV予測子のセットを導出し、または、ビデオコーダは、ソースアフィンブロックのすべての制御点を別のMVPセット候補として挿入する。²⁰

【0153】

本開示の技法によれば、アフィンインターモード(すなわち、AF_INTER)またはアフィンマージモード(すなわち、AF_MERGE)において、現在のブロックの各サブブロック(たとえば、 4×4 のブロック)のアフィン動きは、それ自体の隣接ブロックの外挿された動きから予測され、または直接継承され得る。一例では、隣接ブロックは、各サブブロックのための最も近い隣接アフィンブロックとして選択される。言い換えると、現在のブロックは、複数の等しいサイズのサブブロック(たとえば、 4×4 のサブブロック)へと区分され得る。複数のサブブロックの各々のそれぞれのサブブロックに対して、ビデオコーダは、アフィン動きモデルを使用して予測された最も近いブロックを決定し得る。³⁰それぞれのサブブロックが現在のブロックの境界に沿っている場合、アフィン動きモデルを使用して予測された最も近いブロックは、現在のブロックの外側にあり得る。たとえば、現在のブロックの左上のサブブロックに対して、アフィン動きモデルを使用して予測された最も近いブロックは、左上のサブブロックの上および左のブロックであり得る。同様に、現在のブロックの右上のサブブロックに対して、アフィン動きモデルを使用して予測された最も近いブロックは、右上のサブブロックの上のブロックまたは右上のサブブロックの右上のブロックであり得る。現在のブロックの内部のサブブロックサブブロックに対して、アフィン動きモデルを使用して予測された最も近いブロックは、サブブロックの上または左の現在のブロックの別のサブブロックであり得る。⁴⁰

【0154】

この手法は、サブブロックがJEM3.0において使用される方法と異なる。上で論じられたように、JEM3.0では、ビデオコーダは、現在のブロックの左上および右上の角にある制御点の動きベクトルだけに基づいて、現在のブロックの各サブブロックの動きベクトルを計算する。対照的に、本開示のこの技法によれば、サブブロックの動きベクトルは、制御点の動きベクトルに基づいて計算されず、むしろ、隣接ブロックの外挿された動きから予測され、または直接継承される。これにより、サブブロックの動きベクトルをサブブロックにより近い位置からとることができるので、コーディング効率がより高くなり得る。

【0155】

図10は、現在のブロック1000の各サブブロック(たとえば、 4×4 のブロック)のアフィン

動きがそれ自体の隣接ブロックの外挿された動きから予測され、または直接継承され得る、本開示の技法に従ったサブブロック動き予測またはパラメータ予測を示す。図10の例に示されるように、TLサブブロック1002のアフィン動きは隣接ブロックB2の外挿された動きを使用して予測される(1004)が、TRサブブロック1006は隣接ブロックB1の外挿された動きを使用する。別の例では、サブブロックはまた、時間的隣接ブロックの外挿された動きを使用することができる。

【0156】

アフィンインターモードまたはアフィンマージモードでは、現在のブロックのアフィンパラメータ(たとえば、各サブブロック(たとえば、 4×4 のブロック)の式(1)または(4)におけるa、b、c、d、e、およびf)は、それ自体の隣接ブロックから予測され、または直接継承され得る。一例では、隣接ブロックは、各サブブロックのための最も近い隣接アフィンブロックとして選択される。たとえば、図10に示されるように、TLサブブロック1002のアフィンパラメータは隣接ブロックB2を使用して予測される(1004)が、TRサブブロック1006は隣接ブロックB1を使用する(1008)。10

【0157】

JEM3.0におけるアフィン動き予測の現在の設計では、現在のブロックのアフィンMVPセット候補リストのサイズが2より小さい場合、ビデオコーダは、AMVP候補の各々を複製することによって合成される1つまたは複数のアフィンMVPセットを、アフィンMVPセット候補リストにパディングする。本開示は、AMVP候補を複製することによって合成されるアフィンMVPセットを指して、「AMVPで導出された候補」という用語を使用することがある。20
しかしながら、制御点V0(図6A)のリストX動きベクトルおよび制御点V1のリストX動きベクトルが4パラメータのアフィン動きモデルにおいて同じである場合、または、制御点V0、V1、およびV2(図8A)のリストX動きベクトルが6パラメータのアフィン動きモデルにおいて同じである場合、現在のブロックの各サブブロックのための計算されたMVは同一である。現在のブロックの各サブブロックのための計算されたMVが同一であるとき、その結果は従来の動き補償された予測と同じである。しかしながら、アフィン動き予測を使用して同じ動き補償された予測結果に達すると、従来の動き補償された予測よりもコーディング効率が低くなり得る。したがって、従来の動き補償された予測と同じ結果をもたらす、AMVPで導出された候補をアフィンMVPセット候補リストに含めることは、従来の動き補償された予測より高いコーディング効率をもたらし得るアフィンMVPセットをアフィンMVPセット候補リストへ含めるための機会の喪失となる。30

【0158】

したがって、本開示の技法によれば、ビデオコーダは、AMVPで導出された候補内のすべてのMVPが同一とは限らないことを確実にするために、1つまたは複数のMVPに対するオフセットをAMVPで導出された候補へ加算する。アフィンMVPセット候補リストに多様なアフィンMVPセットを含めることで、アフィンMVPセット候補リストにおけるアフィンMVPセットのうちの1つの使用がより高いコーディング効率をもたらす確率を高めることができる。。

【0159】

図11Aは、本開示の技法に従って、4パラメータのアフィン動きモデルのための例示的なアフィンMVPセット候補リストを示す。図11Bは、本開示の技法に従って、6パラメータのアフィン動きモデルのための例示的なアフィンMVPセット候補リストを示す。図11Aの例に示されるように、4パラメータのアフィン動きモデルのもとで、ビデオコーダは、制御点V1のためのMVPにオフセットを加算する。図11Bの例に示されるように、6パラメータのアフィン動きモデルのもとで、ビデオコーダは、制御点V2のためのMVPにオフセットを加算する。いくつかの例では、オフセットは、アフィンMVPセット候補リストの中の異なる場所にある、AMVPで導出された候補に対しては異なる。たとえば、ビデオコーダは、第1のAMVPで導出された候補のためのオフセットとして+4を使用することがあり、第2のAMVPで導出された候補のためのオフセットとして-4を使用することがある。40

【0160】

図11Aの例では、ビデオコーダは、現在のブロックのためのリストX候補リスト1100(すなわち、アフィンMVPセット)を生成し、ここでXは0または1である。候補リスト1100を生成するために、ビデオコーダは潜在的な候補を確認する。潜在的な候補の各々は、ブロック V_A 、 V_B 、および V_C (図6A)から選択されたブロックのリストX動きベクトルと、 V_D および V_E (図6A)から選択されたブロックのリストX動きベクトルとの組合せである。潜在的な候補の中の両方のブロックがリストX動きベクトルを指定する場合、ビデオコーダは、その潜在的な候補を候補リスト1100に候補として含める。ビデオコーダは、候補リスト1100が2つの候補を含んだ後で、候補の追加を止める。

【0161】

潜在的な候補のすべてを確認した後で、候補リスト1100の中にまだ2つ未満の候補しかない場合、ビデオコーダは、第1のAMVPで導出される候補1102を候補リスト1100に追加し得る。第1のAMVPで導出される候補1102は、第1のAMVPで導出される動きベクトル予測子1104および第2のAMVPで導出される動きベクトル予測子1106を指定する。第1のAMVPで導出される動きベクトル予測子1104は、現在のブロックの第1の制御点の動きベクトル予測子である。第2のAMVPで導出される動きベクトル予測子1106は、現在のブロックの第2の制御点の動きベクトル予測子である。ビデオコーダは、ビデオコーダがAMVPにおいて第1の動きベクトル候補を導出するために使用するのと同じ方式で、第1のAMVPで導出される動きベクトル予測子1104を導出する。本開示は、上記のAMVPにおいて動きベクトル候補を導出するための技法を説明する。第2のAMVPで導出される動きベクトル予測子1106は、第1のAMVPで導出される動きベクトル予測子1104と、第1のAMVPで導出される動きベクトル予測子1104の水平成分または垂直成分の少なくとも1つに対する第1のオフセット(すなわち、 $Offset_0$)とを足したものに等しい。

【0162】

第1のAMVPで導出される候補1102を候補リスト1100に追加した後で、候補リスト1100の中にまだ2つ未満の候補しかない場合、ビデオコーダは、第2のAMVPで導出された候補1108を候補リスト1100に追加する。第2のAMVPで導出される動き候補1108は、第3のAMVPで導出される動きベクトル予測子1110および第4のAMVPで導出される動きベクトル予測子1112を指定する。ビデオコーダは、ビデオコーダがAMVPにおいて第2の動きベクトル候補を導出するために使用するのと同じ方式で、第3のAMVPで導出される動きベクトル予測子1110を導出する。第4のAMVPで導出される動きベクトル予測子1112は、ビデオコーダが第3のAMVPで導出される動きベクトル予測子1110の水平成分または垂直成分の少なくとも1つに対する第2のオフセット(すなわち、 $Offset_1$)を加算することを除き、第3のAMVPで導出される動きベクトル1110と同じである。第1のAMVPで導出される候補1102を候補リスト1100に追加した後で、候補リスト1100の中に2つの候補がある場合、ビデオコーダは、第2のAMVPで導出された候補1108を候補リスト1100に追加しない。現在のブロックが双方向予測される場合、ビデオコーダはリストY動きベクトルに関して候補リスト1100を生成する上で説明されたプロセスを繰り返すことができ、Yは1-Xに等しい。

【0163】

図11Bの例は、6パラメータのアフィン動きモデルを使用することを除き、同様のプロセスを示す。したがって、候補リスト1120を生成するために、ビデオコーダは潜在的な候補を確認する。潜在的な候補の各々は、ブロック V_A 、 V_B 、および V_C (図8A)から選択されたブロックのリストX動きベクトルと、 V_D および V_E (図8A)から選択されたブロックのリストX動きベクトルとの組合せである。組合せの各々を確認した後で、候補リスト1120の中にまだ2つ未満の候補しかない場合、ビデオコーダは、第1のAMVPで導出される候補1122を候補リスト1120に追加し得る。第1のAMVPで導出される候補1122は、第1のAMVPで導出される動きベクトル予測子1124(図11BにおいてAMVP₀と表記される)、第2のAMVPで導出される動きベクトル予測子1126、および第3のAMVPで導出される動きベクトル予測子1128を指定する。第1のAMVPで導出される動きベクトル予測子1124は現在のブロックの第1の制御点の動きベクトル予測子であり、第2のAMVPで導出される動きベクトル予測子1126は現在のブロックの第2の制御点の動

10

20

30

40

50

きベクトル予測子であり、第3のAMVPで導出される動きベクトル予測子1128は現在のブロックの第3の制御点の動きベクトル予測子である。ビデオコーダは、ビデオコーダがAMVPにおいて第1の動きベクトル候補を導出するために使用するのと同じ方式で、第1のAMVPで導出される動きベクトル予測子1124を導出する。第2のAMVPで導出される動きベクトル予測子1128は、第1のAMVPで導出される動きベクトル予測子1126に等しい。第3のAMVPで導出される動きベクトル予測子1128は、第1のAMVPで導出される動きベクトル予測子と、第1のAMVPで導出される動きベクトル予測子1124の水平成分または垂直成分の少なくとも1つに対する第1のオフセット(すなわち、 $Offset_0$)とを足したものに等しい。

【 0 1 6 4 】

第1のAMVPで導出される候補1122を候補リスト1120に追加した後で、候補リスト1120の中にまだ2つ未満の候補しかない場合、ビデオコーダは、第2のAMVPで導出された候補1130を候補リスト1120に追加する。第2のAMVPで導出される候補1130は、第4のAMVPで導出される動きベクトル予測子1132(図11BにおいてAMVP₁と表記される)、第5のAMVPで導出される動きベクトル予測子1134、および第6のAMVPで導出される動きベクトル予測子1136を指定する。ビデオコーダは、ビデオコーダがAMVPにおいて第2の動きベクトル候補を導出するために使用するのと同じ方式で、第4のAMVPで導出される動きベクトル予測子1132を導出する。第5のAMVPで導出される動きベクトル予測子1134は、第4のAMVPで導出される動きベクトル予測子1132と同じである。第6のAMVPで導出される動きベクトル予測子1136は、第3のAMVPで導出される動きベクトル予測子1132と、第3のAMVPで導出される動きベクトル予測子1132の水平成分または垂直成分の少なくとも1つに対する第2のオフセット(すなわち、 $Offset_1$)とを足したものに等しい。第1のAMVPで導出される候補1122を候補リスト1120に追加した後で、候補リスト1120の中に2つの候補がある場合、ビデオコーダは、第2のAMVPで導出された候補1130を候補リスト1120に追加しない。現在のブロックが双方向予測される場合、ビデオコーダはリストYに関して候補リスト1120を生成するための上で説明されたプロセスを繰り返すことができ、Yは1-Xに等しい。

【 0 1 6 5 】

いくつかの例では、HEVCにおけるMVP導出のために使用される隣接ブロック以外のブロックからの動きベクトル予測は、候補リストに追加され得る。いくつかの例では、ビデオコーダはアフィン動きのためのグローバルMVPをオンザフライで更新し、ビデオコーダは、候補リストのサイズが2より小さいとき、アフィン動きのためのグローバルMVPを使用する。たとえば、ビデオコーダは、利用可能なアフィンブロックを使用してグローバルアフィン動きモデルを構築することができ、ビデオコーダがアフィンブロックを再構築するときは常に、グローバルアフィン動きモデルを更新することができる。ビデオコーダは次いで、このグローバルアフィン動きモデルを使用して、後続のアフィンブロックのためのグローバルMVPを生成することができる。

【 0 1 6 6 】

HEVCおよびJEM3.0において、スライスヘッダの中のフラグmvd_l1_zero_flagは、第2の参照ピクチャリスト(たとえば、リスト1)のためのMVDが0に等しく、したがってコーディング効率をさらに改善するためにビットストリームにおいてシグナリングされないかどうかを示す。言い換えると、スライスのスライスヘッダの中の単一のフラグが、スライスのすべてのブロックのためのすべてのリスト1 MVDが0に等しいことを示し得る。このフラグを使用すると、スライスの各々のAMVPコーディングされたブロックまたはAF_INTRAコーディングされたブロックに対して0に等しいリスト1 MVDを別々にシグナリングする必要をなくすことによって、コーディング効率を高めることができる。

【 0 1 6 7 】

しかしながら、本開示の技法によれば、mvd_l1_zero_flagはいくつかのコーディングモードに適用可能であることがあり、他のモードでは、このフラグは、MVDが0に等しいことを示す場合であっても無視される。一例では、このゼロMVD設計はアフィン動きモードでは無効であるが、このゼロMVD設計は従来のインターモード(AMVPモード)では依然として保たれる。言い換えると、スライスのmvd_l1_zero_flagが、スライスの中のすべてのリス

10

20

30

40

50

ト1 MVDが0に等しいことを示す場合であっても、ビデオエンコーダ20はまだ、アフィン動きモードを使用して符号化されるスライスのブロックのためにリスト1 MVDをシグナリングすることができる。リスト1 MVDが0に等しいことをmvd_l1_zero_flagが示していても、アフィン動きモードを使用して符号化されるブロックのためにリスト1 MVDを依然としてシグナリングできることで、ビデオエンコーダ20は、アフィン動きモードを使用して符号化されないブロックのためにリスト1 MVDをシグナリングするのを回避することが可能でありながら、依然として、アフィン動きモードを使用して符号化されるブロックのためにリスト1 MVDをシグナリングすることが可能であり得る。これはコーディング効率の上昇をもたらし得る。本開示は、図19Aおよび図19Bに関する以下の例示的な技法に従った例示的な動作を記述する。

10

【0168】

図12は、本開示の技法を実施することができる例示的なビデオエンコーダ20を示すブロック図である。図12は説明のために提供され、広く例示されるとともに本開示において説明されるような技法の限定と見なされるべきでない。本開示の技法は、様々なコーディング規格または方法に適用可能であり得る。

【0169】

図12の例では、ビデオエンコーダ20は、予測処理ユニット1200、ビデオデータメモリ1201、残差生成ユニット1202、変換処理ユニット1204、量子化ユニット1206、逆量子化ユニット1208、逆変換処理ユニット1210、再構築ユニット1212、フィルタユニット1214、復号ピクチャバッファ1216、およびエントロピー符号化ユニット1218を含む。予測処理ユニット1200は、インター予測処理ユニット1220およびイントラ予測処理ユニット1222を含む。インター予測処理ユニット1220は、動き推定ユニットおよび動き補償ユニット(図示せず)を含み得る。

20

【0170】

ビデオデータメモリ1201は、ビデオエンコーダ20の構成要素によって符号化されるべきビデオデータを記憶するように構成され得る。ビデオデータメモリ1201の中に記憶されるビデオデータは、たとえば、ビデオソース18から取得され得る。復号ピクチャバッファ1216は、たとえば、イントラコーディングモードまたはインターコーディングモードにおいて、ビデオエンコーダ20によってビデオデータを符号化する際に使用するための参照ビデオデータを記憶する参照ピクチャメモリであり得る。ビデオデータメモリ1201および復号ピクチャバッファ1216は、シンクロナスDRAM(SDRAM)を含むダイナミックランダムアクセスメモリ(DRAM)、磁気抵抗RAM(MRAM)、抵抗性RAM(RRAM(登録商標))、または他のタイプのメモリデバイスなどの、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ1201および復号ピクチャバッファ1216は、同じメモリデバイスまたは別個のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ1201は、ビデオエンコーダ20の他の構成要素とともにオンチップであってよく、またはそれらの構成要素に対してオフチップであってもよい。ビデオデータメモリ1201は、図1の記憶媒体19と同じであるか、またはその一部であり得る。

30

【0171】

ビデオエンコーダ20は、ビデオデータを受信する。ビデオエンコーダ20は、ビデオデータのピクチャのスライスの中の各CTUを符号化し得る。CTUの各々は、ピクチャの、等しいサイズの輝度コーディングツリーブロック(CTB)、および対応するCTBと関連付けられ得る。CTUを符号化することの一部として、予測処理ユニット1200は、区分を実行して、CTUのCTBを次第に小さくなるブロックに分割し得る。より小さいブロックは、CUのコーディングブロックであり得る。たとえば、予測処理ユニット1200は、CTUと関連付けられたCTBを木構造に従って区分し得る。

40

【0172】

ビデオエンコーダ20は、CTUのCUを符号化して、CUの符号化された表現(すなわち、コーディングされたCU)を生成し得る。CUを符号化することの一部として、予測処理ユニット1200は、CUの1つまたは複数のPUの間でCUと関連付けられたコーディングブロックを区分し

50

得る。したがって、各PUは、輝度予測ブロックおよび対応する色差予測ブロックと関連付けられ得る。ビデオエンコーダ20およびビデオデコーダ30は、様々なサイズを有するPUをサポートし得る。上で示されたように、CUのサイズは、CUの輝度コーディングブロックのサイズを指すことがあり、PUのサイズは、PUの輝度予測ブロックのサイズを指すことがある。特定のCUのサイズが $2N \times 2N$ であると仮定すると、ビデオエンコーダ20およびビデオデコーダ30は、イントラ予測に対して $2N \times 2N$ または $N \times N$ というPUサイズ、およびインター予測に対して $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、 $N \times N$ 、または類似の、対称のPUサイズをサポートし得る。ビデオエンコーダ20およびビデオデコーダ30はまた、インター予測に対して、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、および $nR \times 2N$ というPUサイズのための非対称区分をサポートし得る。

【0173】

10

インター予測処理ユニット1220は、PUの予測データを生成し得る。PUの予測データを生成することの一部として、インター予測処理ユニット1220は、PUに対するインター予測を実行する。PUに対する予測データは、PUの予測ブロックおよびPUのための動き情報を含み得る。インター予測処理ユニット1220は、PUがIスライスの中にあるか、Pスライスの中にあるか、またはBスライスの中にあるかに応じて、CUのPUに対して異なる動作を実行し得る。Iスライスでは、すべてのPUがイントラ予測される。したがって、PUがIスライスの中にある場合、インター予測処理ユニット1220は、PUに対してインター予測を実行しない。したがって、Iモードで符号化されるブロックの場合、予測されるブロックは、同じフレーム内の以前に符号化された隣接ブロックから空間予測を使用して形成される。PUがPスライスの中にある場合、インター予測処理ユニット1220は、単方向インター予測を使用してPUの予測ブロックを生成し得る。PUがBスライスの中にある場合、インター予測処理ユニット1220は、単方向または双方向インター予測を使用してPUの予測ブロックを生成し得る。

【0174】

20

インター予測処理ユニット1220は、本開示の他の箇所で説明されたように、アフィン動きモデルのための技法を適用し得る。たとえば、インター予測処理ユニット1220はソースアフィンブロックを選択することができ、ソースアフィンブロックは現在のブロックに空間的に隣接するアフィンコーディングされたブロックである。この例では、インター予測処理ユニット1220は、ソースアフィンブロックの制御点の動きベクトルを外挿して、現在のブロックの制御点の動きベクトル予測子を決定し得る。さらに、この例では、インター予測処理ユニット1220は、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを、アフィンMVPセット候補リストに挿入し得る。この例では、インター予測処理ユニット1220は、アフィンMVPセット候補リストの中のアフィンMVPセットを選択し得る。加えて、この例では、インター予測処理ユニット1220は、ビットストリームにおいて、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示すMVDをシグナリングし得る。インター予測処理ユニット1220はまた、ビットストリームにおいて、選択されたアフィンMVPセットのアフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングし得る。

30

【0175】

40

イントラ予測処理ユニット1222は、PUに対してイントラ予測を実行することによって、PUの予測データを生成し得る。PUの予測データは、PUの予測ブロックおよび様々なシンタクス要素を含み得る。イントラ予測処理ユニット1222は、Iスライス、Pスライス、およびBスライスの中のPUに対して、イントラ予測を実行し得る。

【0176】

イントラ予測をPUに対して実行するために、イントラ予測処理ユニット1222は、複数のイントラ予測モードを使用して、PUの予測データの複数のセットを生成し得る。イントラ予測処理ユニット1222は、隣接PUのサンプルブロックからのサンプルを使用して、PUの予測ブロックを生成し得る。PU、CU、およびCTUに対して左から右、上から下への符号化順序を仮定すると、隣接PUは、PUの上、右上、左上、または左であり得る。イントラ予測処理ユニット1222は、様々な数のイントラ予測モード、たとえば、33個の方向性イントラ予

50

測モードを使用し得る。いくつかの例では、イントラ予測モードの数は、PUと関連付けられる領域のサイズに依存し得る。

【0177】

予測処理ユニット1200は、PUに対してインター予測処理ユニット1220によって生成された予測データ、またはPUに対してイントラ予測処理ユニット1222によって生成された予測データの中から、CUのPUの予測データを選択し得る。いくつかの例では、予測処理ユニット1200は、予測データのセットのレート/ひずみの尺度に基づいて、CUのPUの予測データを選択する。選択される予測データの予測ブロックは、本明細書で選択予測ブロックと呼ばれることがある。

【0178】

残差生成ユニット1202は、CUのコーディングブロック(たとえば、輝度コーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)およびCUのPUの選択された予測ブロック(たとえば、予測輝度ブロック、予測Cbブロック、および予測Crブロック)に基づいて、CUの残差ブロック(たとえば、輝度残差ブロック、Cb残差ブロック、およびCr残差ブロック)を生成し得る。たとえば、残差生成ユニット1202は、残差ブロックの中の各サンプルがCUのコーディングブロックの中のサンプルとCUのPUの対応する選択される予測ブロックの中の対応するサンプルとの間の差分に等しい値を有するような、CUの残差ブロックを生成し得る。

【0179】

変換処理ユニット1204は、四分木区分を実行して、CUの残差ブロックをCUのTUの変換ブロックに区分し得る。たとえば、変換処理ユニット1204は、四分木区分を実行して、CUの残差ブロックをCUのTUの変換ブロックに区分し得る。したがって、TUは、輝度変換ブロックおよび2つの色差変換ブロックと関連付けられ得る。CUのTUの輝度変換ブロックおよび色差変換ブロックのサイズおよび位置は、CUのPUの予測ブロックのサイズおよび位置に基づいても基づかなくてもよい。「残差四分木」(RQT)と呼ばれる四分木構造が、領域の各々と関連付けられたノードを含み得る。CUのTUは、RQTのリーフノードに相当し得る。

【0180】

変換処理ユニット1204は、TUの変換ブロックに1つまたは複数の変換を適用することによって、CUのTUごとに変換係数ブロックを生成し得る。変換処理ユニット1204は、TUと関連付けられる変換ブロックに様々な変換を適用し得る。たとえば、変換処理ユニット1204は、離散コサイン変換(DCT)、方向変換、または概念的に類似の変換を、変換ブロックに適用し得る。いくつかの例では、変換処理ユニット1204は、変換ブロックに変換を適用しない。そのような例では、変換ブロックは、変換係数ブロックとして扱われ得る。

【0181】

量子化ユニット1206は、係数ブロックの中の変換係数を量子化し得る。量子化プロセスは、変換係数の一部またはすべてと関連付けられるビット深度を低減し得る。たとえば、nビットの変換係数は、量子化の間にmビットの変換係数に切り捨てられてよく、nはmよりも大きい。量子化ユニット1206は、CUと関連付けられる量子化パラメータ(QP)値に基づいて、CUのTUと関連付けられる係数ブロックを量子化し得る。ビデオエンコーダ20は、CUと関連付けられるQP値を調整することによって、CUと関連付けられた係数ブロックに適用される量子化の程度を調整し得る。量子化は、情報の喪失をもたらし得る。したがって、量子化された変換係数は、元の精度よりも精度が低いことがある。

【0182】

逆量子化ユニット1208および逆変換処理ユニット1210は、それぞれ、逆量子化および逆変換を係数ブロックに適用して、係数ブロックから残差ブロックを再構築し得る。再構築ユニット1212は、予測処理ユニット1200によって生成された1つまたは複数の予測ブロックからの対応するサンプルに、再構築された残差ブロックを加算して、TUと関連付けられる再構築された変換ブロックを生成し得る。このようにしてCUのTUごとに変換ブロックを再構築することによって、ビデオエンコーダ20は、CUのコーディングブロックを再構築し得る。

【0183】

フィルタユニット1214は、1つまたは複数のデブロッキング動作を実行して、CUと関連付けられるコーディングブロックにおけるブロッキングアーティファクトを低減し得る。フィルタユニット1214が1つまたは複数のデブロッキング動作を再構築されたコーディングブロックに対して実行した後、復号ピクチャバッファ1216は、再構築されたコーディングブロックを記憶し得る。インター予測処理ユニット1220は、インター予測を他のピクチャのPUに対して実行するために、再構築されたコーディングブロックを含む参照ピクチャを使用し得る。加えて、イントラ予測処理ユニット1222は、CUと同じピクチャの中の他のPUに対してイントラ予測を実行するために、復号ピクチャバッファ1216の中の再構築されたコーディングブロックを使用し得る。

10

【0184】

エントロピー符号化ユニット1218は、ビデオエンコーダ20の他の機能構成要素からデータを受け取り得る。たとえば、エントロピー符号化ユニット1218は、係数ブロックを量子化ユニット1206から受け取ることができ、シンタックス要素を予測処理ユニット1200から受け取ることができる。エントロピー符号化ユニット1218は、1つまたは複数のエントロピー符号化動作をデータに対して実行して、エントロピー符号化されたデータを生成し得る。たとえば、エントロピー符号化ユニット1218は、CABAC動作、コンテキスト適応型可変長コーディング(CAVLC)動作、可変長対可変長(V2V)コーディング動作、シンタックスベースコンテキスト適応型バイナリ算術コーディング(SBAC)動作、確率区間区分エントロピー(PIPE)コーディング動作、指数ゴロム符号化動作、または別のタイプのエントロピー符号化動作を、データに対して実行し得る。ビデオエンコーダ20は、エントロピー符号化ユニット1218によって生成された、エントロピー符号化されたデータを含むビットストリームを出力し得る。たとえば、ビットストリームは、CUのための変換係数の値を表すデータを含み得る。

20

【0185】

図13は、本開示の技法を実施するように構成される例示的なビデオデコーダ30を示すブロック図である。図13は説明のために提供され、本開示において広く例示され説明されるような技法の限定でない。説明のために、本開示は、HEVCコーディングの文脈においてビデオデコーダ30を説明する。しかしながら、本開示の技法は、他のコーディング規格または方法に適用であり得る。

30

【0186】

図13の例において、ビデオデコーダ30は、エントロピー復号ユニット1300、ビデオデータメモリ1301、予測処理ユニット1302、逆量子化ユニット1304、逆変換処理ユニット1306、再構築ユニット1308、フィルタユニット1310、および復号ピクチャバッファ1312を含む。予測処理ユニット1302は、動き補償ユニット1314およびイントラ予測処理ユニット1316を含む。他の例では、ビデオデコーダ30は、より多数の、より少数の、または異なる機能構成要素を含み得る。

【0187】

ビデオデータメモリ1301は、ビデオデコーダ30の構成要素によって復号されるべき、符号化されたビデオビットストリームなどの符号化されたビデオデータを記憶し得る。ビデオデータメモリ1301に記憶されるビデオデータは、たとえば、コンピュータ可読媒体16から、たとえば、カメラなどのローカルビデオソースから、ビデオデータの有線ネットワーク通信もしくはワイヤレスネットワーク通信を介して、または物理データ記憶媒体にアクセスすることによって、取得され得る。ビデオデータメモリ1301は、符号化されたビデオビットストリームからの符号化されたビデオデータを記憶するコーディングピクチャバッファ(CPB)を形成し得る。復号ピクチャバッファ1312は、たとえば、イントラコーディングモードまたはインターフォーマンスモードにおいて、ビデオデコーダ30によってビデオデータを復号する際に使用するための、または出力のための、参照ビデオデータを記憶する参照ピクチャメモリであってもよい。ビデオデータメモリ1301および復号ピクチャバッファ1312は、シンクロナスDRAM(SDRAM)を含むダイナミックランダムアクセスメモリ(DRAM)

40

50

)、磁気抵抗RAM(MRAM)、抵抗変化RAM(RRAM(登録商標))、または他のタイプのメモリデバイスなどの、様々なメモリデバイスのいずれかによって形成され得る。ビデオデータメモリ1301および復号ピクチャバッファ1312は、同一のメモリデバイスまたは別々のメモリデバイスによって提供され得る。様々な例では、ビデオデータメモリ1301は、ビデオデコーダ30の他の構成要素とともにオンチップであることがあり、または、それらの構成要素に対してオフチップであることがある。ビデオデータメモリ1301は、図1の記憶媒体28と同じであることがある、またはその一部であることがある。

【0188】

ビデオデータメモリ1301は、ビットストリームの符号化されたビデオデータ(たとえば、NALユニット)を受信し記憶する。エントロピー復号ユニット1300は、符号化されたビデオデータ(たとえば、NALユニット)をビデオデータメモリ1301から受信することができ、NALユニットを構文解析して、シンタックス要素を取得することができる。エントロピー復号ユニット1300は、NALユニットの中のエントロピー符号化されたシンタックス要素をエントロピー復号し得る。予測処理ユニット1302、逆量子化ユニット1304、逆変換処理ユニット1306、再構築ユニット1308、およびフィルタユニット1310は、ビットストリームから抽出されたシンタックス要素に基づいて、復号されたビデオデータを生成し得る。エントロピー復号ユニット1300は、エントロピー符号化ユニット1218のプロセスとは全般に逆のプロセスを実行し得る。

【0189】

ビットストリームからシンタックス要素を取得することに加えて、ビデオデコーダ30は、CUに対して再構築動作を実行し得る。CUに対して再構築動作を実行するために、ビデオデコーダ30は、CUの各TUに対して再構築動作を実行し得る。CUのTUごとに再構築動作を実行することによって、ビデオデコーダ30は、CUの残差ブロックを再構築し得る。

【0190】

CUのTUに対して再構築動作を実行することの一部として、逆量子化ユニット1304は、TUと関連付けられる係数ブロックを逆量子化(inverse quantize)、すなわち、逆量子化(dequantize)し得る。逆量子化ユニット1304が係数ブロックを逆量子化した後、逆変換処理ユニット1306は、TUと関連付けられた残差ブロックを生成するために、1つまたは複数の逆変換を係数ブロックに適用し得る。たとえば、逆変換処理ユニット1306は、逆DCT、逆整数変換、逆カルーネンレーベ変換(KLT)、逆回転変換、逆方向変換、または別の逆変換を係数ブロックに適用し得る。

【0191】

逆量子化ユニット1304は、本開示の特定の技法を実行し得る。たとえば、ビデオデータのピクチャのCTUのCTB内の複数の量子化グループの少なくとも1つのそれぞれの量子化グループに対して、逆量子化ユニット1304は、ビットストリームにおいてシグナリングされるローカル量子化情報に少なくとも一部に基づいて、それぞれの量子化グループのためのそれぞれの量子化パラメータを導出し得る。加えて、この例では、逆量子化ユニット1304は、それぞれの量子化グループのためのそれぞれの量子化パラメータに基づいて、CTUのCUのTUの変換ブロックの少なくとも1つの変換係数を逆量子化し得る。この例では、逆量子化グループはコーディング順序において連続するCUまたはコーディングブロックのグループとして定義されるので、それぞれの量子化グループの境界がCUまたはコーディングブロックの境界であるはずであり、それぞれの量子化グループのサイズは閾値以上である。ビデオデコーダ30(たとえば、逆変換処理ユニット1306、再構築ユニット1308、およびフィルタユニット1310)は、変換ブロックの逆量子化された変換係数に基づいて、CUのコーディングブロックを再構築し得る。

【0192】

PUがイントラ予測を使用して符号化される場合、イントラ予測処理ユニット1316は、イントラ予測を実行してPUの予測ブロックを生成し得る。イントラ予測処理ユニット1316は、イントラ予測モードを使用して、空間的に隣接するブロックのサンプルに基づいて、PUの予測ブロックを生成し得る。イントラ予測処理ユニット1316は、ビットストリームから

10

20

30

40

50

取得された1つまたは複数のシンタックス要素に基づいて、PUのイントラ予測モードを決定し得る。

【0193】

PUがインター予測を使用して符号化される場合、動き補償ユニット1314は、PUの動き情報を決定し得る。動き補償ユニット1314は、PUの動き情報に基づいて、1つまたは複数の参照ブロックを決定し得る。動き補償ユニット1314は、1つまたは複数の参照ブロックに基づいて、PUの予測ブロック(たとえば、予測輝度ブロック、予測Cbブロック、および予測Crブロック)を生成し得る。

【0194】

動き補償ユニット1314は、本開示の他の箇所で説明されたように、アフィン動きモデルのための技法を適用し得る。たとえば、動き補償ユニット1314はソースアフィンブロックを選択することができ、ソースアフィンブロックは現在のブロックに空間的に隣接するアフィンコーディングされたブロックである。この例では、動き補償ユニット1314は、ソースアフィンブロックの制御点の動きベクトルを外挿して、現在のブロックの制御点の動きベクトル予測子を決定し得る。この例では、動き補償ユニット1314は、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを、アフィンMVPセット候補リストに挿入する。さらに、動き補償ユニット1314は、ビットストリームにおいてシグナリングされるインデックスに基づいて、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定する。この例では、エントロピー復号ユニット1300は、ビットストリームから、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示すMVDを取得し得る。動き補償ユニット1314は、この例では、選択されたアフィンMVPセットおよびMVDに含まれる動きベクトル予測子に基づいて、現在のブロックの制御点の動きベクトルを決定し得る。この例では、動き補償ユニット1314は、現在のブロックの制御点の動きベクトルに基づいて、予測ブロックを生成し得る。

10

20

30

【0195】

再構築ユニット1308は、CUのTUのための変換ブロック(たとえば、輝度変換ブロック、Cb変換ブロック、およびCr変換ブロック)、ならびにCUのPUの予測ブロック(たとえば、輝度ブロック、Cbブロック、およびCrブロック)、すなわちイントラ予測データまたはインター予測データのいずれかを適宜使用して、CUのコーディングブロック(たとえば、輝度コーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)を再構築し得る。たとえば、再構築ユニット1308は、予測ブロック(たとえば、輝度予測ブロック、Cb予測ブロック、およびCr予測ブロック)の対応するサンプルに、変換ブロック(たとえば、輝度変換ブロック、Cb変換ブロック、およびCr変換ブロック)のサンプルを加算して、CUのコーディングブロック(たとえば、輝度コーディングブロック、Cbコーディングブロック、およびCrコーディングブロック)を再構築し得る。

【0196】

フィルタユニット1310は、デブロッキング動作を実行して、CUのコーディングブロックと関連付けられるブロッキングアーティファクトを低減し得る。ビデオデコーダ30は、CUのコーディングブロックを復号ピクチャバッファ1312に記憶し得る。復号ピクチャバッファ1312は、後続の動き補償、イントラ予測、および図1のディスプレイデバイス32などのディスプレイデバイス上での提示のために、参照ピクチャを提供し得る。たとえば、ビデオデコーダ30は、復号ピクチャバッファ1312の中のブロックに基づいて、他のCUのPUのためにイントラ予測動作またはインター予測動作を実行し得る。

40

【0197】

図14Aは、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフロー チャートである。本開示のフロー チャートは、例として与えられる。本開示の技法による他の例は、より多数の、少数の、もしくは異なる活動を含むことがあり、または、活動は異なる順序で実行されることがある。

【0198】

50

上で説明されたように、本開示の1つまたは複数の技法によれば、1つのインター予測方向のためのアフィン動きモデルが、別のインター予測方向のためのアフィン動きモデルのシグナリングを改善するために使用され得る。図14Aおよび図14Bは、そのような技法に従った例示的な動作を示す。

【0199】

図14Aの例では、ビデオエンコーダ20は、ビデオデータの現在のブロックのアフィン動きモデルの制御点の第1の動きベクトルに基づいて、現在のブロックのアフィン動きモデルの制御点の第2の動きベクトルを決定する(1400)。第1の動きベクトルはリストX(Xは0または1である)に対応し、第2の動きベクトルはリストY(Yは1-Xである)に対応する。制御点の第2の動きベクトルを決定するための一例では、ビデオエンコーダ20は、制御点のリストX動きベクトルを決定する。ビデオエンコーダ20は、本開示において提供される例のいずれかに従って、制御点のリストX動きベクトルを決定し得る。さらに、ビデオエンコーダ20は、制御点のリストX動きベクトルのもとで最良のレート-ひずみコストをもたらす、制御点のリストY動きベクトルを決定するための探索を実行し得る。この例では、ビデオエンコーダ20は、制御点のリストX動きベクトルと制御点のリストY動きベクトルとの間の差分を示す、動きベクトル差分をシグナリングし得る。

10

【0200】

加えて、ビデオエンコーダ20は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(1402)。ビデオエンコーダ20は、本開示の他の箇所で提供される例に従って、予測ブロックを生成し得る。たとえば、ビデオエンコーダ20は、現在のブロックのアフィン動きモデルの制御点のリストX動きベクトルおよびリストY動きベクトルを使用して、現在のブロックのサブブロックのためのリストX動きベクトルおよびリストY動きベクトルを決定することができ、次いで、動き補償補間フィルタを適用してサブブロックの各々の予測ブロックを生成し、それにより現在のブロックの予測ブロックを生成することができる。

20

【0201】

図14Aの例では、ビデオエンコーダ20はまた、予測ブロックに基づいて現在のブロックを復号するために使用されるデータを生成する(1404)。ビデオエンコーダ20は、本開示の他の箇所で提供される例のいずれかに従って、現在のブロックを復号するために使用されるデータを生成し得る。たとえば、本開示の他の箇所で説明されるように、ビデオエンコーダ20は、残差データを生成し、変換を残差データに適用し、変換された残差データを量子化し、量子化された変換された残差データを表すシンタックス要素にエントロピー符号化を適用し得る。

30

【0202】

図14Bは、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。図14Bの例では、ビデオデコーダ30は、ビデオデータの現在のブロックのアフィン動きモデルの制御点の第1の動きベクトルに基づいて、現在のブロックのアフィン動きモデルの制御点の第2の動きベクトルを決定する(1420)。第1の動きベクトルは、第1の参照ピクチャリスト(すなわち、リストX、ここでXは0または1である)に対応する。第2の動きベクトルは、第2の異なる参照ピクチャリスト(すなわち、リストY、ここでYは1-Xである)に対応する。この例では、ビデオデコーダ30は、本開示において提供される他の例に従って、制御点のリストX動きベクトルを決定し得る。さらに、この例では、制御点のリストY動きベクトルを決定するために、ビデオエンコーダ30は、制御点のリストX動きベクトルと制御点のリストY動きベクトルとの間の差分を示す動きベクトル差分を、ビットストリームから取得し得る。この例では、ビデオデコーダ30は、動きベクトル差分を制御点のリストX動きベクトルに加算して、制御点のリストY動きベクトルを決定し得る。

40

【0203】

加えて、ビデオデコーダ30は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(1422)。ビデオデコーダ30は、図14Aのビデオエンコーダ20と同じ方

50

式で予測ブロックを生成し得る。ビデオデコーダ30は、残差データおよび予測ブロックに基づいて現在のブロックを再構築し得る(1424)。たとえば、ビデオデコーダ30は、残差データのサンプルを予測ブロックの対応するサンプルに加算することによって少なくとも一部、現在のブロックを再構築し得る。

【0204】

図15Aは、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフロー チャートである。上で論じられたように、本開示のいくつかの技法によれば、現在のブロックの制御点のMVは、現在のブロックの他の制御点のMVのためのMVP候補として使用され得る。図15Aおよび図15Bは、これらの技法に従った例示的な動作を示す。

【0205】

図15Aの例では、ビデオエンコーダ20は、ビデオデータの現在のブロックのアフィン動きモデルの第1の制御点の動きベクトルに基づいて、現在のブロックのアフィン動きモデルの第2の制御点の動きベクトルを決定する(1500)。たとえば、ビデオエンコーダ20は、現在のブロックのアフィン動きモデルの第1の制御点(たとえば、左上の制御点)の動きベクトルを、現在のブロックのアフィン動きモデルの第2の制御点(たとえば、右上の制御点)の動きベクトルの予測のために使用される候補リストに候補として含め得る。候補リストの中の他の候補は、隣接するアフィンコーディングされたブロックの対応する制御点(たとえば、左上の制御点)の動きベクトルを含み得る。この例では、ビデオエンコーダ20は次いで、(たとえば、レート-ひずみコストに基づいて)候補リストからある候補を選択し得る。さらに、この例では、ビデオエンコーダ20は次いで、第2の制御点の動きベクトル予測子として、選択された候補の動きベクトルを使用し得る。いくつかの例では、ビデオエンコーダ20は、選択された候補の動きベクトルと第2の制御点の動きベクトル予測子との間の差分を示すMVDをシグナリングする。

【0206】

さらに、ビデオエンコーダ20は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(1502)。加えて、ビデオエンコーダ20は、予測ブロックに基づいて現在のブロックを復号するために使用されるデータを生成する(1504)。ビデオエンコーダ20は、本開示の他の箇所で提供される例に従って、予測ブロックを生成し、現在のブロックを復号するために使用されるデータを生成し得る。

【0207】

図15Bは、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフロー チャートである。図15Bの例では、ビデオデコーダ30は、ビデオデータの現在のブロックのアフィン動きモデルの第1の制御点の動きベクトルに基づいて、現在のブロックのアフィン動きモデルの第2の制御点の動きベクトルを決定する(1520)。ビデオデコーダ30は、図15Aにおいてビデオエンコーダ20に関して上で説明されたのと同じ方式で、現在のブロックのアフィン動きモデルの第2の制御点の動きベクトルを決定し得る。第2の制御点の動きベクトルを決定するために、ビデオデコーダ30は、候補リストの中の選択された候補を示すインデックスをピットストリームから取得し得る。候補リストは、現在のブロックの第1の制御点の動きベクトルを含む、制御点の動きベクトルを含み得る。いくつかの例では、ビデオデコーダ30は、シグナリングされたMVDを選択された候補の動きベクトルに加算することによって、第2の制御点の動きベクトルを決定し得る。

【0208】

加えて、ビデオデコーダ30は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(1522)。ビデオデコーダ30は、予測ブロックに基づいてブロックを再構築する(1524)。ビデオデコーダ30は、本開示の他の箇所で提供される例に従って、予測ブロックを生成し、ブロックを再構築し得る。たとえば、ビデオデコーダ30は、予測ブロックおよび復号された残差データに基づいてブロックを再構築し得る。

【0209】

図16Aは、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフロー チャートである。上で論じられたように、本開示のいくつかの技法によれば、ビデ

10

20

30

40

50

オコーダは、現在のブロックの1つまたは複数の隣接アフィンブロックの動きモデルを使用して、現在のアフィン動きモデルを予測し得る。図16Aの例では、ビデオエンコーダ20は、単一の隣接アフィンブロックの動きモデルを使用して、現在のブロックのためのアフィン動きモデルの少なくとも2つの制御点を予測する(1600)。加えて、ビデオエンコーダ20は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(1602)。ビデオエンコーダ20は次いで、予測ブロックに基づいて現在のブロックを復号するために使用されるデータを生成する(1604)。ビデオエンコーダ20は、本開示の他の箇所で提供される例に従って、データを生成し得る。

【0210】

図16Bは、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。図16Bの例では、ビデオデコーダ30は、単一の隣接アフィンブロックの動きモデルを使用して、現在のブロックのためのアフィン動きモデルを予測する(1620)。さらに、ビデオデコーダ30は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(1622)。ビデオデコーダ30は次いで、予測ブロックに基づいて現在のブロックを再構築し得る(1624)。ビデオデコーダ30は、本開示の他の箇所で提供される例のいずれかに従って、予測ブロックを生成し、現在のブロックを再構築し得る。

10

【0211】

図17は、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。図17および図18は、図16Aおよび図16Bにおいて概説された動作のためのより詳細なフローチャートである。図17の例では、ビデオエンコーダ20がソースアフィンブロックを選択する(1700)。ソースアフィンブロックは、現在のブロックに空間的に隣接するアフィンコーディングされたブロックである。ビデオエンコーダ20は、様々な方法でソースアフィンブロックを選択し得る。たとえば、ビデオエンコーダ20は、ソースアフィンブロックが、事前に定義された訪問順序で訪問される複数の隣接ブロックの最初に生じるアフィンコーディングされたブロックであると決定し得る。いくつかの例では、ビデオエンコーダ20は、ソースアフィンブロックが、事前に定義された訪問順序に基づく複数の事前に定義された優先度セットに従って複数の隣接ブロックの最初に発生する利用可能なアフィンコーディングされたブロックであると決定し得る。アフィンコーディングされたブロックは、アフィンコーディングされたブロックが事前に定義された優先度セットのうちの1つの中にはない場合、利用可能であると見なされない。優先度セットの様々な例が、本開示の他の箇所で説明される。

20

【0212】

加えて、ビデオエンコーダ20は、ソースアフィンブロックの制御点の動きベクトルを外挿して、現在のブロックの制御点の動きベクトル予測子を決定し得る(1702)。たとえば、ビデオエンコーダ20は、アフィンソースブロックの制御点の動きベクトルによって決定されるアフィンパラメータによって定義されるアフィン動きモデルを構築し得る。ビデオエンコーダ20は次いで、構築されたアフィン動きモデルを使用して、現在のブロック(またはいわゆる外挿された)の制御点の動きベクトルを導出し得る。たとえば、現在のブロックの制御点の動きベクトルを外挿するために、ビデオエンコーダ20は、構築されたアフィン動きモデルの動きベクトルと、式2の現在のブロックの制御点の(x,y)の場所とを使用して、制御点の動きベクトルを決定し得る。

30

【0213】

さらに、ビデオエンコーダ20は、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを、アフィンMVPセット候補リストに挿入し得る(1704)。いくつかの例では、ビデオエンコーダ20はまた、アフィンMVPセットに従来のアフィンMVPセットを含め得る。たとえば、一例では、ビデオエンコーダ20は、現在のブロックの第1の制御点の隣のブロックの動きベクトルとして、第1の動きベクトル予測子を決定し得る。この例では、ビデオエンコーダ20は、現在のブロック(たとえば、図6AのブロックA、B、もしくはC、またはブロックDもしくはE)の第2の制御点の隣のブロックの動きベクトルとして第2の動きベクトル予測子を決定する。この例では、ビデオエンコーダ20は、アフィンMVPセット

40

50

候補リストに、第1の動きベクトル予測子および第2の動きベクトル予測子を含むアフィンMVPセットを挿入する。

【0214】

いくつかの例では、ビデオエンコーダ20は第2のソースアフィンブロックを選択する。第2のソースアフィンブロックは、現在のブロックに空間的に隣接する異なるアフィンコーディングされたブロックである。この例では、ビデオエンコーダ20は、第2のソースアフィンブロックの制御点の動きベクトルを外挿して、現在のブロックの制御点の第2の動きベクトル予測子を決定する。さらに、ビデオエンコーダ20は、第2のアフィンMVPセットをアフィンMVPセット候補リストに挿入する。第2のアフィンMVPセットは、現在のブロックの制御点の第2の動きベクトル予測子を含む。

10

【0215】

続いて、ビデオエンコーダ20は、アフィンMVPセット候補リストの中のアフィンMVPセットを選択する(1706)。ビデオエンコーダ20は、アフィンMVPセット候補リストの中のアフィンMVPセットのレート-ひずみ分析に基づいて、アフィンMVPセットを選択し得る。

【0216】

ビデオエンコーダ20は、ビットストリームにおいて、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示すMVDをシグナリングし得る(1708)。加えて、ビデオエンコーダ20は、ビットストリームにおいて、選択されたアフィンMVPセットのアフィンMVPセット候補リストにおける場所を示すインデックスをシグナリングし得る(1710)。

20

【0217】

図18は、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。図18の例では、ビデオデコーダ30がソースアフィンブロックを選択する(1800)。ソースアフィンブロックは、現在のブロックに空間的に隣接するアフィンコーディングされたブロックである。ビデオデコーダ30は、本開示の他の箇所で説明されるように、ビデオエンコーダ20と同じ方法でソースアフィンブロックを選択し得る。

【0218】

加えて、ビデオデコーダ30は、ソースアフィンブロックの制御点の動きベクトルを外挿して、現在のブロックの制御点の動きベクトル予測子を決定する(1802)。ビデオエンコーダ30は、現在のブロックの制御点の動きベクトル予測子を含むアフィンMVPセットを、アフィンMVPセット候補リストに挿入する(1804)。本開示の他の箇所で説明されるように、ビデオデコーダ30は、制御点の動きベクトルを外挿し、ビデオエンコーダ20と同じ方法でアフィンMVPセットを挿入し得る。ビデオデコーダ30はまた、ビデオエンコーダ20に関して上で説明されたように、追加のアフィンMVPセットをアフィンMVPセット候補リストに追加し得る。

30

【0219】

さらに、ビデオデコーダ30は、ビットストリームにおいてシグナリングされるインデックスに基づいて、アフィンMVPセット候補リストの中の選択されたアフィンMVPセットを決定する(1806)。ビデオデコーダ30は、ビットストリームから、現在のブロックの制御点の動きベクトルと選択されたアフィンMVPセットの中の動きベクトル予測子との間の差分を示すMVDを取得する(1808)。加えて、ビデオデコーダ30は、選択されたアフィンMVPセットおよびMVDに含まれる動きベクトル予測子に基づいて、現在のブロックの制御点の動きベクトルを決定する(1810)。たとえば、ビデオデコーダ30は、MVDを対応する動きベクトル予測子に加算して、現在のブロックの制御点の動きベクトルを決定し得る。

40

【0220】

ビデオデコーダ30は次いで、現在のブロックの制御点の動きベクトルに基づいて、予測ブロックを生成し得る(1812)。ビデオデコーダ30は、残差データおよび予測ブロックに基づいて現在のブロックを再構築し得る(1814)。ビデオデコーダ30は、本開示の他の箇所で提供される例に従って、予測ブロックを生成し、現在のブロックを再構築し得る。

【0221】

50

図19Aは、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフロー チャートである。上で論じられたように、本開示のいくつかの技法によれば、アフィンインター モードまたはアフィンマージモードにおいて、現在のブロックの各サブブロック(たとえば、 4×4 のブロック)のアフィンパラメータは、サブブロック自身の隣接ブロックから予測され、または直接継承され得る。図19Aおよび図19Bは、そのような技法に従った例示的な動作を示す。

【0222】

図19Aの例では、ビデオデータの現在のブロックは複数のサブブロックに区分される。複数のサブブロックの各々のそれぞれのサブブロックに対して、ビデオエンコーダ20は、それぞれの隣接アフィンブロックの外挿された動きを使用して、それぞれのサブブロックのアフィン動きを予測する(1900)。ビデオエンコーダ20は、本開示の他の箇所で説明されるのと同じ方式で、それぞれの隣接アフィンブロックの動きを外挿し得る。いくつかの例では、それぞれの隣接アフィンブロックの外挿された動きを使用してそれぞれのサブブロックのアフィン動きを予測するために、ビデオエンコーダ20は、隣接アフィンブロックの外挿された動きからそれぞれのサブブロックのアフィン動きを予測する。いくつかの事例では、隣接アフィンブロックは現在のブロックの別のサブブロックである。いくつかの例では、それぞれの隣接アフィンブロックの外挿された動きを使用するために、ビデオエンコーダ20は、隣接アフィンブロックの外挿された動きから直接、各々のそれぞれのサブブロックのアフィン動きを継承する。言い換えると、ビデオエンコーダ20は、動きインデックスに等しいそれぞれのサブブロックの制御点の参照インデックスおよび動きベクトルと、それぞれの隣接アフィンブロックの制御点の外挿された動きベクトルとを設定する。たとえば、図10に示されるように、サブブロック1002は、アフィンブロック1004の制御点の参照インデックスおよび外挿された動きベクトルを継承し得る。

【0223】

加えて、ビデオエンコーダ20は、サブブロックのアフィン動きに基づいて、予測ブロックを生成する(1902)。たとえば、複数のサブブロックの各々のそれぞれのサブブロックに対して、ビデオエンコーダ20は、それぞれのサブブロックのアフィン動きを使用して、それぞれのサブブロックのそれぞれの予測サブブロックを生成し得る。この例では、ビデオエンコーダ20は、アフィン動きを使用して予測ブロックを生成することについて本開示の他の箇所で説明されたのと同じ方式で、それぞれのサブブロックのアフィン動きを使用してそれぞれのサブブロックのそれぞれの予測サブブロックを生成し得る。たとえば、ビデオエンコーダ20は、式(2)を使用してそれぞれのサブブロックの動きベクトルのx成分およびy成分を計算し得る。ビデオエンコーダ20は次いで、それぞれのサブブロックの動きベクトルを使用して、それぞれのサブブロックの予備予測ブロックまたは最終予測ブロックを決定し得る。さらに、この例では、ビデオエンコーダ20は、予測サブブロックを合成して現在のブロックの予測ブロックを生成し得る。

【0224】

ビデオエンコーダ20は、予測ブロックに基づいて現在のブロックを復号するために使用されるデータを生成する(1904)。ビデオエンコーダ20は、本開示の他の箇所で提供される対応する例のいずれかに従って、現在のブロックを復号するために使用されるデータを生成し得る。

【0225】

図19Bは、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフロー チャートである。図16Bの例では、ビデオデータの現在のブロックは複数のサブブロックに区分される。複数のサブブロックの各々のそれぞれのサブブロックに対して、ビデオデコーダ30は、それぞれの隣接アフィンブロックの外挿された動きを使用して、それぞれのサブブロックのアフィン動きを予測する(1920)。ビデオデコーダ30は、ビデオエンコーダ20に関して上で説明されたのと同じ方式で、それぞれのサブブロックのアフィン動きを予測し得る。いくつかの例では、それぞれの隣接アフィンブロックの外挿された動きを使用するために、ビデオデコーダ30は、隣接アフィンブロックの外挿された動きから、各

々のそれぞれのサブブロックのアフィン動きを予測する。いくつかの例では、それぞれの隣接アフィンブロックの外挿された動きを使用するために、ビデオデコーダ30は、隣接アフィンブロックの外挿された動きから直接、各々のそれぞれのサブブロックのアフィン動きを継承する。

【0226】

さらに、ビデオデコーダ30は、サブブロックのアフィン動きに基づいて、予測ブロックを生成する(1922)。ビデオデコーダ30は、ビデオエンコーダ20に関して上で説明されたのと同じ方式で、予測ブロックを生成し得る。ビデオデコーダ30は、予測ブロックに基づいて現在のブロックを再構築する(1924)。たとえば、ビデオデコーダ30は、ビットストリームから復号された対応する残差サンプルに予測ブロックのサンプルを加算することによって、現在のブロックを再構築し得る。10

【0227】

図20Aは、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフロー チャートである。上で論じられたように、本開示のいくつかの技法によれば、1つのMVP候補セット内のすべてのMVPが同一であるとは限らないことを確実にするために、オフセットがMVPに加算され得る。図20Aおよび図20Bは、これらの技法に従った例示的な動作を示す。

【0228】

具体的には、図20Aの例では、第1の動きベクトルは、現在のブロックのアフィン動きモデルの第1の制御点の動きベクトルである。第2の動きベクトルは、現在のブロックのアフィン動きモデルの第2の制御点の動きベクトルである。動きベクトルの候補リストの中の候補の総数が2未満であることと、第1の動きベクトルおよび第2の動きベクトルが同じであることとに基づいて、ビデオエンコーダ20はオフセットを動きベクトル予測子に加算する(2000)。本開示の他の箇所で説明されるように、オフセットは、候補リストの中の異なる場所に対しては異なり得る。20

【0229】

加えて、ビデオエンコーダ20は、動きベクトル予測子を候補リストに含める(2002)。たとえば、ビデオエンコーダ20は、動きベクトル予測子を動きベクトル予測子のアレイに含め得る。加えて、ビデオエンコーダ20は、候補リストの中の候補を選択する(2004)。ビデオエンコーダ20は、選択された候補が候補リストの中の候補のうちで最良のレート-ひずみ値をもたらすように、候補を選択し得る。さらに、ビデオエンコーダ20は、選択された候補を使用して予測ブロックを決定する(2006)。たとえば、ビデオエンコーダ20は、選択された候補によって指定される動きベクトルを使用して、参照ピクチャの中の位置を特定し得る。この例では、ビデオエンコーダ20は、参照ピクチャの中の特定された位置にあるサンプルのブロックのコピーに回転を適用することによって、予測ブロックを決定し得る。いくつかの例では、ビデオエンコーダ20は、選択された候補を使用して第1の予備予測ブロックを決定し、また、第2の予測ブロックを(たとえば、異なる参照ピクチャリストの中の参照ピクチャの中のサンプルに基づいて)決定し得る。この例では、ビデオエンコーダ20は、第1の予備予測ブロックおよび第2の予備予測ブロックの中の対応するサンプルの加重平均として、予測ブロックの中のサンプルを決定し得る。3040

【0230】

ビデオエンコーダ20は次いで、現在のブロックおよび予測ブロックのサンプルに基づいて残差データを生成し得る(2008)。たとえば、ビデオエンコーダ20は、残差データの各サンプルが、現在のブロックの中の対応するサンプルと予測ブロックの中の対応するサンプルとの間の差分を示すように、残差データを生成し得る。加えて、ビデオエンコーダ20は、ビデオデータの符号化された表現を備えるビットストリームに、候補リストの中の選択された候補の指示を含める(2010)。

【0231】

図20Bは、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフロー チャートである。図18Bの例では、動きベクトルの候補リストの中の候補の総数が2未50

満であることと、第1の動きベクトルおよび第2の動きベクトルが同じであることとに基づいて、ビデオデコーダ30はオフセットを動きベクトル予測子に加算する(2020)。この例では、第1の動きベクトルは、ビデオデータの現在のブロックのアフィン動きモデルの第1の制御点の動きベクトルである。第2の動きベクトルは、現在のブロックのアフィン動きモデルの第2の制御点の動きベクトルである。

【0232】

さらに、図20Bの例では、ビデオエンコーダ30は、動きベクトル予測子を候補リストに含める(2022)。ビデオデコーダ30は次いで、候補リストの中の選択された候補を決定し得る(2024)。加えて、ビデオデコーダ30は、選択された候補を使用して予測ブロックを決定し得る(2026)。ビデオデコーダ30は次いで、予測ブロックに基づいて現在のブロックを再構築し得る(2028)。いくつかの例では、ビデオデコーダ30は、予測ブロックおよび残差データに基づいて現在のブロックを再構築し得る。

10

【0233】

図21Aは、本開示の技法による、ビデオデータを符号化するための例示的な動作を示すフローチャートである。上で述べられたように、本開示の技法によれば、`mvd_l1_zero_flag`はいくつかのコーディングモードに適用可能であることがあり、他のモードでは、このフラグは、MVDが0に等しいことを`mvd_l1_zero_flag`が示す場合であっても無視される。図21Aおよび図21Bは、この技法に従った例示的な動作を示す。

【0234】

具体的には、図21Aの例では、ビデオエンコーダ20は、ビットストリームに、第2の参照ピクチャリスト(たとえば、リスト1)に対する動きベクトル差分がビットストリームにおいてシグナリングされるかどうかを示すフラグ(たとえば、`mvd_l1_zero_flag`)を含める(2100)。ビデオデータの現在のブロックの動きがアフィン動きモードであることに基づいて、フラグの値とは無関係に、ビデオエンコーダ20はビットストリームにMVDを含める(2102)。たとえば、ビデオエンコーダ20は、MVDの垂直成分を示すシンタックス要素およびMVDの水平成分を示す第2の要素をビットストリームに含め得る。

20

【0235】

さらに、ビデオエンコーダ20は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生成する(2104)。ビデオデコーダ20は、本開示の他の箇所で提供される例のいずれかに従って、現在のブロックのアフィン動きモデルに基づいて予測ブロックを生成し得る。ビデオエンコーダ20は次いで、予測ブロックに基づいて現在のブロックを復号するために使用されるデータを生成し得る(2106)。たとえば、本開示の他の箇所で説明されるように、ビデオエンコーダ20は、残差データを生成し、変換を残差データに適用し、変換された残差データを量子化し、量子化された変換された残差データを表すシンタックス要素にエントロピー符号化を適用し得る。

30

【0236】

図21Bは、本開示の技法による、ビデオデータを復号するための例示的な動作を示すフローチャートである。図21Bの例では、ビデオデコーダ30は、ビットストリームから、第2の参照ピクチャリスト(たとえば、リスト1)に対する動きベクトル差分がビットストリームにおいてシグナリングされるかどうかを示すフラグ(たとえば、`mvd_l1_zero_flag`)を取得する(2120)。

40

【0237】

加えて、図21Bの例では、ビデオデータの現在のブロックの動きがアフィン動きモードであることに基づいて、フラグの値とは無関係に、ビデオデコーダ30はビットストリームからMVDを取得する(2122)。言い換えると、ビデオデコーダ30はMVDをビットストリームから復号する。いくつかの例では、ビットストリームからMVDを取得するために、ビデオデコーダ30は、MVDの垂直成分を示す第1のシンタックス要素およびMVDの水平成分を示す第2のシンタックス要素をビットストリームから取得する。ビデオデコーダ30は、動きベクトル差分に基づいて、現在のブロックのアフィン動きモデルを決定する(2124)。加えて、ビデオデコーダ30は、現在のブロックのアフィン動きモデルに基づいて、予測ブロックを生

50

成する(2126)。さらに、ビデオデコーダ30は、予測ブロックに基づいてブロックを再構築する(2128)。ビデオデコーダ30は、本開示の他の箇所で提供される例のいずれかに従って、予測ブロックを生成し、ブロックを再構築し得る。

【0238】

本開示のいくつかの態様は、説明を目的にHEVC規格の拡張に関して説明されている。しかしながら、本開示において説明される技法は、現在開発中のまたはまだ開発されていない他の標準的なまたは独自のビデオコーディングプロセスを含む、他のビデオコーディングプロセスにとって有用であり得る。

【0239】

本開示で説明されるようなビデオコーダは、ビデオエンコーダまたはビデオデコーダを指すことがある。同様に、ビデオコーディングユニットは、ビデオエンコーダまたはビデオデコーダを指すことがある。同様に、ビデオコーディングは、場合により、ビデオ符号化またはビデオ復号を指すことがある。本開示では、「に基づいて」という句は、「だけに基づいて」、「に少なくとも一部に基づいて」、または「何らかの形でに基づいて」を示し得る。本開示は、サンプルの1つまたは複数のブロックのサンプルをコーディングするために使用される1つまたは複数のサンプルブロックおよびシンタックス構造を指すために、「ビデオ単位」または「ビデオブロック」または「ブロック」という用語を使用することがある。例示的なタイプのビデオ単位は、CTU、CU、PU、変換単位(TU)、マクロブロック、マクロブロック区分などを含み得る。いくつかの文脈では、PUの議論はマクロブロックまたはマクロブロック区分の議論と相互に交換され得る。例示的なタイプのビデオブロックは、コーディングツリーブロック、コーディングブロック、およびビデオデータの他のタイプのブロックを含み得る。

10

20

【0240】

例によっては、本明細書において説明された技法のうちのいずれかのいくつかの行為またはイベントが、異なるシーケンスで実行されてよく、追加され、統合され、または完全に除外されてよい(たとえば、説明されたすべての行為またはイベントが技法の実践にとって必要であるとは限らない)ことを認識されたい。その上、いくつかの例では、行為またはイベントは、連続的にではなく、たとえば、マルチスレッド処理、割込み処理、または複数のプロセッサを通じて並行して実行されてよい。

【0241】

30

1つまたは複数の例では、説明された機能は、ハードウェア、ソフトウェア、ファームウェア、またはそれらの任意の組合せとして実装され得る。ソフトウェアで実装される場合、機能は、1つまたは複数の命令またはコードとして、コンピュータ可読媒体上に記憶されるかまたはコンピュータ可読媒体を介して送信されてよく、ハードウェアベースの処理ユニットによって実行されてよい。コンピュータ可読媒体は、データ記憶媒体などの有形媒体に相当するコンピュータ可読記憶媒体、または、たとえば、通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を容易にする任意の媒体を含む通信媒体を含み得る。このようにして、コンピュータ可読媒体は、一般に、(1)非一時的である有形コンピュータ可読記憶媒体、または(2)信号もしくは搬送波などの通信媒体に相当し得る。データ記憶媒体は、本開示で説明された技法の実装のための命令、コード、および/またはデータ構造を取り出すために、1つまたは複数のコンピュータまたは1つまたは複数の処理回路によってアクセスされ得る任意の利用可能な媒体であり得る。コンピュータプログラム製品はコンピュータ可読媒体を含み得る。

40

【0242】

限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROMもしくは他の光ディスクストレージ、磁気ディスクストレージもしくは他の磁気記憶デバイス、フラッシュメモリ、または、命令もしくはデータ構造の形態の所望のプログラムコードを記憶するために使用されコンピュータによってアクセスされ得る任意の他の媒体を備え得る。また、いかなる接続もコンピュータ可読媒体と適切に呼ばれる。たとえば、命令が、同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線

50

(DSL)、または赤外線、無線、およびマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、または他のリモートソースから送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、DSL、または赤外線、無線、およびマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。しかしながら、コンピュータ可読記憶媒体およびデータ記憶媒体は、接続、搬送波、信号、または他の一時的媒体を含まず、代わりに非一時的有形記憶媒体を対象とすることを理解されたい。本明細書で使用されるディスク(disk)およびディスク(disc)は、コンパクトディスク(disc)(CD)、レーザーディスク(登録商標)(disc)、光ディスク(disc)、デジタル多用途ディスク(disc)(DVD)、フロッピー(登録商標)ディスク(disk)、およびblu-rayディスク(disc)を含み、ディスク(disk)は通常、データを磁気的に再生し、ディスク(disc)は、レーザーを用いてデータを光学的に再生する。上記の組合せもまた、コンピュータ可読媒体の範囲内に含まれるべきである。

【0243】

本開示において説明される機能は、固定された機能および/またはプログラム可能な処理回路によって実行され得る。たとえば、命令は、固定された機能および/またはプログラム可能な処理回路によって実行され得る。そのような処理回路は、1つまたは複数のデジタル信号プロセッサ(DSP)、汎用マイクロプロセッサ、特定用途向け集積回路(ASIC)、フィールドプログラマブルロジックアレイ(FPGA)、または他の等価な集積論理回路もしくはディスクリート論理回路などの、1つまたは複数のプロセッサを含み得る。したがって、本明細書において使用される「プロセッサ」という用語は、上記の構造、または本明細書において説明される技法の実装に適した任意の他の構造のいずれかを指すことがある。加えて、いくつかの態様では、本明細書で説明される機能は、符号化および復号のために構成された専用のハードウェアモジュールおよび/もしくはソフトウェアモジュール内で与えられることがあり、または複合コーデックに組み込まれることがある。また、技法は、1つまたは複数の回路または論理要素で全体的に実装され得る。処理回路は、様々な方法で他の構成要素に結合され得る。たとえば、処理回路は、内部のデバイス相互接続、有線もしくはワイヤレスのネットワーク接続、または別の通信媒体を介して、他の構成要素に結合され得る。

【0244】

本開示の技法は、ワイヤレスハンドセット、集積回路(IC)、またはICのセット(たとえば、チップセット)を含む、多種多様なデバイスまたは装置において実施され得る。開示される技法を実行するように構成されたデバイスの機能的態様を強調するために、様々な構成要素、モジュール、またはユニットが本開示で説明されるが、それらは必ずしも異なるハードウェアユニットによる実現を必要とするとは限らない。むしろ、上で説明されたように、様々なユニットは、コーデックハードウェアユニットにおいて組み合わされてよく、または適切なソフトウェアおよび/もしくはファームウェアとともに、上で説明されたような1つまたは複数のプロセッサを含む、相互動作可能なハードウェアユニットの集合によって提供されてよい。

【0245】

様々な例が説明された。これらおよび他の例は、以下の特許請求の範囲内に入る。

【符号の説明】

【0246】

- 10 ビデオ符号化および復号システム
- 12 ソースデバイス
- 14 宛先デバイス
- 16 コンピュータ可読媒体
- 18 ビデオソース
- 19 記憶媒体
- 20 ビデオエンコーダ
- 22 出力インターフェース
- 26 入力インターフェース

10

20

30

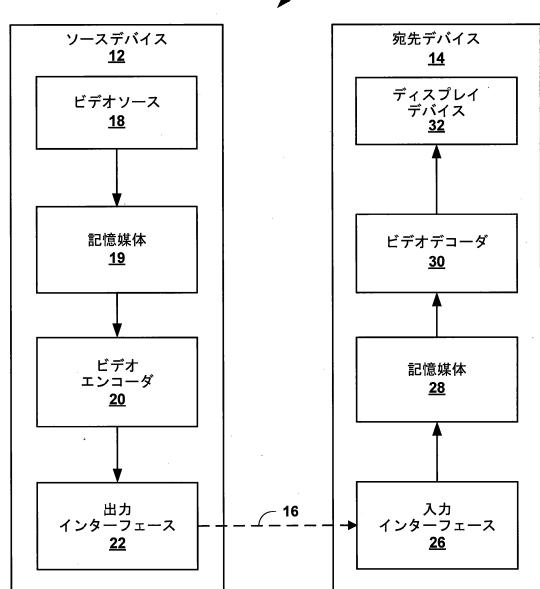
40

50

28	記憶媒体	
30	ビデオデコーダ	
32	ディスプレイデバイス	
300	右下のブロック	
302	中心のブロック	
320	現在のブロック	
322	現在のピクチャ	
323	併置されたブロック	
324	併置されたピクチャ	
326	併置されたMV	10
328	併置された参照ピクチャ	
330	TMVP	
332	現在の参照ピクチャ	
400	現在のブロック	
402	左上の制御点	
404	右上の制御点	
500	現在のブロック	
502	左上の制御点	
504	右上の制御点	
506	動きベクトル	20
508	動きベクトル	
600	現在のブロック	
620	候補リスト	
700	現在のブロック	
720	現在のブロック	
820	アフィンMVPセット候補リスト	
900	アフィンMVPセット候補リスト	
902	外挿されたMVPセット	
1000	現在のブロック	
1002	TLサブブロック	30
1004	隣接ブロックB2	
1006	TRサブブロック	
1008	隣接ブロックB1	
1100	リストX候補リスト	
1102	第1のAMVPで導出される候補	
1104	第1のAMVPで導出される動きベクトル予測子	
1106	第2のAMVPで導出される動きベクトル予測子	
1108	第2のAMVPで導出される動き候補	
1110	第3のAMVPで導出される動きベクトル予測子	
1112	第4のAMVPで導出される動きベクトル予測子	40
1120	候補リスト	
1122	第1のAMVPで導出される候補	
1124	第1のAMVPで導出される動きベクトル予測子	
1126	第2のAMVPで導出される動きベクトル予測子	
1128	第3のAMVPで導出される動きベクトル予測子	
1130	第2のAMVPで導出される動き候補	
1132	第4のAMVPで導出される動きベクトル予測子	
1134	第5のAMVPで導出される動きベクトル予測子	
1136	第6のAMVPで導出される動きベクトル予測子	
1200	予測処理ユニット	50

1201	ビデオデータメモリ	
1202	加算器	
1204	変換処理ユニット	
1206	量子化ユニット	
1208	逆量子化ユニット	
1210	逆変換処理ユニット	
1212	加算器	
1214	フィルタユニット	
1216	復号ピクチャバッファ	
1218	エントロピー符号化ユニット	10
1222	インター予測処理ユニット	
1226	イントラ予測処理ユニット	
1300	エントロピー復号ユニット	
1301	ビデオデータメモリ	
1302	予測処理ユニット	
1304	逆量子化ユニット	
1306	逆変換処理ユニット	
1308	加算器	
1310	フィルタユニット	
1312	復号ピクチャバッファ	
1314	動き補償ユニット	20
1316	イントラ予測処理ユニット	

【図1】



【図2A】

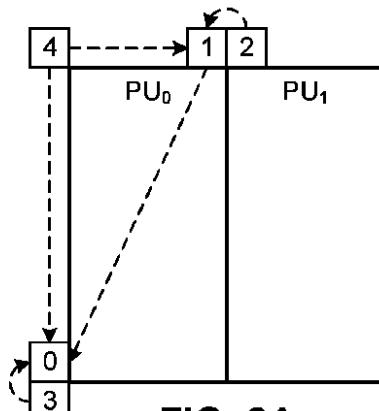


FIG. 2A

【図 2 B】

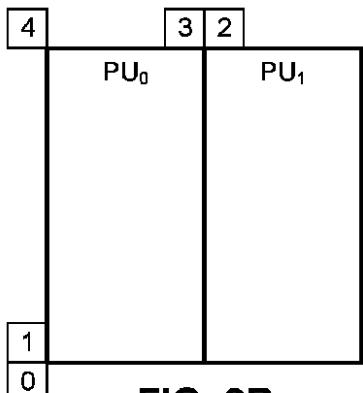
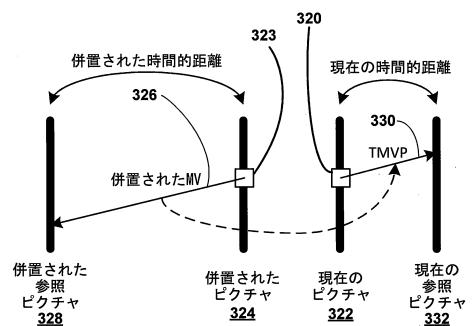
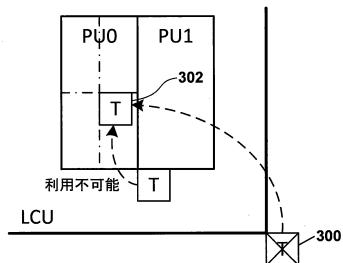


FIG. 2B

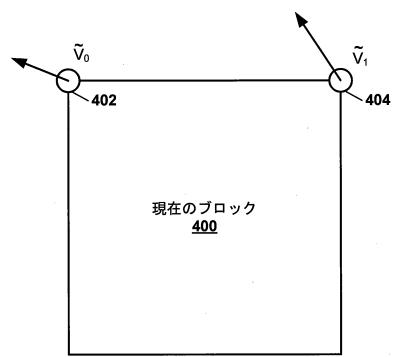
【図 3 B】



【図 3 A】



【図 4】



【図 5】

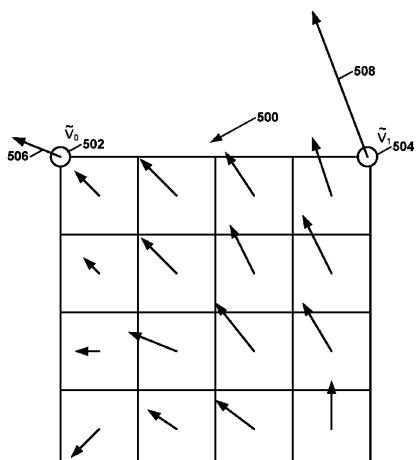


FIG. 5

【図 6 A】



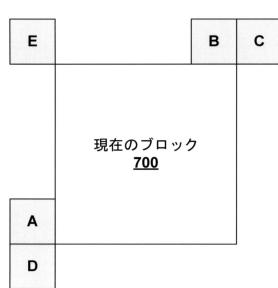
【図 6 B】

620

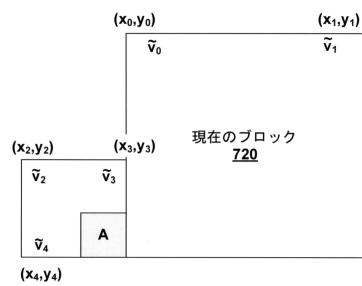
インデックス	アフィンMVPセット	
0	{V _i , V _j }	V _i ∈ {V _A , V _B , V _C }, V _j ∈ {V _D , V _E },
1	:	
2	:	
3	:	
4		
5		
6	{AMVP ₀ , AMVP ₀ }	
7	{AMVP ₁ , AMVP ₁ }	

ソートされる

【図 7 A】



【図 7 B】



【図 8 A】

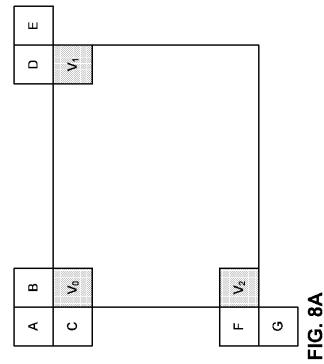
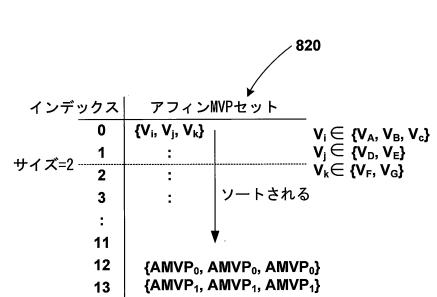
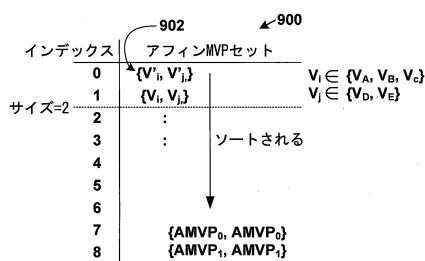


FIG. 8A

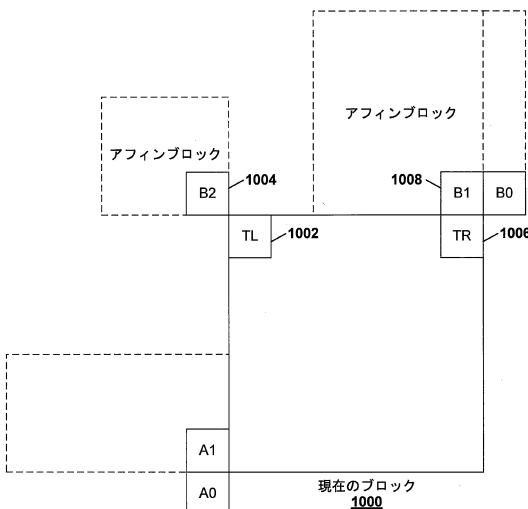
【図 8 B】



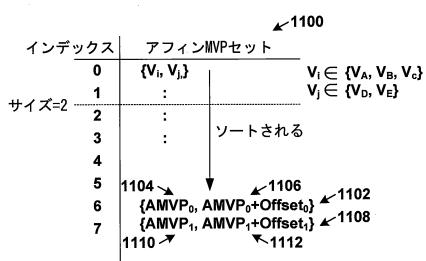
【図 9】



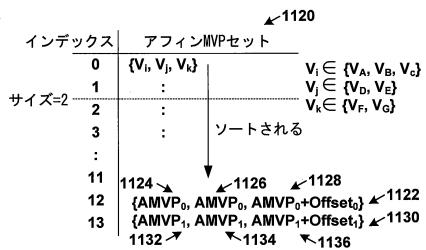
【図 10】



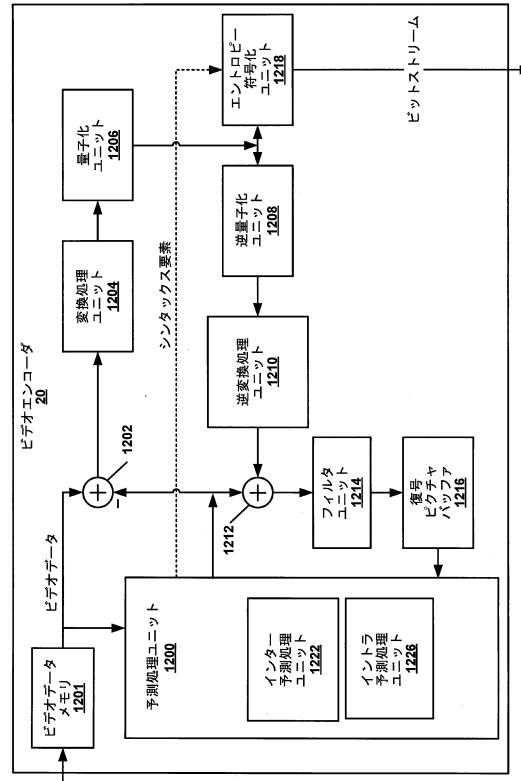
【図 1 1 A】



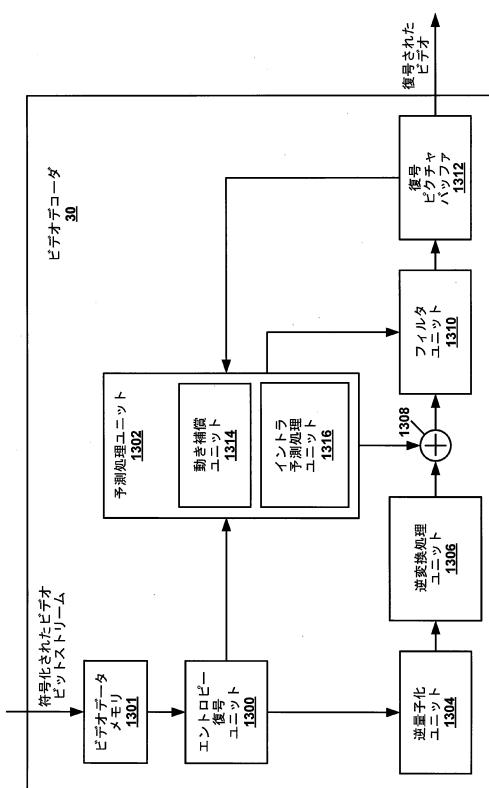
【図 1 1 B】



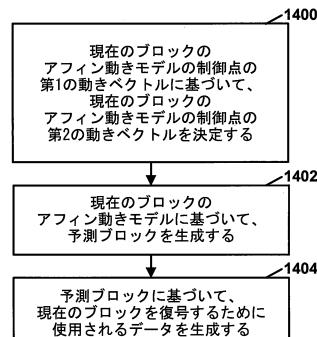
【図 1 2】



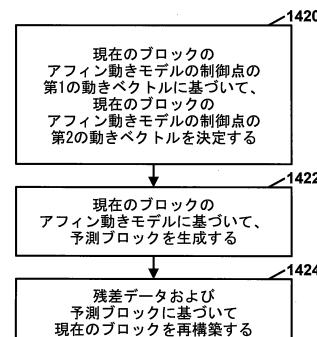
【図 1 3】



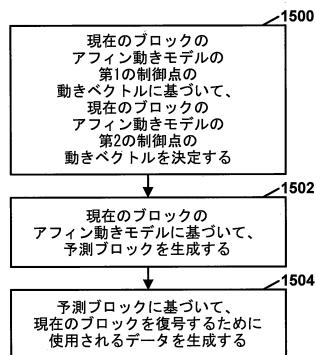
【図 1 4 A】



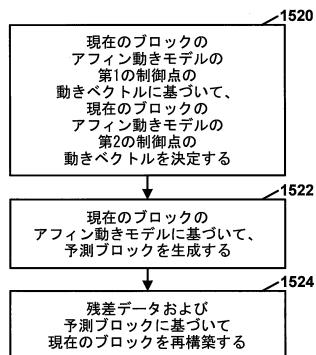
【図 1 4 B】



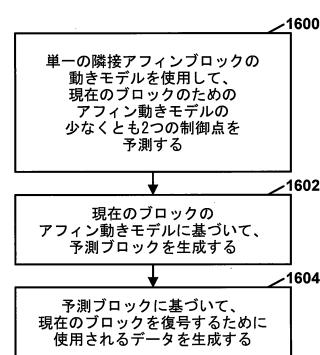
【図15A】



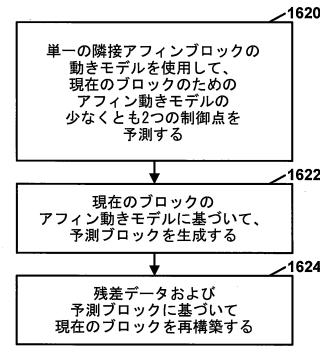
【図15B】



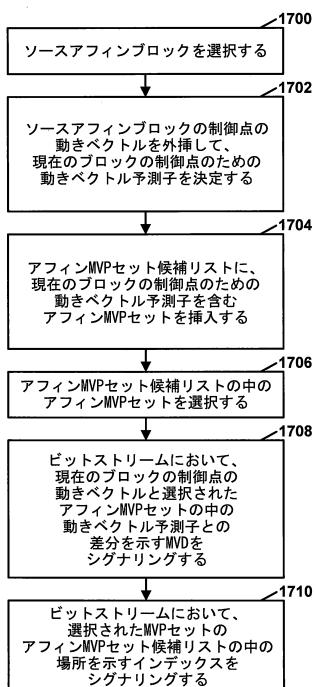
【図16A】



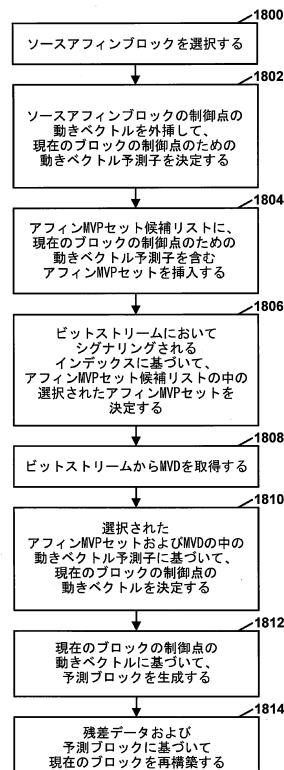
【図16B】



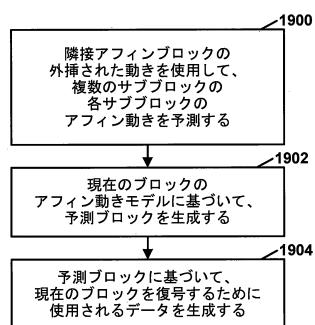
【図17】



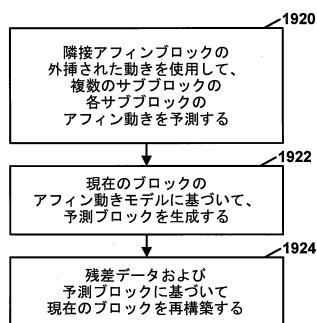
【図18】



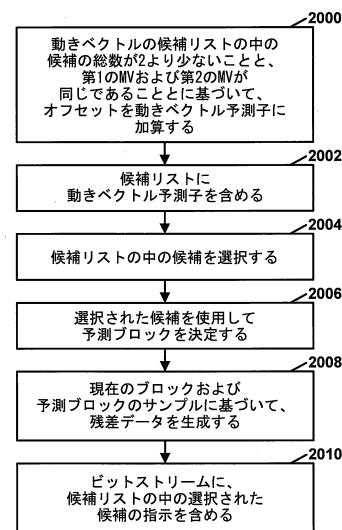
【図19A】



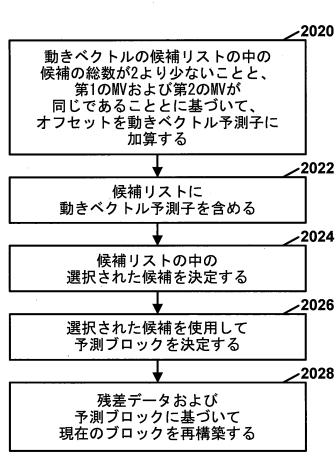
【図19B】



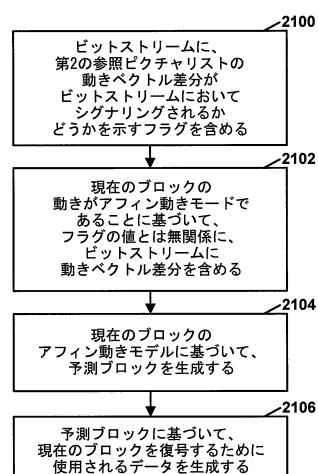
【図20A】



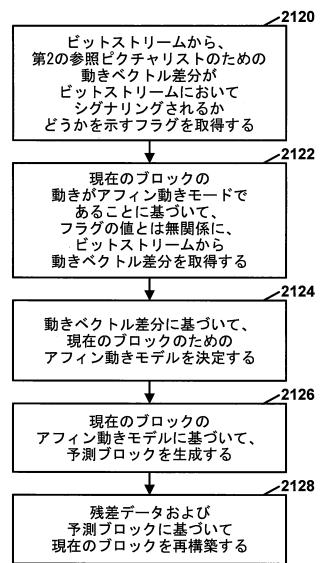
【図20B】



【図21A】



【図 2 1 B】



フロントページの続き

早期審査対象出願

(72)発明者 ウェイ - ジュン・チエン

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ
ヴ・5775

(72)発明者 リ・ジャン

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ
ヴ・5775

(72)発明者 ユ・チェン・スン

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ
ヴ・5775

(72)発明者 ジエンレ・チェン

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ
ヴ・5775

(72)発明者 マルタ・カルチェヴィッチ

アメリカ合衆国・カリフォルニア・92121-1714・サン・ディエゴ・モアハウス・ドライ
ヴ・5775

審査官 鉢呂 健

(56)参考文献 国際公開第2017/130696 (WO, A1)

特表2018-511997 (JP, A)

CHEN, Jianle et al., Algorithm Description of Joint Exploration Test Model 3, Joint Video Exploration Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 3rd Meeting: Geneva, CH, 26 May - 1 June 2016, [JVET-C1001_v3], 2016年 7月 6日, JVET-C 1001 (version 3), pp. 16-18

(58)調査した分野(Int.Cl., DB名)

H04N 19/00 - 19/98