

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7590337号
(P7590337)

(45)発行日 令和6年11月26日(2024.11.26)

(24)登録日 令和6年11月18日(2024.11.18)

(51)国際特許分類

F I

H 04 N	19/105 (2014.01)	H 04 N	19/105
H 04 N	19/137 (2014.01)	H 04 N	19/137
H 04 N	19/176 (2014.01)	H 04 N	19/176
H 04 N	19/70 (2014.01)	H 04 N	19/70

請求項の数 18 (全87頁)

(21)出願番号	特願2021-556816(P2021-556816)
(86)(22)出願日	令和2年3月20日(2020.3.20)
(65)公表番号	特表2022-527701(P2022-527701)
A)	
(43)公表日	令和4年6月3日(2022.6.3)
(86)国際出願番号	PCT/CN2020/080412
(87)国際公開番号	WO2020/187316
(87)国際公開日	令和2年9月24日(2020.9.24)
審査請求日	令和3年11月19日(2021.11.19)
審判番号	不服2023-19912(P2023-19912/J 1)
審判請求日	令和5年11月24日(2023.11.24)
(31)優先権主張番号	62/821,440
(32)優先日	平成31年3月20日(2019.3.20)
(33)優先権主張国・地域又は機関	最終頁に続く

(73)特許権者	504161984 ホアウェイ・テクノロジーズ・カンパニー・リミテッド 中華人民共和国・518129・グアン ドン・シェンツェン・ロンガン・ディス トリクト・バンティアン・(番地なし) ・ホアウェイ・アドミニストレーション ・ビルディング
(74)代理人	100110364 弁理士 実広 信哉
(74)代理人	100133569 弁理士 野村 進
(72)発明者	ファンパン・チェン 中華人民共和国・518129・グアン ドン・シェンツェン・ロンガン・ディス 最終頁に続く

(54)【発明の名称】 アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化のための方法および装置

(57)【特許請求の範囲】**【請求項 1】**

アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PROF)のための方法であって、

前記アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、前記アフィンコーディングされたブロックの前記現在のサブブロックに対するPROFプロセスを行うステップを含み、PROFを適用するための複数の制約条件は、前記アフィンコーディングされたブロックに対して満たされておらず、

前記アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前記ステップは、

前記現在のサブブロックの現在のサンプルのデルタ予測値を取得するために前記現在のサブブロックに対するオプティカルフロー処理を行うステップと、

前記現在のサンプルの前記デルタ予測値および前記現在のサブブロックの前記現在のサンプルの予測サンプル値に基づいて、前記現在のサンプルの洗練化された予測サンプル値を取得するステップと

を含み、

PROFを適用するための前記複数の制約条件は、

PROFが前記アフィンコーディングされたブロックを含むピクチャに対して無効であることを第1の指示情報が示すこと、または、前記PROFが前記アフィンコーディングされたブロックを含むピクチャに関連付けられるライスに対して無効であることを第1の指示情

報が示すこと、および、

前記アフィンコーディングされたブロックの区分なしを第2の指示情報が示すことを含み

前記現在のサブブロックの現在のサンプルのデルタ予測値を取得するために前記現在のサブブロックに対するオプティカルフロー処理を行う前記ステップは、

第2の予測行列を取得するステップであって、前記第2の予測行列は、前記現在のサブブロックの動き情報に基づいて生成される、ステップと、

前記第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を生成するステップであって、前記水平予測勾配行列および前記垂直予測勾配行列は同じサイズを有し、前記第2の予測行列のサイズは前記水平予測勾配行列および前記垂直予測勾配行列の前記サイズ以上である、ステップと、

前記水平予測勾配行列の中の前記現在のサンプルの水平予測勾配値、前記垂直予測勾配行列の中の前記現在のサンプルの垂直予測勾配値、および前記現在のサブブロックの前記現在のサンプルの動きベクトルと前記現在のサブブロックの中心のサンプルの動きベクトルとの差分に基づいて、前記現在のサブブロックの前記現在のサンプルのデルタ予測値を計算するステップとを含む、

方法。

【請求項 2】

前記アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前記ステップの前に、

PROFを適用するための前記複数の制約条件が前記アフィンコーディングされたブロックに対して満たされていないと決定するステップをさらに含む、請求項1に記載の方法。

【請求項 3】

前記第2の指示情報は、変数fallbackModeTriggeredであり、前記変数fallbackModeTriggeredが1に設定されることは前記アフィンコーディングされたブロックの区分なしを示すことである、請求項1または2に記載の方法。

【請求項 4】

第2の予測行列を取得する前記ステップは、

前記現在のサブブロックの動き情報に基づいて第1の予測行列を生成するステップであって、前記第1の予測行列の要素は前記現在のサブブロックの予測サンプル値に対応する、ステップ、および前記第1の予測行列に基づいて前記第2の予測行列を生成するステップ、または、

前記現在のサブブロックの前記動き情報に基づいて前記第2の予測行列を生成するステップを含む、請求項1に記載の方法。

【請求項 5】

前記第2の予測行列の要素は $I_1(p,q)$ によって表され、pの値の範囲は[-1, sbW]であり、qの値の範囲は[-1, sbH]であり、

前記水平予測勾配行列の要素は、 $X(i,j)$ によって表され、前記アフィンコーディングされたブロックの中の前記現在のサブブロックのサンプル(i,j)に対応し、iの値の範囲は[0, sbW-1]であり、jの値の範囲は[0, sbH-1]であり、

前記垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、前記アフィンコーディングされたブロックの中の前記現在のサブブロックのサンプル(i,j)に対応し、iの値の範囲は[0, sbW-1]であり、jの値の範囲は[0, sbH-1]であり、

sbW は前記アフィンコーディングされたブロックの中の前記現在のサブブロックの幅を表し、 sbH は前記アフィンコーディングされたブロックの中の前記現在のサブブロックの高さを表す、請求項1から4のいずれか一項に記載の方法。

【請求項 6】

前記アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前記ステップの前に、

前記現在のサブブロックの予測サンプル値を取得するために、前記アフィンコーディン

10

20

30

40

50

グされたブロックの前記現在のサブブロックに対するサブブロックベースのアフィン動き補償を行うステップをさらに含む、請求項1から5のいずれか一項に記載の方法。

【請求項 7】

アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PROF)のための方法であって、

前記アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、前記アフィンコーディングされたブロックの前記現在のサブブロックに対するPROFプロセスを行うステップを含み、複数のオプティカルフロー決定条件は、前記アフィンコーディングされたブロックに対して満たされており、

前記アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前記ステップは、

前記現在のサブブロックの現在のサンプルのデルタ予測値を取得するために前記現在のサブブロックに対するオプティカルフロー処理を行うステップと、

前記現在のサンプルの前記デルタ予測値および前記現在のサブブロックの前記現在のサンプルの予測サンプル値に基づいて、前記現在のサンプルの洗練化された予測サンプル値を取得するステップと

を含み、

前記複数のオプティカルフロー決定条件は、

PROFが前記アフィンコーディングされたブロックを含むピクチャに対して有効であることを第1の指示情報が示すこと、または、PROFが前記アフィンコーディングされたブロックを含むピクチャに関連付けられるスライスに対して有効であることを第1の指示情報が示すこと、および、

前記アフィンコーディングされたブロックに区分が適用されることを第2の指示情報が示すことを含み、

前記現在のサブブロックの現在のサンプルのデルタ予測値を取得するために前記現在のサブブロックに対するオプティカルフロー処理を行う前記ステップは、

第2の予測行列を取得するステップであって、前記第2の予測行列は、前記現在のサブブロックの動き情報に基づいて生成される、ステップと、

前記第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を生成するステップであって、前記水平予測勾配行列および前記垂直予測勾配行列は同じサイズを有し、前記第2の予測行列のサイズは前記水平予測勾配行列および前記垂直予測勾配行列の前記サイズ以上である、ステップと、

前記水平予測勾配行列の中の前記現在のサンプルの水平予測勾配値、前記垂直予測勾配行列の中の前記現在のサンプルの垂直予測勾配値、および前記現在のサブブロックの前記現在のサンプルの動きベクトルと前記サブブロックの中心のサンプルの動きベクトルとの差分に基づいて、前記現在のサブブロックの前記現在のサンプルのデルタ予測値を計算するステップとを含む、

方法。

【請求項 8】

前記アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前記ステップの前に、

前記複数のオプティカルフロー決定条件が前記アフィンコーディングされたブロックに対して満たされていると決定するステップをさらに含む、請求項7に記載の方法。

【請求項 9】

前記第2の指示情報は、変数fallbackModeTriggeredであり、前記変数fallbackModeTriggeredが0に設定されることとは、前記アフィンコーディングされたブロックが区分されるべきであることを示す、請求項7または8に記載の方法。

【請求項 10】

第2の予測行列を取得する前記ステップは、

前記現在のサブブロックの動き情報に基づいて第1の予測行列を生成するステップであ

10

20

30

40

50

つて、前記第1の予測行列の要素は前記現在のサブプロックの予測サンプル値に対応する、ステップ、および前記第1の予測行列に基づいて前記第2の予測行列を生成するステップ、または、

前記現在のサブプロックの前記動き情報に基づいて前記第2の予測行列を生成するステップを含む、請求項7に記載の方法。

【請求項 1 1】

前記第2の予測行列の要素は $I_1(p,q)$ によって表され、 p の値の範囲は $[-1, sbW]$ であり、 q の値の範囲は $[-1, sbH]$ であり、

前記水平予測勾配行列の要素は、 $X(i,j)$ によって表され、前記アフィンコーディングされたブロックの中の前記現在のサブプロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

前記垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、前記アフィンコーディングされたブロックの中の前記現在のサブプロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

sbW は前記アフィンコーディングされたブロックの中の前記現在のサブプロックの幅を表し、 sbH は前記アフィンコーディングされたブロックの中の前記現在のサブプロックの高さを表す、請求項7から10のいずれか一項に記載の方法。

【請求項 1 2】

前記アフィンコーディングされたブロックの現在のサブプロックに対するPROFプロセスを行う前記ステップの前に、

前記現在のサブプロックの予測サンプル値を取得するために、前記アフィンコーディングされたブロックの前記現在のサブプロックに対するサブプロックベースのアフィン動き補償を行うステップをさらに含む、請求項7から11のいずれか一項に記載の方法。

【請求項 1 3】

請求項1から12のいずれか一項に記載の方法を行うための処理回路を備える、エンコーダ(20)。

【請求項 1 4】

請求項1から12のいずれか一項に記載の方法を行うための処理回路を備える、デコーダ(30)。

【請求項 1 5】

コンピュータに、請求項1から12のいずれか一項に記載の方法を行わせるためのコンピュータプログラム。

【請求項 1 6】

デコーダであって、

1つまたは複数のプロセッサと、

前記プロセッサに結合され、前記プロセッサによる実行のためのプログラミングを記憶する、非一時的コンピュータ可読記憶媒体とを備え、前記プログラミングは、前記プロセッサによって実行されると、請求項1から12のいずれか一項に記載の方法を行うように前記デコーダを構成する、デコーダ。

【請求項 1 7】

エンコーダであって、

1つまたは複数のプロセッサと、

前記プロセッサに結合され、前記プロセッサによる実行のためのプログラミングを記憶する、非一時的コンピュータ可読記憶媒体とを備え、前記プログラミングは、前記プロセッサによって実行されると、請求項1から12のいずれか一項に記載の方法を行うように前記エンコーダを構成する、エンコーダ。

【請求項 1 8】

コンピュータデバイスによって実行されると、前記コンピュータデバイスに請求項1から12のいずれか一項に記載の方法を行わせるプログラムコードを担持する、非一時的コンピュータ可読媒体。

10

20

30

40

50

【発明の詳細な説明】**【技術分野】****【0001】****関連出願の相互参照**

本特許出願は、2019年3月20日に出願された米国仮特許出願第62/821,440号の優先権、および2019年4月28日に出願された米国仮特許出願第62/839,765号に基づく優先権を主張する。前述の特許出願の開示は、その全体が参照により本明細書に組み込まれる。

【0002】

本開示の実施形態は、概してピクチャ処理の分野に関し、より具体的には、1つまたは複数の制約が必要とされるときにオプティカルフローを用いてサブブロックベースのアフィン動き補償された予測を洗練化するための方法に関する。

10

【背景技術】**【0003】**

ビデオコーディング(ビデオ符号化および復号)は、広範囲のデジタルビデオ用途、たとえばデジタルTV放送、インターネットおよびモバイルネットワークを介したビデオ送信、ビデオチャットなどのリアルタイム会話アプリケーション、ビデオ会議、DVDおよびBlu-rayディスク、ビデオコンテンツ取得および編集システム、ならびにセキュリティ用途のカムコーダにおいて使用される。

【0004】

比較的短いビデオであっても、それを描画するために必要とされるビデオデータの量はかなり多いことがあり、これは、帯域幅容量が限られている通信ネットワークを介してデータがストリーミングまたは他の方法で通信されることになるとき、困難さをもたらすことがある。したがって、ビデオデータは一般に、現代の遠隔通信ネットワークを介して通信される前に圧縮される。メモリリソースは限られていることがあるので、ビデオがストレージデバイスに記憶されるときには、ビデオのサイズも問題になり得る。ビデオ圧縮デバイスはしばしば、送信または記憶の前にソースにおいてソフトウェアおよび/またはハードウェアを使用してビデオデータをコーディングし、それにより、デジタルビデオ画像を表現するために必要とされるデータの量を減らす。次いで、圧縮されたデータは、ビデオデータを復号するビデオ解凍デバイスによって宛先において受信される。ネットワーククリソースが限られており、また、より高いビデオ品質の需要が高まり続けているので、ピクチャ品質をほとんどまたはまったく犠牲にすることなく圧縮比を改善する、改善された圧縮および解凍の技法が望まれる。

20

【0005】

最近、アフィンツールがVersatile Video Codingに導入され、理論上は、コーディングブロックの中の各サンプルの動きベクトルを導出するために、アフィン動きモデルパラメータを使用することができる。しかしながら、サンプルベースのアフィン動き補償された予測を生成することは非常に複雑であるため、サブブロックベースのアフィン動き補償方法が用いられる。この方法では、コーディングブロックはサブブロックへと分割され、サブブロックの各々に、アフィン動きモデルパラメータから導出された動きベクトル(MV)が割り当てられる。しかしながら、それはサブブロックベースの予測により、予測の正確さを失う。したがって、コーディングの複雑さと予測の正確さとの間で、良好なトレードオフを達成する必要がある。

30

【発明の概要】**【課題を解決するための手段】****【0006】**

本出願の実施形態は、独立請求項による符号化および復号のための装置と方法を提供する。本出願の実施形態は、複雑さとサブブロックベースのアフィン予測の正確さとの間で良好なトレードオフが達成され得るように、アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PRF)のための装置および方法を提供する。

40

【0007】

50

実施形態は、独立請求項の特徴によって、および従属請求項の特徴による実施形態のさらに有利な実装形態によって定義される。

【 0 0 0 8 】

特定の実施形態が添付の独立請求項において概説され、他の実施形態は従属請求項において概説される。

【 0 0 0 9 】

前述の目的および他の目的は、独立請求項の主題によって達成される。さらなる実装形式が、従属請求項、説明、および図面から明らかである。

【 0 0 1 0 】

第1の態様によれば、本発明は、アフィンコーディングされたブロック(すなわち、アフィンツールを使用して符号化または復号されたブロック)に対するオプティカルフローを用いた予測洗練化(PROP:prediction refinement with optical flow)のための方法に関する。方法は、アフィンコーディングされたブロック内のサンプルのサブブロックに適用される。方法は、符号化装置または復号装置によって行われる。方法は、

アフィンコーディングされたブロックの現在のサブブロック(各サブブロックなど)の洗練化された予測サンプル値(すなわち、最終的な予測サンプル値)を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するPROPプロセスを行うステップを含んでもよく、PROPを適用するための複数の制約条件は、アフィンコーディングされたブロックに対して満たされず、または満足されず、

アフィンコーディングされたブロックの現在のサブブロックに対するPROPプロセスを行うステップは、現在のサブブロックの現在のサンプルのデルタ予測値を取得するために現在のサブブロックに対するオプティカルフロー処理を実行するステップ、ならびに現在のサンプルのデルタ予測値および現在のサブブロックの現在のサンプルの予測サンプル値に基づいて、現在のサンプルの洗練化された予測サンプル値を取得するステップ(現在のサブブロックのデルタ予測値を取得するために現在のサブブロックに対するオプティカルフロー処理を実行するステップ、ならびに現在のサブブロックのデルタ予測値および現在のサブブロックの予測サンプル値に基づいて現在のサブブロックの洗練化された予測サンプル値を取得するステップ)を含む。アフィンコーディングされたブロックの各サブブロックの洗練化された予測サンプル値が生成されるとき、アフィンコーディングされたブロックの洗練化された予測サンプル値が自然に生成されることが理解され得る。

【 0 0 1 1 】

したがって、コーディングの複雑さと予測の正確さとのより良好なトレードオフの達成を可能にする、改善された方法が提供される。ピクセル/サンプルレベルの粒度でオプティカルフローを用いたサブブロックベースのアフィン動き補償された予測を洗練化するために、オプティカルフローを用いた予測洗練化(PROP)プロセスが条件的に実行される。これらの条件は、予測の正確さを改善できるときにのみPROPに関わる計算が発生することを確実にし、それにより計算の複雑さの不必要的増大を減らす。したがって、本明細書において開示される技術によって達成される有益な効果は、コーディング方法の全体的な圧縮性能を高める。

【 0 0 1 2 】

本開示において使用される「ブロック」、「コーディングブロック」、または「画像ブロック」という用語は、変換ユニット(TU)、予測ユニット(PU)、コーディングユニット(CU)などを含み得ることに留意されたい。VVCでは、変換ユニットおよびコーディングユニットは、TUの傾きまたはサブブロック変換(SBT)が使用される少数のシナリオを除き、概ね揃っている。「ブロック」、「画像ブロック」、「コーディングブロック」、および「ピクチャブロック」という用語は、本明細書では入れ替え可能に使用され得ることが理解され得る。「アフィンブロック」、「アフィンピクチャブロック」、「アフィンコーディングされたブロック」、および「アフィン動きブロック」という用語は、本明細書では入れ替え可能に使用され得る。「サンプル」および「ピクセル」という用語は、本開示において互いに入れ替え可能に使用され得る。「予測サンプル値」および「予測ピクセル値」

10

20

30

40

50

という用語は、本開示において互いに入れ替え可能に使用され得る。「サンプル位置」および「ピクセル位置」という用語は、本開示において互いに入れ替え可能に使用され得る。

【0013】

第1の態様自体による方法の可能な実装形式において、アフィンコーディングされたブロックの現在のサブプロックに対するPROFプロセスを行う前に、方法はさらに、PROFを適用するための複数の制約条件がアフィンコーディングされたブロックに対して満たされていないと決定するステップを含む。

【0014】

第1の態様自体による方法の可能な実装形式において、PROFを適用するための複数の制約条件は、PROFがアフィンコーディングされたブロックを含むピクチャに対して無効であることを第1の指示情報が示すこと、またはPROFがアフィンコーディングされたブロックを含むピクチャに関連付けられるスライスに対して無効であることを第1の指示情報が示すこと、および、第2の指示情報がアフィンコーディングされたブロックの区分なしを示すこと、すなわち変数fallbackModeTriggeredが1に設定されることを含む。変数fallbackModeTriggeredが1に設定されるとき、アフィンコーディングされたブロックの区分は必要とされず、すなわち、アフィンコーディングされたブロックの各サブブロックは同じ動きベクトルを有することが理解され得る。これは、アフィンコーディングされたブロックが並進運動のみを有することを示す。変数fallbackModeTriggeredが0に設定されるとき、アフィンコーディングされたブロックの区分が必要とされ、すなわち、アフィンコーディングされたブロックの各サブブロックはそれぞれの動きベクトルを有する。これは、アフィンコーディングされたブロックが非並進運動を有することを示す。

10

【0015】

本開示では、アフィンコーディングされたブロックに対するいくつかの事例または状況において、PROFが適用されないことが許容される。PROFを適用するための制約に従って、それらの事例または状況が決定される。こうして、コーディングの複雑さと予測の正確さとの間で、より良好なトレードオフを達成することができる。

【0016】

第1の態様の任意の先行する実装形態または第1の態様自体による方法の可能な実装形式において、現在のサブプロックの現在のサンプルのデルタ予測値を取得するために現在のサブプロックに対するオプティカルフロー処理を実行するステップは、

30

第2の予測行列を取得するステップであって(ある例では、第2の予測行列は、現在のサブプロックの予測サンプル値に対応する第1の予測行列に基づいて生成される。ここで、現在のサブプロックの予測サンプル値は、現在のサブプロックに対するサブプロックベースのアフィン動き補償を行うことによって取得され得る。)、第2の予測行列のサイズは第1の予測行列のサイズより大きく(たとえば、第1の予測行列は $sbWidth * sbHeight$ のサイズを有し、第2の予測行列は $(sbWidth+2) * (sbHeight+2)$ のサイズを有し、変数 $sbWidth$ および $sbHeight$ はそれぞれ現在のサブプロックの幅および高さを表す)、すなわち、第2の予測行列を取得するステップは、現在のサブプロックの動き情報に基づいて第1の予測行列を生成し、ただし第1の予測行列の要素は現在のサブプロックの予測サンプル値に対応し、第2の予測行列を取得するステップは、さらに第1の予測行列に基づいて第2の予測行列を生成し、または、現在のサブプロックの動き情報に基づいて第2の予測行列を生成するステップと、

40

第2の予測行列に基づいて、水平予測勾配行列および垂直予測勾配行列を生成するステップであって、第2の予測行列のサイズは、水平予測勾配行列および垂直予測勾配行列のサイズ以上である(たとえば、水平予測勾配行列または垂直予測勾配行列は $sbWidth * sbHeight$ のサイズを有し、第2の予測行列は $(sbWidth+2) * (sbHeight+2)$ のサイズを有する)、ステップと、

水平予測勾配行列の中の現在のサンプルの水平予測勾配値、垂直予測勾配行列の中の現在のサンプルの垂直予測勾配値、および現在のサブプロックの現在のサンプルの動きベクトルとサブプロックの中心サンプルの動きベクトルとの差分(MVD)に基づいて、現在のサブ

50

ブロックの現在のサンプルのデルタ予測値($\Delta(i,j)$)を計算するステップとを含む。MVDは水平成分および垂直成分を有することが理解され得る。水平予測勾配行列の中の現在のサンプルの水平予測勾配値は、MVDの水平成分に対応し、垂直予測勾配行列の中の現在のサンプルの垂直予測勾配値は、MVDの垂直成分に対応する。

【0017】

アフィンブロックは、ビデオ信号のピクチャのコーディングブロックまたは復号ブロックであり得ることに留意されたい。アフィンコーディングされたブロックの現在のサブブロックは、たとえば 4×4 ブロックである。ルマ位置(x_{Cb}, y_{Cb})は、現在のピクチャの左上サンプルに対する相対的なアフィンコーディングされたブロックの左上サンプルの位置を表記する。現在のサブブロックのサンプルは、ピクチャの左上サンプルに関する(またはそれに対して相対的な)サンプルの絶対位置、たとえば(x, y)、または、(他の座標と組み合わせて)サブブロックの左上サンプルに関するサンプルの相対位置、たとえば($x_{Sb}+i, y_{Sb}+j$)を使用して参照され得る。ここで、(x_{Sb}, y_{Sb})は、ピクチャの左上サンプルに関するサブブロックの左上サンプルの座標である。

10

【0018】

第1の予測行列は行および列を含む2次元アレイであってもよく、アレイの要素は(i, j)を使用して参照されてもよく、 i は水平/行インデックスであり、 j は垂直/列インデックスである。 i および j の範囲は、たとえば $i=0..sbWidth-1$ 、および $j=0..sbHeight-1$ であり得る。ここで、 $sbWidth$ はサブブロックの幅を示し、 $sbHeight$ はサブブロックの高さを示す。いくつかの例では、第1の予測行列のサイズは、現在のブロックのサイズと同じである。たとえば、第1の予測行列のサイズは 4×4 であってもよく、現在のブロックは 4×4 のサイズを有する。

20

【0019】

第2の予測行列は行および列を含む2次元アレイであってもよく、アレイの要素は(i, j)を使用して参照されてもよく、 i は水平/行インデックスであり、 j は垂直/列インデックスである。 i および j の範囲は、たとえば $i=-1..sbWidth$ 、および $j=-1..sbHeight$ であり得る。ここで、 $sbWidth$ はサブブロックの幅を示し、 $sbHeight$ はサブブロックの高さを示す。いくつかの例では、第2の予測行列のサイズは第1の予測行列のサイズより大きい。すなわち、第2の予測行列のサイズは現在のブロックのサイズより大きくてもよい。たとえば、第2の予測行列のサイズは $(sbWidth+2)*(sbHeight+2)$ であり得るが、現在のブロックは $sbWidth*sbHeight$ のサイズを有する。たとえば、第2の予測行列のサイズは 6×6 であってもよく、現在のブロックは 4×4 のサイズを有する。

30

【0020】

水平予測勾配行列および垂直予測勾配行列は、行および列を含む任意の2次元アレイであってもよく、アレイの要素は(i, j)を使用して参照されてもよく、 x は水平/行インデックスであり、 y は垂直/列インデックスである。 i および j の範囲は、たとえば $i=0..sbWidth-1$ および $j=0..sbHeight-1$ であり得る。 $sbWidth$ はサブブロックの幅を示し、 $sbHeight$ はサブブロックの高さを示す。いくつかの例では、水平予測勾配行列および垂直予測勾配行列のサイズは、現在のブロックのサイズと同じである。たとえば、水平予測勾配行列および垂直予測勾配行列のサイズは 4×4 であってもよく、現在のブロックは 4×4 のサイズを有する。

40

【0021】

水平予測勾配行列における要素の位置(x, y)が垂直予測勾配行列における要素の位置(p, q)と同じである、すなわち(x, y)=(p, q)である場合、水平予測勾配行列の要素は、垂直予測勾配行列の要素に対応する。

【0022】

したがって、PROFプロセスは、メモリアクセス帯域幅を増やすことなく(第2の予測行列が第1の予測行列または現在のサブブロックの(元の)予測サンプル値に基づくことにより)、サンプルレベル粒度でオプティカルフローを用いたサブブロックベースのアフィン動き補償された予測を洗練化することが可能になり、それにより動き補償のより高い粒度を達

50

成する。

【 0 0 2 3 】

第1の態様の任意の先行する実装形態または第1の態様自体による方法の可能な実装形式において、現在のサンプルを含む現在のサンプルユニット(たとえば、 2×2 サンプルブロック)の動きベクトルとサブブロックの中心サンプルの動きベクトルとの動きベクトル差分が、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心サンプルの動きベクトルとの差分として使用される。ここで、サブブロックの中心サンプルの動きベクトルは、現在のサンプル(i,j)が属するサブブロックのMV(すなわち、サブブロックMV)として理解され得る。動きベクトル差分を計算するために 2×2 サンプルブロックなどのサンプルユニットを使用することによって、処理オーバーヘッドと予測の正確さのバランスを取ることが可能になる。第1の態様の任意の先行する実装形態または第1の態様自体による方法の可能な実装形式において、第2の予測行列の要素は $I_1(p,q)$ によって表され、pの値の範囲は[-1, sbW]であり、qの値の範囲は[-1, sbH]であり、

水平予測勾配行列の要素は、 $X(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル(i,j)に対応し、iの値の範囲は[0, sbW-1]であり、jの値の範囲は[0, sbH-1]であり、

垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル(i,j)に対応し、iの値の範囲は[0, sbW-1]であり、jの値の範囲は[0, sbH-1]であり、

sbW はアフィンコーディングされたブロックの中の現在のサブブロックの幅を表し、 sbH はアフィンコーディングされたブロックの中の現在のサブブロックの高さを表す。

別の表現方式では、第2の予測行列の要素は $I_1(p,q)$ によって表され、pの値の範囲は[0, subW+1]であり、qの値の範囲は[0, subH+1]である。

水平予測勾配行列の要素は、 $X(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル(i,j)に対応し、iの値の範囲は[1, sbW]であり、jの値の範囲は[1, sbH]であり、

垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル(i,j)に対応し、iの値の範囲は[1, sbW]であり、jの値の範囲は[1, sbH]であり、

sbW はアフィンコーディングされたブロックの中の現在のサブブロックの幅を表し、 sbH はアフィンコーディングされたブロックの中の現在のサブブロックの高さを表す。

pは[0, subW+1]からの値を有しqは[0, subH+1]からの値を有するので、左上サンプル(または座標の原点)は(1,1)に位置し、一方、pは[-1, subW]からの値を有しqは[-1, subH]からの値を有するので、左上サンプル(または座標の原点)は(0,0)に位置することが理解され得る。

【 0 0 2 4 】

第1の態様の任意の先行する実装形態または第1の態様自体による方法の可能な実装形式において、アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前に、方法はさらに、現在のサブブロックの(元のまたは洗練化されることになる)予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するサブブロックベースのアフィン動き補償を行うステップを含む。

【 0 0 2 5 】

第2の態様によれば、本発明はアフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PROF)のための方法に関する。この方法は、

アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値(すなわち、最終的な予測サンプル値)を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行うステップを含み、複数のオプティカルフロー決定条件がアフィンコーディングされたブロックに対して満たされ、ここで、複数のオプティカルフロー決定条件が満たされることは、PROFを適用するためのすべての制約が満たされてはいないことを意味し、

10

20

30

40

50

アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行うステップは、現在のサブブロックの現在のサンプルのデルタ予測値を取得するために現在のサブブロックに対するオプティカルフロー処理を実行するステップと、現在のサンプルのデルタ予測値および現在のサブブロックの現在のサンプルの(元のまたは洗練化されることになる)予測サンプル値に基づいて、現在のサンプルの洗練化された予測サンプル値を取得するステップとを含む。

【 0 0 2 6 】

したがって、コーディングの複雑さと予測の正確さとの良好なトレードオフの達成を可能にする、改善された方法が提供される。ピクセル/サンプルレベルの粒度でオプティカルフローを用いたサブブロックベースのアフィン動き補償された予測を洗練化するために、オプティカルフローを用いた予測洗練化(PROF)プロセスが条件的に実行される。これらの条件は、予測の正確さを改善できるときにのみPROFに関わる計算が発生することを確実にし、それにより計算の複雑さの不必要的増大を減らす。したがって、本明細書において開示される技術によって達成される有益な効果は、コーディング方法の全体的な圧縮性能を高める。

10

【 0 0 2 7 】

第2の態様自体による方法の可能な実装形式において、アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行う前に、方法はさらに、複数のオプティカルフロー決定条件がアフィンコーディングされたブロックに対して満たされていると決定するステップを含む。

20

【 0 0 2 8 】

第2の態様自体による方法の可能な実装形式において、複数のオプティカルフロー決定条件は、アフィンコーディングされたブロックを含むピクチャに対してPROFが有効であることを第1の指示情報が示すこと、またはアフィンコーディングされたブロックを含むピクチャに関連付けられるスライスに対してPROFが有効であることを第1の指示情報が示すこと、および、第2の指示情報が、変数fallbackModeTriggeredが0に等しく設定されることなどの、アフィンコーディングされたブロックの区分を示すことを含む。変数fallbackModeTriggeredが0に等しく設定されるとき、アフィンコーディングされたブロックの区分が必要とされ、すなわち、アフィンコーディングされたブロックの各サブブロックはそれぞれの動きベクトルを有し、それはアフィンコーディングされたブロックが非並進運動を有することを示すことが理解され得る。

30

【 0 0 2 9 】

PROFを適用するためのすべての制約が、PROFを適用するための制約の設計に従って満たされていないとき、PROFが適用され得ることが許容される。したがって、コーディングの複雑さと予測の正確さとのトレードオフが可能になる。

【 0 0 3 0 】

第2の態様の任意の先行する実装形態または第2の態様自体による方法の可能な実装形式において、現在のサブブロックの現在のサンプルのデルタ予測値を取得するために現在のサブブロックに対するオプティカルフロー処理を実行するステップは、

第2の予測行列を取得するステップであって、第2の予測行列の要素は現在のサブブロックの予測サンプル値に基づき、いくつかの例では、第2の予測行列を取得するステップは、現在のサブブロックの動き情報に基づいて第1の予測行列を生成するステップであって、第1の予測行列の要素は現在のサブブロックの予測サンプル値に対応する、ステップ、および第1の予測行列に基づいて第2の予測行列を生成するステップ、または、現在のサブブロックの動き情報に基づいて第2の予測行列を生成するステップを含む、ステップと、第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を生成するステップであって、第2の予測行列のサイズは水平予測勾配行列および垂直予測勾配行列のサイズ以上である、ステップと、

40

水平予測勾配行列の中の現在のサンプルの水平予測勾配値、垂直予測勾配行列の中の現在のサンプルの垂直予測勾配値、および現在のサブブロックの現在のサンプルの動きベクト

50

ルとサブブロックの中心サンプルの動きベクトルとの差分に基づいて、現在のサブブロックの現在のサンプルのデルタ予測値($\Delta(i,j)$)を計算するステップとを含む。

【0031】

第2の態様の任意の先行する実装形態または第2の態様自体による方法の可能な実装形式において、方法は、アフィンコーディングされたブロックの現在のサブブロックの(元の)予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するサブブロックベースのアフィン動き補償を行うステップをさらに含む。

【0032】

第2の態様の任意の先行する実装形態または第2の態様自体による方法の可能な実装形式において、現在のサンプルが属する現在のサンプルユニット(たとえば、 2×2 サンプルブロック)の動きベクトルとサブブロックの中心サンプルの動きベクトルとの動きベクトル差分が、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心サンプルの動きベクトルとの差分として使用される。

【0033】

第2の態様の任意の先行する実装形態または第2の態様自体による方法の可能な実装形式において、

第2の予測行列の要素は $I_1(p,q)$ によって表され、 p の値の範囲は $[-1, sbW]$ であり、 q の値の範囲は $[-1, sbH]$ であり、

水平予測勾配行列の要素は、 $X(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

sbW はアフィンコーディングされたブロックの中の現在のサブブロックの幅を表し、 sbH はアフィンコーディングされたブロックの中の現在のサブブロックの高さを表す。

【0034】

第3の態様によれば、本発明は、アフィンコーディングされたブロック(すなわち、アフィンツールを使用して符号化または復号されたブロック)に対するオプティカルフローを用いた予測洗練化(PROF)のための装置が提供される。装置は、符号化装置または復号装置に対応する。装置は、

PROFを適用するための複数の制約条件がアフィンコーディングされたブロックに対して満たされていないと決定するために構成される決定ユニットと、

アフィンコーディングされたブロックの現在のサブブロック(各サブブロックなど)の洗練化された予測サンプル値(すなわち、最終的な予測サンプル値)を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するPROFプロセスを行うために構成される予測処理ユニットとを含んでもよく、PROFを適用するための複数の制約条件は、アフィンコーディングされたブロックに対して満たされず、または満足されず、予測処理ユニットは、現在のサブブロックの現在のサンプルのデルタ予測値を取得するために現在のサブブロックに対するオプティカルフロー処理を実行すること、ならびに現在のサンプルのデルタ予測値および現在のサブブロックの現在のサンプルの予測サンプル値に基づいて、現在のサンプルの洗練化された予測サンプル値を取得すること(現在のサブブロックのデルタ予測値を取得するために現在のサブブロックに対するオプティカルフロー処理を実行すること、ならびに現在のサブブロックのデルタ予測値および現在のサブブロックの予測サンプル値に基づいて現在のサブブロックの洗練化された予測サンプル値を得ること)のために構成される。アフィンコーディングされたブロックの各サブブロックの洗練化された予測サンプル値が生成されるとき、アフィンコーディングされたブロックの洗練化された予測サンプル値が自然に生成されることが理解され得る。

【0035】

第3の態様自体による装置の可能な実装形式において、PROFを適用するための複数の制

10

20

30

40

50

約条件は、PROFがアフィンコーディングされたブロックを含むピクチャに対して無効であることを第1の指示情報が示すこと、またはPROFがアフィンコーディングされたブロックを含むピクチャに関連付けられるスライスに対して無効であることを第1の指示情報が示すこと、および、第2の指示情報がアフィンコーディングされたブロックの区分なしを示すこと、すなわち変数fallbackModeTriggeredが1に設定されることを含む。変数fallbackModeTriggeredが1に設定されるとき、アフィンコーディングされたブロックの区分は必要とされず、すなわち、アフィンコーディングされたブロックの各サブブロックは同じ動きベクトルを有することが理解され得る。これは、アフィンコーディングされたブロックが並進運動のみを有することを示す。変数fallbackModeTriggeredが0に設定されるとき、アフィンコーディングされたブロックの区分が必要とされ、すなわち、アフィンコーディングされたブロックの各サブブロックはそれぞれの動きベクトルを有する。これは、アフィンコーディングされたブロックが非並進運動を有することを示す。

【 0 0 3 6 】

第3の態様の任意の先行する実装形態または第3の態様自体による装置の可能な実装形式において、予測処理ユニットは、第2の予測行列を取得するために構成される(ある例では、第2の予測行列は、現在のサブブロックの予測サンプル値に対応する第1の予測行列に基づいて生成される。ここで、現在のサブブロックの予測サンプル値は、現在のサブブロックに対するサブブロックベースのアフィン動き補償を行うことによって取得され得る。)、第2の予測行列のサイズは第1の予測行列のサイズより大きく(たとえば、第1の予測行列はsbWidth*sbHeightのサイズを有し、第2の予測行列は(sbWidth+2)*(sbHeight+2)のサイズを有し、変数sbWidthおよびsbHeightはそれぞれ現在のサブブロックの幅および高さを表す)、すなわち、第2の予測行列を取得するステップは、現在のサブブロックの動き情報に基づいて第1の予測行列を生成し、ただし第1の予測行列の要素は現在のサブブロックの予測サンプル値に対応し、第2の予測行列を取得するステップは、さらに第1の予測行列に基づいて第2の予測行列を生成し、または、現在のサブブロックの動き情報に基づいて第2の予測行列を生成するステップと、

第2の予測行列に基づいて、水平予測勾配行列および垂直予測勾配行列を生成するステップであって、第2の予測行列のサイズは、水平予測勾配行列および垂直予測勾配行列のサイズ以上である(たとえば、水平予測勾配行列または垂直予測勾配行列はsbWidth*sbHeightのサイズを有し、第2の予測行列は(sbWidth+2)*(sbHeight+2)のサイズを有する)、ステップと、

水平予測勾配行列の中の現在のサンプルの水平予測勾配値、垂直予測勾配行列の中の現在のサンプルの垂直予測勾配値、および現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心サンプルの動きベクトルとの差分に基づいて、現在のサブブロックの現在のサンプルのデルタ予測値($\Delta I(i,j)$)を計算するステップとを含む。

【 0 0 3 7 】

第3の態様の任意の先行する実装形態または第3の態様自体による装置の可能な実装形式において、現在のサンプルを含む現在のサンプルユニット(たとえば、 2×2 サンプルブロック)の動きベクトルとサブブロックの中心サンプルの動きベクトルとの動きベクトル差分が、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心サンプルの動きベクトルとの差分として使用される。ここで、サブブロックの中心サンプルの動きベクトルは、現在のサンプル(i,j)が属するサブブロックのMV(すなわち、サブブロックMV)として理解され得る。動きベクトル差分を計算するために 2×2 サンプルブロックなどのサンプルユニットを使用することによって、処理オーバーヘッドと予測の正確さのバランスを取ることが可能になる。第3の態様の任意の先行する実装形態または第3の態様自体による装置の可能な実装形式において、第2の予測行列の要素は $I_1(p,q)$ によって表され、 p の値の範囲は $[-1, sbW]$ であり、 q の値の範囲は $[-1, sbH]$ であり、

水平予測勾配行列の要素は、 $X(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル(i,j)に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

10

20

30

40

50

垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、アフィンコーディングされたプロックの中の現在のサブプロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

sbW はアフィンコーディングされたプロックの中の現在のサブプロックの幅を表し、 sbH はアフィンコーディングされたプロックの中の現在のサブプロックの高さを表す。

【 0 0 3 8 】

第3の態様の任意の先行する実装形態または第3の態様自体による装置の可能な実装形式において、予測処理ユニット1503は、現在のサブプロックの(元のまたは洗練化されることになる)予測サンプル値を取得するために、アフィンコーディングされたプロックの現在のサブプロックに対するサブプロックベースのアフィン動き補償を行うために構成される。 10

【 0 0 3 9 】

第4の態様によれば、本発明はアフィンコーディングされたプロックに対するオプティカルフローを用いた予測洗練化(PROF)のための装置に関する。この装置は、

複数のオプティカルフロー決定条件がアフィンコーディングされたプロックに対して満たされていると決定するために構成される決定ユニットであって、ここで、複数のオプティカルフロー決定条件が満たされていることは、PROFを適用するためのすべての制約が満たされていないことを意味する、決定ユニットと、

アフィンコーディングされたプロックの現在のサブプロックの洗練化された予測サンプル値(すなわち、最終的な予測サンプル値)を取得するために、アフィンコーディングされたプロックの現在のサブプロックに対するPROFプロセスを行うために構成される予測処理ユニットとを備えてもよく、複数のオプティカルフロー決定条件はアフィンコーディングされたプロックに対して満たされており、予測処理ユニットは、現在のサブプロックの現在のサンプルのデルタ予測値を取得するために現在のサブプロックに対するオプティカルフロー処理を実行すること、ならびに、現在のサンプルのデルタ予測値および現在のサブプロックの現在のサンプルの(元のまたは洗練化されることになる)予測サンプル値に基づいて現在のサンプルの洗練化された予測サンプル値を取得することのために構成される。 20

【 0 0 4 0 】

第4の態様自体による装置の可能な実装形式において、複数のオプティカルフロー決定条件は、アフィンコーディングされたプロックを含むピクチャに対してPROFが有効であることを第1の指示情報が示すこと、またはアフィンコーディングされたプロックを含むピクチャに関連付けられるスライスに対してPROFが有効であることを第1の指示情報が示すこと、および、第2の指示情報が、変数fallbackModeTriggeredが0に等しく設定されることなどの、アフィンコーディングされたプロックの区分を示すことを含む。変数fallbackModeTriggeredが0に等しく設定されるとき、アフィンコーディングされたプロックの区分が必要とされ、すなわち、アフィンコーディングされたプロックの各サブプロックはそれぞれの動きベクトルを有し、それはアフィンコーディングされたプロックが非並進運動を有することを示すことが理解され得る。 30

【 0 0 4 1 】

第4の態様の任意の先行する実装形態または第4の態様自体による装置の可能な実装形式において、予測処理ユニットは、

第2の予測行列を取得することであって、第2の予測行列の要素は現在のサブプロックの予測サンプル値に基づき、いくつかの例では、第2の予測行列を取得するステップは、現在のサブプロックの動き情報に基づいて第1の予測行列を生成するステップであって、第1の予測行列の要素は現在のサブプロックの予測サンプル値に対応する、ステップ、および第1の予測行列に基づいて第2の予測行列を生成するステップ、または、現在のサブプロックの動き情報に基づいて第2の予測行列を生成するステップを含む、取得することと、第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を生成することであって、第2の予測行列のサイズは水平予測勾配行列および垂直予測勾配行列のサイズ以上である、生成することと、

水平予測勾配行列の中の現在のサンプルの水平予測勾配値、垂直予測勾配行列の中の現在

10

20

30

40

50

のサンプルの垂直予測勾配値、および現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心サンプルの動きベクトルとの差分に基づいて、現在のサブブロックの現在のサンプルのデルタ予測値($\Delta I(i,j)$)を計算することのために構成される。

【0042】

第4の態様の任意の先行する実装形態または第4の態様自体による装置の可能な実装形式において、予測処理ユニットは、アフィンコーディングされたブロックの現在のサブブロックの(元の)予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するサブブロックベースのアフィン動き補償を行うために構成される。

【0043】

第4の態様の任意の先行する実装形態または第4の態様自体による装置の可能な実装形式において、現在のサンプルが属する現在のサンプルユニット(たとえば、 2×2 サンプルブロック)の動きベクトルとサブブロックの中心サンプルの動きベクトルとの動きベクトル差分が、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心サンプルの動きベクトルとの差分として使用される。

【0044】

第4の態様の任意の先行する実装形態または第4の態様自体による装置の可能な実装形式において、第2の予測行列の要素は $I_1(p,q)$ によって表され、 p の値の範囲は $[-1, sbW]$ であり、 q の値の範囲は $[-1, sbH]$ であり、

水平予測勾配行列の要素は、 $X(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

垂直予測勾配行列の要素は、 $Y(i,j)$ によって表され、アフィンコーディングされたブロックの中の現在のサブブロックのサンプル (i,j) に対応し、 i の値の範囲は $[0, sbW-1]$ であり、 j の値の範囲は $[0, sbH-1]$ であり、

sbW はアフィンコーディングされたブロックの中の現在のサブブロックの幅を表し、 sbH はアフィンコーディングされたブロックの中の現在のサブブロックの高さを表す。

【0045】

本発明の第1の態様による方法は、本発明の第3の態様による装置によって実行され得る。本発明の第3の態様による装置のさらなる特徴および実装形式は、本発明の第1の態様による方法の特徴および実装形式に対応する。

【0046】

本発明の第2の態様による方法は、本発明の第4の態様による装置によって実行され得る。本発明の第4の態様による装置のさらなる特徴および実装形式は、本発明の第2の態様による方法の特徴および実装形式に対応する。

【0047】

第5の態様によれば、本発明は、第1の態様もしくは第2の態様自体またはそれらの実装形式による方法を行うための処理回路を備える、エンコーダ(20)に関する。

【0048】

第6の態様によれば、本発明は、第1の態様もしくは第2の態様自体またはそれらの実装形式による方法を行うための処理回路を備える、デコーダ(30)に関する。

【0049】

第7の態様によれば、本発明はデコーダに関する。デコーダは、
1つまたは複数のプロセッサと、

プロセッサに結合され、プロセッサによる実行のためのプログラミングを記憶する、非一時的コンピュータ可読記憶媒体とを備え、プログラミングは、プロセッサによって実行されると、第1の態様自体またはその実装形式による方法を行うようにデコーダを構成する。

【0050】

第8の態様によれば、本発明はエンコーダに関する。エンコーダは、

10

20

30

40

50

1つまたは複数のプロセッサと、

プロセッサに結合され、プロセッサによる実行のためのプログラミングを記憶する、非一時的コンピュータ可読記憶媒体とを備え、プログラミングは、プロセッサによって実行されると、第1の態様自体またはその実装形式による方法を行うようにエンコーダを構成する。

【0051】

第9の態様によれば、本発明は、ビデオストリームを符号化するための装置がプロセッサおよびメモリを含むに関する。メモリは、プロセッサに第2の態様による方法を実行させる命令を記憶している。

【0052】

第10の態様によれば、本発明は、プロセッサおよびメモリを含む、ビデオストリームを復号するための装置に関する。メモリは、プロセッサに第1の態様による方法を実行させる命令を記憶している。

10

【0053】

第11の態様によれば、本発明は、コンピュータ上で実行されると、第1の態様もしくは第2の態様または第1の態様もしくは第2の態様の任意の可能な実施形態による方法を実行するためのプログラムコードを含む、コンピュータプログラムに関する。

【0054】

第12の態様によれば、本発明は、実行されると、1つまたは複数のプロセッサにビデオデータをコーディングさせる命令を記憶したコンピュータ可読記憶媒体に関する。命令は、1つまたは複数のプロセッサに、第1の態様もしくは第2の態様または第1の態様もしくは第2の態様の任意の可能な実施形態による方法を実行させる。

20

【0055】

さらなる態様によれば、ビデオピクチャ符号化方法が提供され、これは、指示情報を決定するステップであって、指示情報は、符号化されることになるピクチャブロックが標的インター予測方法に従って符号化されることになるかどうかを示すために使用され、標的インター予測方法は、第1の態様もしくは第2の態様または第1の態様もしくは第2の態様の任意の可能な実施形態によるインター予測方法を含む、ステップと、指示情報をビットストリームへと符号化するステップとを含む。

【0056】

30

さらなる態様によれば、ビデオピクチャ復号方法が提供され、これは、指示情報を取得するためにビットストリームを解析するステップであって、指示情報は、復号されることになるピクチャブロックが標的インター予測方法に従って処理されることになるかどうかを示すために使用され、標的インター予測方法は、第1の態様もしくは第2の態様または第1の態様もしくは第2の態様の任意の可能な実施形態によるインター予測方法を含む、ステップと、標的インター予測方法に従って処理が実行されることを指示情報が示すとき、標的インター予測方法に従って復号されることになるピクチャブロックを処理するステップとを含む。

【0057】

1つまたは複数の実施形態の詳細は、添付の図面および以下の説明に記載される。他の特徴、目的、および利点は、説明、図面、および特許請求の範囲から明らかになるであろう。

40

【0058】

本発明の以下の実施形態では、添付の図面および描画を参照してより詳しく説明される。

【図面の簡単な説明】

【0059】

【図1A】本発明の実施形態を実装するように構成されるビデオコーディングシステムの例を示すブロック図である。

【図1B】本発明の実施形態を実装するように構成されるビデオコーディングシステムの別の例を示すブロック図である。

50

【図2】本発明の実施形態を実装するように構成されるビデオエンコーダの例を示すブロック図である。

【図3】本発明の実施形態を実装するように構成されるビデオデコーダの例示的な構造を示すブロック図である。

【図4】符号化装置または復号装置の例を示すブロック図である。

【図5】符号化装置または復号装置の別の例を示すブロック図である。

【図6】現在のブロックの空間的および時間的な動き情報候補を示す図である。

【図7】現在のアフィンコーディングされたブロックおよびA1が位置する隣接するアフィンコーディングされたブロックを示す図である。

【図8 A】構築された制御点動きベクトル予測方法を説明するための例を示す図である。 10

【図8 B】構築された制御点動きベクトル予測方法を説明するための例を示す図である。

【図9 A】本出願のある実施形態による復号方法のプロセスを示すフローチャートである。

【図9 B】構築された制御点動きベクトル予測方法を示す図である。

【図9 C】現在のアフィンコーディングされたブロックのサンプルまたはピクセルを示し、左上制御点および右上制御点の動きベクトルを示す図である。

【図9 D】水平予測勾配行列および垂直予測勾配行列および 4×4 サブブロックを計算または生成するための 6×6 予測信号ウィンドウを示す図である。

【図9 E】水平予測勾配行列および垂直予測勾配行列および 16×16 ブロックを計算または生成するための 18×18 予測信号ウィンドウを示す図である。 20

【図10】 $v(i,j)$ により表記されるサンプル位置 (i,j) のために計算されるサンプルMVと、サンプル (i,j) が属するサブブロックのサブブロックMV(V_{SB})との差分 $v(i,j)$ (赤の矢印)を示す図である。

【図11 A】本開示のある実施形態による、アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PRF)のための方法を示す図である。

【図11 B】本開示の別の実施形態による、アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PRF)のための別の方法を示す図である。

【図12】本開示のある実施形態によるPRFプロセスを示す図である。

【図13】本開示のある実施形態による、 $(M+2) \times (N+2)$ 予測ブロックの周辺領域および内側領域を示す図である。 30

【図14】本開示の別の実施形態による、 $(M+2) \times (N+2)$ 予測ブロックの周辺領域および内側領域を示す図である。

【図15】本開示のいくつかの実施形態による、ビデオ信号のアフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PRF)のための装置の例示的な構造を示すブロック図である。

【図16】コンテンツ配信サービスを実現するコンテンツ供給システムの例示的な構造を示すブロック図である。

【図17】端末デバイスの例の構造を示すブロック図である。

【発明を実施するための形態】

【0060】

以下では、同一の参照符号は、別段明示的に指定されない限り、同一のまたは少なくとも機能的に等価な特徴を指す。 40

【0061】

以下の説明では、本開示の一部をなす、および例示として、本発明の実施形態の特定の態様または本発明の実施形態が使用され得る特定の態様を示す、添付の図面への参照が行われる。本発明の実施形態は、他の態様において使用されてもよく、図面に示されない構造的または論理的な変化を備えてもよいことが理解される。したがって、以下の詳細な説明は、限定する意味で捉えられるべきではなく、本発明の範囲は、添付の特許請求の範囲によって定義される。

【0062】

たとえば、説明される方法に関連する開示は、方法を実行するように構成される対応す

10

20

30

40

50

るデバイスまたはシステムに対しても当てはまることがあり、その逆も然りであることが理解される。たとえば、1つまたは複数の特定の方法ステップが説明される場合、対応するデバイスは、説明された1つまたは複数の方法ステップを実行するために、1つまたは複数のユニット、たとえば機能ユニット(たとえば、1つまたは複数のステップを実行する1つのユニット、または複数のステップのうちの1つまたは複数を各々実行する複数のユニット)を、そのような1つまたは複数のユニットが図面において明示的に説明または図示されていなくても含むことがある。一方、たとえば、特定の装置が1つまたは複数のユニット、たとえば機能ユニットに基づいて説明される場合、対応する方法は、1つまたは複数のユニットの機能を実行するための1つのステップ(たとえば、1つまたは複数のユニットの機能を実行する1つのステップ、または複数のユニットのうちの1つまたは複数の機能を各々実行する複数のステップ)を、そのような1つまたは複数のステップが図面において明示的に説明または図示されていなくても含むことがある。さらに、本明細書において説明される様々な例示的な実施形態および/または態様の特徴は、別段特に述べられない限り、互いに組み合わせられてもよいことが理解される。

【0063】

ビデオコーディングは通常、ビデオまたはビデオシーケンスを形成するピクチャのシーケンスの処理を指す。「ピクチャ」という用語の代わりに、ビデオコーディングの分野では「フレーム」または「画像」という用語が同義語として使用され得る。ビデオコーディング(または一般にコーディング)は、ビデオ符号化およびビデオ復号という2つの部分を含む。ビデオ符号化はソース側において実行され、通常、ビデオピクチャを表現するために必要とされるデータの量を減らすために(より効率的な記憶および/または送信のために)元のビデオピクチャを処理する(たとえば、圧縮によって)ことを含む。ビデオ復号は、宛先側において実行され、通常、ビデオピクチャを再構築するためにエンコーダと比較して逆の処理を含む。ビデオピクチャ(または一般にピクチャ)の「コーディング」に言及する実施形態は、ビデオピクチャまたはそれぞれのビデオシーケンスの「符号化」もしくは「復号」に関係するものと理解されるべきである。符号化部分と復号部分の組合せは、コーデック(CODEC)(コーディングおよび復号(Coding and Decoding))とも呼ばれる。

【0064】

無損失ビデオコーディングの場合、元のビデオピクチャを再構築することができ、すなわち、再構築されるビデオピクチャは、元のビデオピクチャと同じ品質を有する(記憶または送信の間に伝送損失もしくは他のデータ損失がないと仮定して)。有損失ビデオコーディングの場合、たとえば量子化により、さらなる圧縮が、ビデオピクチャを表現するデータの量を減らすために実行され、これは、デコーダにおいて完全には再構築することができず、すなわち、再構築されたビデオピクチャの品質は、元のビデオピクチャの品質と比較してより低く、または悪い。

【0065】

いくつかのビデオコーディング規格が、「有損失ハイブリッドビデオコーデック」のグループに属する(すなわち、サンプル領域における空間予測および時間予測と、変換領域において量子化を適用するための2D変換コーディングとを組み合わせる)。ビデオシーケンスの各ピクチャは通常、重複しないブロックのセットへと区分され、コーディングは通常、ブロックレベルで実行される。言い換えると、エンコーダにおいて、ビデオは通常、ブロック(ビデオブロック)レベルで、たとえば、予測ブロックを生成するために空間(ピクチャ内)予測および/または時間(ピクチャ間)予測を使用し、残差ブロックを取得するために現在のブロック(現在処理されている/処理されることになるブロック)から予測ブロックを差し引き、送信されることになるデータの量を減らす(圧縮)ために残差ブロックを変換して残差ブロックを変換領域において量子化することによって、処理され、すなわち符号化されるが、デコーダでは、エンコーダと比較して逆の処理が、表現のために現在のブロックを再構築するために、符号化または圧縮されたブロックに適用される。さらに、エンコーダは、エンコーダとデコーダの両方が、後続のブロックの処理、すなわちコーディングのために同一の予測(たとえば、イントラ予測およびインター予測)および/または再構築を

10

20

30

40

50

生成するように、デコーダの処理ループを複製する。

【0066】

以下では、ビデオコーディングシステム10、ビデオエンコーダ20、およびビデオデコーダ30の実施形態が、図1から図3に基づいて説明される。

【0067】

図1Aは、例示的なコーディングシステム10、たとえば本出願の技法を利用し得るビデオコーディングシステム10(または略してコーディングシステム10)を示す概略ブロック図である。ビデオコーディングシステム10のビデオエンコーダ20(または略してエンコーダ20)およびビデオデコーダ30(または略してデコーダ30)は、本出願において説明される様々な例による技法を実行するように構成され得るデバイスの例を代表する。

10

【0068】

図1Aに示されるように、コーディングシステム10は、たとえば、符号化されたピクチャデータ21を復号するために宛先デバイス14へ、符号化されたピクチャデータ21を提供するように構成されるソースデバイス12を備える。

【0069】

ソースデバイス12は、エンコーダ20を備え、追加で、すなわち任意選択で、ピクチャソース16、プリプロセッサ(または前処理ユニット)18、たとえばピクチャプリプロセッサ18、および通信インターフェースまたは通信ユニット22を備え得る。

【0070】

ピクチャソース16は、任意の種類のピクチャキャプチャデバイス、たとえば、現実世界のピクチャをキャプチャするためのカメラ、および/または任意の種類のピクチャ生成デバイス、たとえば、コンピュータアニメーションピクチャを生成するためのコンピュータグラフィックスプロセッサ、または、現実世界のピクチャ、コンピュータで生成されるピクチャ(たとえば、スクリーンコンテンツ、仮想現実(VR)ピクチャ)、および/もしくはこれらの任意の組合せ(たとえば、拡張現実(AR)ピクチャ)を取得および/もしくは提供するための任意の種類の他のデバイスを備えてもよく、またはそれであってもよい。ピクチャソースは、前述のピクチャのいずれかを記憶する、任意の種類のメモリまたはストレージであつてもよい。

20

【0071】

プリプロセッサ18および前処理ユニット18によって実行される処理と区別して、ピクチャまたはピクチャデータ17は、生のピクチャまたは生のピクチャデータ17とも呼ばれ得る。

30

【0072】

プリプロセッサ18は、(生の)ピクチャデータ17を受信し、ピクチャデータ17に対して前処理を実行して前処理されたピクチャ19または前処理されたピクチャデータ19を取得するように構成される。プリプロセッサ18によって実行される前処理は、たとえば、トリミング、カラーフォーマット変換(たとえば、RGBからYCbCrへの)、色補正、またはノイズ除去を備え得る。前処理ユニット18は任意選択の構成要素であってもよいことが理解され得る。

40

【0073】

ビデオエンコーダ20は、前処理されたピクチャデータ19を受信し、符号化されたピクチャデータ21を提供するように構成される(さらなる詳細が、たとえば図2に基づいて以下で説明される)。

【0074】

ソースデバイス12の通信インターフェース22は、符号化されたピクチャデータ21を受信し、記憶または直接の再構築のために、符号化されたピクチャデータ21(またはその任意のさらなる処理されたバージョン)を、通信チャネル13を介して別のデバイス、たとえば宛先デバイス14または任意の他のデバイスに送信するように構成され得る。

【0075】

宛先デバイス14は、デコーダ30(たとえば、ビデオデコーダ30)を備え、追加で、すな

50

わち任意選択で、通信インターフェースまたは通信ユニット28、ポストプロセッサ32(または後処理ユニット32)、および表示デバイス34を備え得る。

【 0 0 7 6 】

宛先デバイス14の通信インターフェース28は、符号化されたピクチャデータ21(またはその任意のさらに処理されたバージョン)を、たとえばソースデバイス12または任意の他のソース、たとえばストレージデバイス、たとえば符号化ピクチャデータストレージデバイスから直接受信し、符号化されたピクチャデータ21をデコーダ30に提供するように構成される。

【 0 0 7 7 】

通信インターフェース22および通信インターフェース28は、ソースデバイス12と宛先デバイス14との間の直接通信リンク、たとえば直接の有線接続もしくはワイヤレス接続を介して、または任意の種類のネットワーク、たとえば有線ネットワークもしくはワイヤレスネットワークもしくはこれらの任意の組合せ、または任意の種類のプライベートネットワークおよびパブリックネットワーク、またはこれらの任意の種類の組合せを介して、符号化されたピクチャデータ21または符号化されたデータ13を、送信または受信するように構成され得る。 10

【 0 0 7 8 】

通信インターフェース22は、たとえば、符号化されたピクチャデータ21を適切なフォーマット、たとえばパケットへとパッケージングし、および/または、通信リンクもしくは通信ネットワークを介した送信のために、任意の種類の送信符号化もしくは処理を使用して符号化されたピクチャデータを処理するように構成され得る。 20

【 0 0 7 9 】

通信インターフェース22のカウンターパートをなす通信インターフェース28は、たとえば、送信されたデータを受信し、任意の種類の対応する送信の復号もしくは処理および/またはデパッケージングを使用して送信データを処理して、符号化されたピクチャデータ21を取得するように構成され得る。

【 0 0 8 0 】

通信インターフェース22と通信インターフェース28の両方が、ソースデバイス12から宛先デバイス14を指示する図1Aの通信チャネル13に対する矢印により示されるような単方向通信インターフェースとして、または双方向通信インターフェースとして構成されてもよく、たとえばメッセージを送信して受信するように、たとえば接続をセットアップして、肯定応答し、通信リンクおよび/またはデータ送信、たとえば符号化されたピクチャデータの送信に関する任意の他の情報を交換するように構成されてもよい。 30

【 0 0 8 1 】

デコーダ30は、符号化されたピクチャデータ21を受信し、復号されたピクチャデータ31または復号されたピクチャ31を提供するように構成される(さらなる詳細が、たとえば図3または図5に基づいて以下で説明される)。

【 0 0 8 2 】

宛先デバイス14のポストプロセッサ32は、後処理されたピクチャデータ33、たとえば後処理されたピクチャ33を取得するために、復号されたピクチャデータ31(再構築されたピクチャデータとも呼ばれる)、たとえば復号されたピクチャ31を後処理するように構成される。後処理ユニット32によって実行される後処理は、たとえば、カラーフォーマット変換(たとえば、YCbCrからRGBへの)、色補正、トリミング、もしくは再サンプリング、または、たとえば表示デバイス34による表示のために、たとえば復号されたピクチャデータ31を準備するための、任意の他の処理を備え得る。 40

【 0 0 8 3 】

宛先デバイス14の表示デバイス34は、たとえばユーザまたは視聴者にピクチャを表示するための、後処理されたピクチャデータ33を受信するように構成される。表示デバイス34は、再構築されたピクチャを表現するための任意の種類のディスプレイ、たとえば統合されたまたは外部のディスプレイもしくはモニタであってもよく、またはそれらを備えて

10

20

30

40

50

もよい。ディスプレイは、たとえば、液晶ディスプレイ(LCD)、有機発光ダイオード(OLE D)ディスプレイ、プラズマディスプレイ、プロジェクタ、マイクロLEDディスプレイ、liquid crystal on silicon(LCoS)、デジタル光プロセッサ(DLP)、または任意の種類の他のディスプレイを備え得る。

【0084】

図1Aは、別々のデバイスとしてソースデバイス12および宛先デバイス14を示すが、デバイスの実施形態はまた、これらの両方または両方の機能、すなわちソースデバイス12または対応する機能および宛先デバイス14または対応する機能を備えてもよい。そのような実施形態では、ソースデバイス12または対応する機能および宛先デバイス14または対応する機能は、同じハードウェアおよび/もしくはソフトウェアを使用して、または別個のハードウェアおよび/もしくはソフトウェアによって、またはこれらの任意の組合せで実装され得る。10

【0085】

説明に基づいて当業者に明らかとなるように、図1Aに示されるような、様々なユニットの機能またはソースデバイス12および/もしくは宛先デバイス14内の機能の存在と(厳密な)分割は、実際のデバイスおよび適用例に応じて変動し得る。

【0086】

エンコーダ20(たとえば、ビデオエンコーダ20)もしくはデコーダ30(たとえば、ビデオデコーダ30)、またはエンコーダ20とデコーダ30の両方が、1つまたは複数のマイクロプロセッサ、デジタルシグナルプロセッサ(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回路、ハードウェア、ビデオコーディング専用、またはこれらの任意の組合せなどの、図1Bに示されるような処理回路を介して実装され得る。エンコーダ20は、図2のエンコーダ20および/または本明細書において説明される任意の他のエンコーダシステムもしくはサブシステムに関して論じられるような、様々なモジュールを具現化するための、処理回路46を介して実装され得る。デコーダ30は、図3のデコーダ30および/または本明細書において説明される任意の他のデコーダシステムもしくはサブシステムに関して論じられるような、様々なモジュールを具現化するための、処理回路46を介して実装され得る。処理回路は、後で論じられるような様々な動作を実行するように構成され得る。図5に示されるように、技法がソフトウェアで部分的に実装される場合、デバイスは、適切な非一時的コンピュータ可読記憶媒体にソフトウェアのための命令を記憶してもよく、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てもよい。ビデオエンコーダ20とビデオデコーダ30のいずれかが、たとえば図1Bに示されるような単一のデバイスにおいて、合成されたエンコーダ/デコーダ(コーデック)の一部として統合され得る。20

【0087】

ソースデバイス12および宛先デバイス14は、任意の種類のハンドヘルドデバイスまたは固定式デバイス、たとえば、ノートブックもしくはラップトップコンピュータ、携帯電話、スマートフォン、タブレットもしくはタブレットコンピュータ、カメラ、デスクトップコンピュータ、セットトップボックス、テレビジョン、表示デバイス、デジタルメディアプレーヤ、ビデオゲームコンソール、ビデオストリーミングデバイス(コンテンツサーバスサーバまたはコンテンツ配信サーバなど)、放送受信器デバイス、放送送信器デバイスなどを含む、広範囲のデバイスのいずれを備えてもよく、オペレーティングシステムを使用しなくてもよく、または任意の種類のオペレーティングシステムを使用してもよい。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。30

【0088】

いくつかの場合、図1Aに示されるビデオコーディングシステム10は単なる例であり、本出願の技法は、符号化デバイスと復号デバイスとの間にどのようなデータ通信も必ずしも含まない、ビデオコーディング設定(たとえば、ビデオ符号化またはビデオ復号)に適用40

され得る。他の例では、データは、ローカルメモリから取り出されること、ネットワークを介してストリーミングされることなどが行われる。ビデオ符号化デバイスは、データを符号化してメモリに記憶してもよく、および/または、ビデオ復号デバイスは、メモリからデータを取り出して復号してもよい。いくつかの例では、符号化および復号は、互いに通信せず、単にデータをメモリへと符号化し、および/またはメモリからデータを取り出して復号する、デバイスによって実行される。

【 0 0 8 9 】

説明の便宜上、本発明の実施形態は、たとえば、High-Efficiency Video Coding(HEVC)を参照して、または、ITU-T Video Coding Experts Group(VCEG)およびISO/IEC Motion Picture Experts Group(MPEG)のJoint Collaboration Team on Video Coding(JCT-VC)によって開発される次世代ビデオコーディング規格である、Versatile Video coding(VVC)の参照ソフトウェアを参照して本明細書において説明される。当業者は、本発明の実施形態がHEVCまたはVVCに限定されないことを理解するであろう。10

【 0 0 9 0 】

エンコーダおよび符号化方法

図2は、本出願の技法を実装するように構成される例示的なビデオエンコーダ20の概略プロック図を示す。図2の例では、ビデオエンコーダ20は、入力201(または入力インターフェース201)、残差計算ユニット204、変換処理ユニット206、量子化ユニット208、逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、ループフィルタユニット220、復号ピクチャバッファ(DBP)230、モード選択ユニット260、エントロピー符号化ユニット270、および出力272(または出力インターフェース272)を備える。20 モード選択ユニット260は、インター予測ユニット244、イントラ予測ユニット254、および区分ユニット262を含み得る。インター予測ユニット244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。図2に示されるようなビデオエンコーダ20は、ハイブリッドビデオエンコーダまたはハイブリッドビデオコーデックに従ったビデオエンコーダとも呼ばれ得る。

【 0 0 9 1 】

残差計算ユニット204、変換処理ユニット206、量子化ユニット208、モード選択ユニット260は、エンコーダ20の順方向信号経路を形成するものとして言及されることがある、一方、逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、バッファ216、ループフィルタ220、復号ピクチャバッファ(DBP)230、インター予測ユニット244、およびイントラ予測ユニット254は、ビデオエンコーダ20の逆方向信号経路を形成するものとして言及されることがある、ビデオエンコーダ20の逆方向信号経路は、デコーダの信号経路に対応する(図3のビデオデコーダ30参照)。逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、ループフィルタ220、復号ピクチャバッファ(DBP)230、インター予測ユニット244、およびイントラ予測ユニット254は、ビデオエンコーダ20の「内蔵デコーダ」を形成するものとしても言及される。30

【 0 0 9 2 】

ピクチャおよびピクチャ区分化(ピクチャおよびブロック)

エンコーダ20は、たとえば入力201を介して、ピクチャ17(またはピクチャデータ17)、たとえば、ビデオまたはビデオシーケンスを形成するピクチャのシーケンスのうちのあるピクチャを受信するように構成され得る。受信されたピクチャまたはピクチャデータは、前処理されたピクチャ19(または前処理されたピクチャデータ19)でもあり得る。簡潔にするために、以下の説明はピクチャ17に言及する。ピクチャ17は、現在のピクチャまたはコーディングされることになるピクチャとも呼ばれることがある(具体的には、ビデオコーディングにおいて、現在のピクチャを他のピクチャ、たとえば、同じビデオシーケンス、すなわち現在のピクチャも含むビデオシーケンスの以前に符号化および/または復号されたピクチャと区別するために)。40

【 0 0 9 3 】

(デジタル)ピクチャは、強度値を伴うサンプルの2次元アレイもしくは行列であり、また

10

20

30

40

50

はそのように見なされ得る。アレイの中のサンプルは、ピクセル(pixel)(ピクチャエレメント(picture element)の短縮形)またはペルとも呼ばれ得る。アレイまたはピクチャの水平方向および垂直方向(または軸)におけるサンプルの数は、ピクチャのサイズおよび/または解像度を決める。色の表現のために、通常は3つの色成分が利用され、すなわち、ピクチャは、3つのサンプルアレイで表され、またはそれらを含むことがある。RGBフォーマットまたは色空間では、ピクチャは、対応する赤、緑、および青のサンプルアレイを含む。しかしながら、ビデオコーディングでは、各ピクセルは通常、ルミナンスおよびクロミナンスのフォーマットまたは色空間、たとえばYCbCrで表され、これは、Yにより示されるルミナンス成分(代わりにIが使用されることもある)およびCbとCrにより示される2つのクロミナンス成分を含む。ルミナンス(または略してルマ)成分Yは明るさまたはグレーレベル強度(たとえば、グレースケールピクチャにおけるような)を表し、一方、2つのクロミナンス(または略してクロマ)成分CbおよびCrは、色度または色情報成分を表す。したがって、YCbCrフォーマットのピクチャは、ルミナンスサンプル値(Y)のルミナンスサンプルアレイ、およびクロミナンス値(CbおよびCr)の2つのクロミナンスサンプルアレイを含む。RGBフォーマットのピクチャは、YCbCrフォーマットへと変換または転換されてもよく、その逆が行われてもよく、このプロセスは、色転換または色変換としても知られている。ピクチャがモノクロームである場合、ピクチャはルミナンスサンプルアレイのみを備え得る。したがって、ピクチャは、たとえば、モノクロームフォーマットのルマサンプルのアレイ、または4:2:0、4:2:2、および4:4:4カラーフォーマットのルマサンプルのアレイおよびクロマサンプルの2つの対応するアレイであり得る。

10

20

【0094】

ビデオエンコーダ20の実施形態は、ピクチャ17を複数の(通常は重複しない)ピクチャブロック203へと区分するように構成されるピクチャ区分ユニット(図2に示されない)を備え得る。これらのブロックは、ルートブロック、マクロブロック(H.264/AVC)、またはコーディングツリーブロック(CTB)もしくはコーディングツリーユニット(CTU)(H.265/HEVCおよびVVC)とも呼ばれ得る。ピクチャ区分ユニットは、ビデオシーケンスのすべてのピクチャに対する同じブロックサイズと、そのブロックサイズを定義する対応するグリッドとを使用するように、または、ピクチャ間またはピクチャのサブセットもしくはピクチャのグループ間でブロックサイズを変更し、各ピクチャを対応するブロックへと区分するよう構成されてもよい。

30

【0095】

さらなる実施形態では、ビデオエンコーダは、ピクチャ17のブロック203、たとえばピクチャ17を形成する1つの、いくつかの、またはすべてのブロックを直接受信するように構成され得る。ピクチャブロック203はまた、現在のピクチャブロックまたはコーディングされるべきピクチャブロックとも呼ばれ得る。

【0096】

ピクチャ17のように、ピクチャブロック203はやはり、強度値(サンプル値)を伴うサンプルの、しかしピクチャ17より小さい寸法の2次元アレイもしくは行列であり、またはそのように見なされ得る。言い換えると、ブロック203は、たとえば、1つのサンプルアレイ(たとえば、モノクロームピクチャ17の場合はルマアレイ、またはカラーピクチャの場合はルマアレイもしくはクロマアレイ)または3つのサンプルアレイ(たとえば、カラーピクチャ17の場合はルマアレイおよび2つのクロマアレイ)または適用されるカラーフォーマットに応じて任意の他の数および/もしくは種類のアレイを備え得る。ブロック203の水平方向および垂直方向(または軸)におけるサンプルの数は、ブロック203のサイズを決める。したがって、ブロックは、たとえば、サンプルのM×N(M列対N行)アレイ、または変換係数のM×Nアレイであり得る。

40

【0097】

図2に示されるようなビデオエンコーダ20の実施形態は、ブロックごとにピクチャ17を符号化するように構成されてもよく、たとえば、符号化および予測はブロック203ごとに実行される。

50

【 0 0 9 8 】

図2に示されるようなビデオエンコーダ20の実施形態はさらに、スライス(ビデオスライスとも呼ばれる)を使用することによってピクチャを区分および/または符号化するように構成されてもよく、ピクチャは1つまたは複数のスライス(通常は重複しない)へと区分され、またはそれらを使用して符号化されてもよく、各スライスは1つまたは複数のブロック(たとえば、CTU)を備えてもよい。

【 0 0 9 9 】

図2に示されるようなビデオエンコーダ20の実施形態はさらに、タイルグループ(ビデオタイルグループとも呼ばれる)および/またはタイル(ビデオタイルとも呼ばれる)を使用することによってピクチャを区分および/または符号化するように構成されてもよく、ピクチャは、1つまたは複数のタイルグループ(通常は重複しない)へと区分され、またはそれらを使用して符号化されてもよく、各タイルグループは、たとえば1つまたは複数のブロック(たとえば、CTU)または1つまたは複数のタイルを備えてもよく、各タイルは、たとえば、長方形の形状であってもよく、1つまたは複数のブロック(たとえば、CTU)、たとえば、完全なブロックまたは部分的なブロックを備えてもよい。

10

【 0 1 0 0 】**残差計算**

残差計算ユニット204は、たとえば、サンプル領域において残差ブロック205を取得するために、ピクチャブロック203のサンプル値から予測ブロック265(予測ブロック265のさらなる詳細は後で与えられる)のサンプル値をサンプルごとに(ピクセルごとに)差し引くことによって、ピクチャブロック203および予測ブロック265に基づいて残差ブロック205(残差205とも呼ばれる)を計算するように構成され得る。

20

【 0 1 0 1 】**変換**

変換処理ユニット206は、変換領域において変換係数207を取得するために、残差ブロック205のサンプル値に対して変換、たとえば離散コサイン変換(DCT)または離散サイン変換(DST)を適用するように構成され得る。変換係数207はまた、変換残差係数とも呼ばれることがあり、変換領域において残差ブロック205を表すことがある。

【 0 1 0 2 】

変換処理ユニット206は、H.265/AVCについて規定される変換などの、DCT/DSTの整数近似を適用するように構成され得る。直交DCT変換と比較して、そのような整数近似は通常、ある係数によりスケーリングされる。順変換および逆変換によって処理される残差ブロックのノルムを保持するために、変換プロセスの一部として追加のスケーリング係数が適用される。スケーリング係数は通常、スケーリング係数がシフト演算のために2のべき乗であること、変換係数のビット深度、正確さと実装コストとのトレードオフなどのような、いくつかの制約に基づいて選ばれる。たとえば、具体的なスケーリング係数は、たとえば逆変換処理ユニット212による逆変換(および、たとえばビデオデコーダ30における逆変換処理ユニット312による、対応する逆変換)について規定され、エンコーダ20における、たとえば変換処理ユニット206による、順変換のための対応するスケーリング係数が、それに従って規定され得る。

30

【 0 1 0 3 】

ビデオエンコーダ20の実施形態(それぞれ変換処理ユニット206)は、変換パラメータ、たとえば1つまたは複数の変換のタイプを、たとえば、直接、またはエントロピー符号化ユニット270を介して符号化もしくは圧縮された状態で出力するように構成され得るので、たとえば、ビデオデコーダ30は、復号のためにその変換パラメータを受信して使用し得る。

40

【 0 1 0 4 】**量子化**

量子化ユニット208は、たとえばスカラー量子化またはベクトル量子化を適用することによって、量子化された係数209を取得するために変換係数207を量子化するように構成

50

され得る。量子化された係数209は、量子化された変換係数209または量子化された残差係数209とも呼ばれ得る。

【0105】

量子化プロセスは、変換係数207の一部またはすべてに関連付けられるビット深度を減らし得る。たとえば、nビットの変換係数は、量子化の間にmビットの変換係数へと丸められてもよく、nはmより大きい。量子化の程度は、量子化パラメータ(QP)を調整することによって修正されてもよい。たとえば、スカラー量子化では、より細かいまたは粗い量子化を達成するために、異なるスケーリングが適用され得る。より小さい量子化ステップサイズはより細かい量子化に対応し、一方でより大きい量子化ステップサイズはより粗い量子化に対応する。適用可能な量子化ステップサイズは、量子化パラメータ(QP)によって示され得る。量子化パラメータは、たとえば、適用可能な量子化ステップサイズのあらかじめ定められたセットに対するインデックスであり得る。たとえば、小さい量子化パラメータが細かい量子化(小さい量子化ステップサイズ)に対応してもよく、大きい量子化パラメータが粗い量子化(大きい量子化ステップサイズ)に対応してもよく、またはこの逆であってもよい。量子化は、量子化ステップサイズによる除算を含むことがあり、たとえば逆量子化ユニット210による、対応するおよび/または量子化解除は、量子化ステップサイズによる乗算を含むことがある。いくつかの規格、たとえばHEVCによる実施形態は、量子化ステップサイズを決定するために量子化パラメータを使用するように構成され得る。一般に、量子化ステップサイズは、除算を含む式の固定点近似を使用して、量子化パラメータに基づいて計算され得る。残差ブロックのノルムを復元するために、追加のスケーリング係数が量子化および量子化解除のために導入されることがあり、これは、量子化ステップサイズおよび量子化パラメータの式の固定点近似において使用されるスケーリングにより、修正されることがある。1つの例示的な実装形態では、逆変換および量子化解除のスケーリングは、合成されてもよい。代替的に、たとえばビットストリームにおいて、カスタマイズされた量子化テーブルが使用されて、エンコーダからデコーダにシグナリングされてもよい。量子化は有損失演算であり、損失は量子化ステップサイズの増大とともに増大する。

10

20

30

【0106】

ビデオエンコーダ20の実施形態(それぞれ量子化ユニット208)は、量子化パラメータ(QP)を、たとえば、直接、またはエントロピー符号化ユニット270を介して符号化された状態で出力するように構成され得るので、たとえば、ビデオデコーダ30は、復号のためにその量子化パラメータを受信して適用し得る。

【0107】

逆量子化

逆量子化ユニット210は、たとえば、量子化ユニット208と同じ量子化ステップサイズに基づいて、またはそれを使用して、量子化ユニット208によって適用される量子化方式の逆を適用することによって、量子化解除された係数211を取得するために、量子化された係数に対して量子化ユニット208の逆量子化を適用するように構成される。量子化解除された係数211はまた、量子化解除された残差係数211とも呼ばれることがあり、量子化による損失により変換係数と通常は同一ではないが、変換係数207に対応することがある。

40

【0108】

逆変換

逆変換処理ユニット212は、サンプル領域において再構築された残差ブロック213(または対応する量子化解除された係数213)を取得するために、変換処理ユニット206により適用される変換の逆変換、たとえば、逆離散コサイン変換(DCT)または逆離散サイン変換(DST)または他の逆変換を適用するように構成される。再構築された残差ブロック213は、変換ブロック213とも呼ばれ得る。

【0109】

再構築

50

再構築ユニット214(たとえば、加算器(adder)または加算部(uinter)214)は、たとえば、再構築された残差ブロック213のサンプル値と予測ブロック265のサンプル値をサンプルごとに加算することによって、サンプル領域において再構築されたブロック215を取得するために、変換ブロック213(すなわち、再構築された残差ブロック213)を予測ブロック265に加算するように構成される。

【0110】

フィルタリング

ループフィルタユニット220(または略して「ループフィルタ」220)は、フィルタリングされたブロック221を取得するために再構築されたブロック215をフィルタリングし、または一般には、フィルタリングされたサンプルを取得するために再構築されたサンプルをフィルタリングするように構成される。ループフィルタユニットは、たとえば、ピクセル遷移を滑らかにし、またはビデオ品質を他の方法で改善するように構成される。ループフィルタユニット220は、デブロッキングフィルタなどの1つまたは複数のループフィルタ、サンプル適応オフセット(SAO)フィルタ、または1つまたは複数の他のフィルタ、たとえば、バイラテラルフィルタ、適応ループフィルタ(ALF)、先鋭化フィルタ、平滑化フィルタ、もしくは協調フィルタ、またはこれらの任意の組合せを備え得る。ループフィルタユニット220は図2ではループフィルタであるものとして示されるが、他の構成では、ループフィルタユニット220は、ポストループフィルタとして実装され得る。フィルタリングされたブロック221は、フィルタリングされた再構築されたブロック221とも呼ばれ得る。

10

【0111】

ビデオエンコーダ20の実施形態(それぞれループフィルタユニット220)は、ループフィルタパラメータ(サンプル適応オフセット情報など)を、たとえば、直接、またはエントロピー符号化ユニット270を介して符号化された状態で出力するように構成され得るので、たとえば、デコーダ30は、復号のために同じループフィルタパラメータまたはそれぞれのループフィルタを受信して適用し得る。

20

【0112】

復号ピクチャバッファ

復号ピクチャバッファ(DPB)230は、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ための、参照ピクチャ、または一般には参照ピクチャデータを記憶するメモリであり得る。DPB230は、同期DRAM(SDRAM)、磁気抵抗RAM(MRAM)、抵抗性RAM(RRAM)、または他のタイプのメモリデバイスを含む、ダイナミックランダムアクセスメモリ(DRAM)などの種々のメモリデバイスのいずれかによって形成され得る。復号ピクチャバッファ(DPB)230は、1つまたは複数のフィルタリングされたブロック221を記憶するように構成され得る。復号ピクチャバッファ230はさらに、同じ現在のピクチャ、または異なるピクチャ、たとえば以前に再構築されたピクチャの、他の以前にフィルタリングされたブロック、たとえば以前に再構築されフィルタリングされたブロック221を記憶するように構成されてもよく、たとえばインター予測のために、完全な以前に再構築された、すなわち復号されたピクチャ(および対応する参照ブロックとサンプル)および/または部分的に再構築された現在のピクチャ(および対応する参照ブロックとサンプル)を提供してもよい。復号ピクチャバッファ(DPB)230はまた、たとえば再構築されたブロック215がループフィルタユニット220によってフィルタリングされていない場合、1つまたは複数のフィルタリングされていない再構築されたブロック215、もしくは一般にはフィルタリングされていない再構築されたサンプルを記憶し、または、再構築されたブロックもしくはサンプルの任意の他のさらに処理されたバージョンを記憶するように構成され得る。

30

40

【0113】

モード選択(区分化および予測)

モード選択ユニット260は、区分ユニット262、インター予測ユニット244、およびイントラ予測ユニット254を備え、元のピクチャデータ、たとえば元のブロック203(現在のピクチャ17の現在のブロック203)、および再構築されたピクチャデータ、たとえば、同

50

じ(現在の)ピクチャのフィルタリングされたおよび/もしくはフィルタリングされていない再構築されたサンプルもしくはブロックを、かつ/または、1つまたは複数の以前に復号されたピクチャから、たとえば復号ピクチャバッファ230もしくは他のバッファ(たとえば、図示されないラインバッファ)から、受信または取得するように構成される。再構築されたピクチャデータは、予測ブロック265または予測子265を取得するために、予測、たとえばインター予測またはイントラ予測のための参照ピクチャデータとして使用される。

【0114】

モード選択ユニット260は、現在のブロック予測モードに対する区分化(区分化なしを含む)および予測モード(たとえば、イントラ予測モードまたはインター予測モード)を決定または選択し、対応する予測ブロック265を生成するように構成されてもよく、この予測ブロックは、残差ブロック205の計算のために、および再構築されたブロック215の再構築のために使用される。

10

【0115】

モード選択ユニット260の実施形態は、区分化および予測モードを(たとえば、モード選択ユニット260によってサポートされるもの、またはそれが利用可能であるものから)選択するように構成されてもよく、これは、最良の一一致、または言い換えると、最小の残差(最小の残差は送信または記憶のためのより良好な圧縮を意味する)、または最小のシグナリングオーバーヘッド(最小のシグナリングオーバーヘッドは送信または記憶のためのより良好な圧縮を意味する)をもたらし、またはそれらの両方を考慮し、もしくはバランスを取る。モード選択ユニット260は、レートひずみ最適化(RDO)に基づいて区分化および予測モードを決定し、すなわち、最小のレートひずみをもたらす予測モードを選択するように構成され得る。この文脈における「最良」、「最小」、「最適」などの用語は、必ずしも全体的な「最良」、「最小」、「最適」などを指さず、値がある閾値を超え、もしくは下回ること、または、「最適ではない選択」につながる可能性があるが複雑さと処理時間を減らす他の制約のような、終了基準または選択基準の充足も指すことがある。

20

【0116】

言い換えると、区分ユニット262は、ブロック203を、たとえば四分木区分化(QT)、二分木区分化(BT)、三分木区分化(TT)またはこれらの任意の組合せを繰り返し使用して、より小さいブロック区分またはサブブロック(これはやはりブロックを形成する)へと区分し、たとえばブロック区分またはサブブロックの各々に対する予測を実行するように構成されてもよく、モード選択は区分されたブロック203の木構造の選択を備え、予測モードはブロック区分またはサブブロックの各々に適用される。

30

【0117】

以下では、例示的なビデオエンコーダ20によって実行される、(たとえば、区分ユニット260による)区分化および(インター予測ユニット244およびイントラ予測ユニット254による)予測処理が、より詳しく説明される。

【0118】

区分化

区分ユニット262は、現在のブロック203をより小さい区分、たとえば正方形または長方形のサイズのより小さいブロックへと区分(またはスプリット)し得る。これらより小さいブロック(サブブロックとも呼ばれ得る)はさらに、より小さい区分へと分割され得る。これは、木区分化または階層的木区分化とも呼ばれ、たとえばルート木レベル0(階層レベル0、深度0)におけるルートブロックが、再帰的に区分され、たとえば次に低い木レベルの2つ以上のブロック、たとえば木レベル1におけるノード(階層レベル1、深度1)へと区分されてもよく、これらのブロックは、次に低いレベル、たとえば木レベル2(階層レベル2、深度2)の2つ以上のブロックへと再び区分などされてもよく、これは、たとえば終了基準が満たされたことにより、たとえば最大の木深度または最小のブロックサイズに達したことにより、区分化が終了するまで続く。さらに区分されないブロックは、木のリーフブロックまたはリーフノードとも呼ばれる。2つの区分への区分化を用いる木は二分木(BT)と呼ばれ、3つの区分への区分化を用いる木は三分木(TT)と呼ばれ、4つの区分への区分化

40

50

を用いる木は四分木(QT)と呼ばれる。

【0119】

前に言及されたように、本明細書において使用される「ブロック」という用語は、ピクチャの一部分、特に正方形部分または長方形部分であり得る。たとえば、HEVCおよびVVCを参照すると、ブロックは、コーディングツリーユニット(CTU)、コーディングユニット(CU)、予測ユニット(PU)、および変換ユニット(TU)、ならびに/または、対応するブロック、たとえばコーディングツリーブロック(CTB)、コーディングブロック(CB)、変換ブロック(TB)、もしくは予測ブロック(PB)であってもよく、またはそれらに対応してもよい。

【0120】

たとえば、コーディングツリーユニット(CTU)は、ルマサンプルのCTB、3つのサンプルアレイを有するピクチャのクロマサンプルの2つの対応するCTB、または、モノクロームピクチャ、もしくはサンプルをコーディングするために使用される3つの別個の色平面およびシンタックス構造を使用してコーディングされるピクチャのサンプルのCTBであってもよく、またはそれらを備えてよい。それに対応して、コーディングツリーブロック(CTB)は、CTBへの成分の分割が区分化であるような、Nの何らかの値に対するサンプルのN × Nブロックであり得る。コーディングユニット(CU)は、ルマサンプルのコーディングブロック、3つのサンプルアレイを有するピクチャのクロマサンプルの2つの対応するコーディングブロック、または、モノクロームピクチャ、もしくはサンプルをコーディングするために使用される3つの別個の色平面およびシンタックス構造を使用してコーディングされるピクチャのサンプルのコーディングブロックであってもよく、またはそれらを備えてよい。それに対応して、コーディングブロック(CB)は、コーディングブロックへのCTBの分割が区分化であるような、MおよびNの何らかの値に対するサンプルのM × Nブロックであり得る。

10

【0121】

実施形態では、たとえばHEVCによれば、コーディングツリーユニット(CTU)は、コーディングツリーとして表記される四分木構造を使用することによってCUへと分割され得る。インターピクチャ(時間)予測を使用してピクチャエリアをコーディングするか、イントラピクチャ(空間)予測を使用してピクチャエリアをコーディングするかの決定は、CUレベルで行われる。各CUはさらに、PU分割タイプに従って、1つ、2つ、または4つのPUへと分割され得る。1つのPUの内部で、同じ予測プロセスが適用され、関連する情報はPUごとにデコーダに送信される。PU分割タイプに基づいて予測プロセスを適用することによって残差ブロックを取得した後、CUは、CUのためのコーディングツリーに類似した別の四分木構造に従って、変換ユニット(TU)へと区分され得る。

20

【0122】

実施形態では、たとえばVersatile Video Coding(VVC)と呼ばれる現在開発中の最新のビデオコーディング規格によれば、たとえば、コーディングブロックを区分するために、合成四分木および二分木(QTBT)区分化が使用される。QTBTブロック構造では、CUは正方形または長方形のいずれかの形状を有し得る。たとえば、コーディングツリーユニット(CTU)はまず、四分木構造によって区分される。四分木リーフノードはさらに、二分木または三分(または三重)木構造によって区分される。区分化木リーフノードはコーディングユニット(CU)と呼ばれ、そのセグメント化は、さらなる区分化なしで予測および変換処理のために使用される。これは、CU、PU、およびTUがQTBTコーディングブロック構造において同じブロックサイズを有することを意味する。並列して、複数の区分、たとえば三重木区分が、QTBTブロック構造とともに使用され得る。

30

【0123】

一例では、ビデオエンコーダ20のモード選択ユニット260は、本明細書において説明される区分化技法の任意の組合せを実行するように構成され得る。

【0124】

上で説明されたように、ビデオエンコーダ20は、(たとえば、所定の)予測モードのセットから最良のまたは最適な予測モードを、決定または選択するように構成される。予測モ

40

50

ードのセットは、たとえばイントラ予測モードおよび/またはインター予測モードを備え得る。

【0125】

イントラ予測

イントラ予測モードのセットは、35個の異なるイントラ予測モード、たとえばDC(または平均)モードおよび平面モードのような非指向性モード、もしくは、たとえばHEVCにおいて定義されるような指向性モードを備えてもよく、または、67個の異なるイントラ予測モード、たとえばDC(または平均)モードおよび平面モードのような非指向性モード、もしくは、たとえばVVCのために定義されるような指向性モードを備えてもよい。

【0126】

イントラ予測ユニット254は、イントラ予測モードのセットのうちのあるイントラ予測モードに従ってイントラ予測ブロック265を生成するために、同じ現在のピクチャの隣接ブロックの再構築されたサンプルを使用するように構成される。

【0127】

イントラ予測ユニット254(または一般にはモード選択ユニット260)はさらに、符号化されたピクチャデータ21に含めるために、シンタックス要素266の形式でイントラ予測パラメータ(または一般には、ブロックのための選択されたイントラ予測モードを示す情報)をエントロピー符号化ユニット270に出力するように構成されるので、たとえば、ビデオデコーダ30は、復号のために予測パラメータを受信して使用し得る。

【0128】

インター予測

インター予測モードのセット(またはあり得るインター予測モード)は、利用可能な参照ピクチャ(すなわち、たとえばDPB230に記憶されている、以前の少なくとも部分的に復号されたピクチャ)、および他のインター予測パラメータ、たとえば、最良の一一致する参照ブロックを探すために、参照ピクチャの全体が使用されるか、もしくは参照ピクチャの一部、たとえば現在のブロックのエリアの周りの探索ウィンドウエリアだけが使用されるか、ならびに/または、たとえばピクセル補間が適用されるかどうか、たとえば、2分の1/セミペルおよび/もしくは4分の1ペル補間が適用されるかどうかに依存する。

【0129】

上記の予測モードに加えて、スキップモードおよび/またはダイレクトモードが適用され得る。

【0130】

インター予測ユニット244は、動き推定(ME)ユニットおよび動き補償(MC)ユニット(ともに図2に示されない)を含み得る。動き推定ユニットは、ピクチャブロック203(現在のピクチャ17の現在のピクチャブロック203)および復号されたピクチャ231を、または、少なくとも1つまたは複数の以前に再構築されたブロック、たとえば、1つまたは複数の他の/異なる以前に復号されたピクチャ231の再構築されたブロックを、動き推定のために受信または取得するように構成され得る。たとえば、ビデオシーケンスは、現在のピクチャおよび以前に復号されたピクチャ231を備えてもよく、または言い換えると、現在のピクチャおよび以前に復号されたピクチャ231は、ビデオシーケンスを形成するピクチャのシーケンスの一部であってもよく、またはそれを形成してもよい。

【0131】

エンコーダ20は、たとえば、複数の他のピクチャの同じピクチャまたは異なるピクチャの複数の参照ブロックからある参照ブロックを選択し、参照ピクチャ(または参照ピクチャインデックス)、および/または、参照ブロックの位置(x、y座標)と現在のブロックの位置との間のオフセット(空間オフセット)を、インター予測パラメータとして動き推定ユニットに提供するように構成され得る。このオフセットは動きベクトル(MV)とも呼ばれる。

【0132】

動き補償ユニットは、インター予測パラメータを取得する、たとえば受信するように構成され、インター予測ブロック265を取得するために、インター予測パラメータに基づい

10

20

30

40

50

て、またはそれをしてインター予測を実行するように構成される。動き補償ユニットによって実行される動き補償は、動き推定によって決定される動き/ブロックベクトルに基づいて予測ブロックをフェッチまたは生成すること、場合によっては、サブサンプル精度への補間を実行することを伴い得る。補間フィルタリングは、既知のサンプルから追加のサンプルを生成し得るので、ピクチャブロックをコーディングするために使用され得る予測ブロック候補の数が増える可能性がある。現在のピクチャブロックのPUのための動きベクトルを受信すると、動き補償ユニットは、参照ピクチャリストのうちの1つにおいて動きベクトルが指示する予測ブロックを見つける。

【0133】

動き補償ユニットはまた、ビデオスライスのピクチャブロックを復号する際にビデオデコーダ30により使用するためのブロックおよびビデオスライスに関連付けられるシンタックス要素を生成し得る。スライスおよびそれぞれのシンタックス要素に加えて、またはその代替として、タイルグループおよび/またはタイルならびにそれぞれのシンタックス要素が、生成または使用され得る。

【0134】

エントロピーコーディング

エントロピー符号化ユニット270は、たとえば、エントロピー符号化アルゴリズムまたは方式(たとえば、可変長コーディング(VLC)方式、コンテキスト適応VLC方式(CAVLC)、算術コーディング方式、二値化、コンテキスト適応バイナリ算術コーディング(CABAC)、シンタックスベースコンテキスト適応バイナリ算術コーディング(SBAC)、確率間隔区分化エンタロピー(PIPE)コーディング、または別のエントロピー符号化方法もしくは技法)またはバイパス(圧縮なし)を、量子化された係数209、インター予測パラメータ、イントラ予測パラメータ、ループフィルタパラメータ、および/または他のシンタックス要素に適用して、たとえば符号化されたビットストリーム21の形式で出力272を介して出力され得る、符号化されたピクチャデータ21を取得するように構成されるので、たとえば、ビデオデコーダ30は、復号のためにそのパラメータを受信して使用し得る。符号化されたビットストリーム21は、ビデオデコーダ30に送信され、または、ビデオデコーダ30による後の送信もしくは取り出しのためにメモリに記憶され得る。

【0135】

ビデオエンコーダ20の他の構造的な変形が、ビデオストリームを符号化するために使用され得る。たとえば、非変換ベースのエンコーダ20は、いくつかのブロックまたはフレームに対して、変換処理ユニット206なしで直接残差信号を量子化することができる。別の実装形態では、エンコーダ20は、単一のユニットへと組み合わせられる量子化ユニット208および逆量子化ユニット210を有し得る。

【0136】

デコーダおよび復号方法

図3は、本出願の技法を実装するように構成されるビデオデコーダ30の例を示す。ビデオデコーダ30は、復号されたピクチャ331を取得するために、たとえばエンコーダ20によって符号化される、符号化されたピクチャデータ21(たとえば、符号化されたビットストリーム21)を受信するように構成される。符号化されたピクチャデータまたはビットストリームは、符号化されたピクチャデータ、たとえば、符号化されたビデオスライス(および/またはタイルグループもしくはタイル)のピクチャブロックおよび関連するシンタックス要素を表すデータを復号するための情報を含む。

【0137】

図3の例では、デコーダ30は、エントロピー復号ユニット304、逆量子化ユニット310、逆変換処理ユニット312、再構築ユニット314(たとえば、加算部314)、ループフィルタ320、復号ピクチャバッファ(DPB)330、モード適用ユニット360、インター予測ユニット344、およびイントラ予測ユニット354を備える。インター予測ユニット344は、動き補償ユニットであってもよく、またはそれを含んでもよい。ビデオデコーダ30は、いくつかの例では、図2からのビデオエンコーダ100に関して説明される符号化パスとは全般

10

20

30

40

50

に逆の復号パスを実行し得る。

【0138】

エンコーダ20に関して説明されるように、逆量子化ユニット210、逆変換処理ユニット212、再構築ユニット214、ループフィルタ220、復号ピクチャバッファ(DBP)230、インター予測ユニット344、およびイントラ予測ユニット354は、ビデオエンコーダ20の「内蔵デコーダ」を形成するものとしても言及される。したがって、逆量子化ユニット310は逆量子化ユニット110と機能的に同一であってもよく、逆変換処理ユニット312は逆変換処理ユニット212と機能的に同一であってもよく、再構築ユニット314は再構築ユニット214と機能的に同一であってもよく、ループフィルタ320はルートブロック220と機能的に同一であってもよく、復号ピクチャバッファ330は復号ピクチャバッファ230と機能的に同一であってもよい。したがって、ビデオエンコーダ20のそれぞれのユニットおよび機能について与えられる説明は、相応にビデオデコーダ30のそれぞれのユニットおよび機能に当てはまる。10

【0139】

エントロピー復号

エントロピー復号ユニット304は、ビットストリーム21(または一般には符号化されたピクチャデータ21)を解析し、たとえば、エントロピー復号を符号化されたピクチャデータ21に対して実行して、たとえば、量子化された係数309および/または復号されたコーディングパラメータ(図3に示されない)、たとえばインター予測パラメータ(たとえば、参照ピクチャインデックスおよび動きベクトル)、イントラ予測パラメータ(たとえば、イントラ予測モードまたはインデックス)、変換パラメータ、量子化パラメータ、ループフィルタパラメータ、および/または他のシンタックス要素のいずれかもしくはすべてを取得するように構成される。エントロピー復号ユニット304は、エンコーダ20のエントロピー符号化ユニット270に関して説明されたような符号化方式に対応する復号アルゴリズムまたは方式を適用するように構成され得る。エントロピー復号ユニット304はさらに、インター予測パラメータ、イントラ予測パラメータ、および/または他のシンタックス要素をモード適用ユニット360に提供し、他のパラメータをデコーダ30の他のユニットに提供するように構成され得る。ビデオデコーダ30は、ビデオスライスレベルおよび/またはビデオブロックレベルにおいてシンタックス要素を受信し得る。スライスおよびそれぞれのシンタックス要素に加えて、またはその代替として、タイルグループおよび/またはタイルならびにそれぞれのシンタックス要素が、受信および/または使用され得る。20

【0140】

逆量子化

逆量子化ユニット310は、量子化パラメータ(QP)(または一般に、逆量子化に関する情報)および量子化された係数を符号化されたピクチャデータ21から(たとえばエントロピー復号ユニット304による、たとえば解析および/または復号によって)受信し、量子化パラメータに基づいて、復号された量子化された係数309に対して逆量子化を適用して、変換係数311とも呼ばれる量子化解除された係数311を取得するように構成され得る。逆量子化プロセスは、適用されるべき量子化の程度、および同様に逆量子化の程度を決定するために、ビデオスライス(またはタイルまたはタイルグループ)の中の各ビデオブロックに対して、ビデオエンコーダ20によって決定される量子化パラメータの使用を含み得る。30

【0141】

逆変換

逆変換処理ユニット312は、変換係数311とも呼ばれる量子化解除された係数311を受信し、サンプル領域において再構築された残差ブロック213を取得するために、変換を量子化解除された係数311に適用するように構成され得る。再構築された残差ブロック213は、変換ブロック313とも呼ばれ得る。変換は、逆変換、たとえば逆DCT、逆DST、逆整数変換、または概念的に同様の逆変換プロセスであり得る。逆変換処理ユニット312はさらに、量子化解除された係数311に適用されるべき変換を決定するために、符号化されたピクチャデータ21から変換パラメータまたは対応する情報を(たとえば、エントロピー復40

号ユニット304による、たとえば解析および/または復号によって)受信するように構成され得る。

【0142】

再構築

再構築ユニット314(たとえば、加算器(adder)または加算部(uinter)314)は、たとえば、再構築された残差ブロック313のサンプル値と予測ブロック365のサンプル値を加算することによって、サンプル領域において再構築されたブロック315を取得するために、再構築された残差ブロック313を予測ブロック365に加算するように構成され得る。

【0143】

フィルタリング

ループフィルタユニット320(コーディングループの中またはコーディングループの後のいずれか)は、フィルタリングされたブロック321を取得するために、たとえば、ピクセル遷移を滑らかにするために、またはビデオ品質を別の方法で改善するために、再構築されたブロック315をフィルタリングするように構成される。ループフィルタユニット320は、デブロッキングフィルタなどの1つまたは複数のループフィルタ、サンプル適応オフセット(SAO)フィルタ、または1つまたは複数の他のフィルタ、たとえば、バイラテラルフィルタ、適応ループフィルタ(ALF)、先鋭化フィルタ、平滑化フィルタ、もしくは協調フィルタ、またはこれらの任意の組合せを備え得る。ループフィルタユニット320は図3ではループフィルタであるものとして示されるが、他の構成では、ループフィルタユニット320は、ポストループフィルタとして実装され得る。

10

【0144】

復号ピクチャバッファ

ピクチャの復号されたビデオブロック321は次いで、復号ピクチャバッファ330に記憶され、これは、他のピクチャに対する後続の動き補償および/または出力それぞれに表示のための参照ピクチャとして、復号されたピクチャ331を記憶する。

【0145】

デコーダ30は、たとえば出力312を介して、ユーザへの提示またはユーザによる視聴のために、復号されたピクチャ311を出力するように構成される。

【0146】

予測

30

インター予測ユニット344はインター予測ユニット244(具体的には動き補償ユニット)と同一であってもよく、イントラ予測ユニット354はインター予測ユニット254と機能的に同一であってもよく、区分および/もしくは予測パラメータ、または、符号化されたピクチャデータ21から受信されるそれぞれの情報に基づいて(たとえばエントロピー復号ユニット304による、たとえば解析および/または復号によって)、分割または区分の決定と予測を実行する。モード適用ユニット360は、予測ブロック365を取得するために、再構築されたピクチャ、ブロック、またはそれぞれのサンプル(フィルタリングされた、またはフィルタリングされていない)に基づいて、ブロックごとに予測(イントラ予測またはインター予測)を実行するように構成され得る。

【0147】

ビデオスライスがイントラコーディングされた(I)スライスとしてコーディングされるとき、モード適用ユニット360のイントラ予測ユニット354は、シグナリングされたイントラ予測モードおよび現在のピクチャの以前に復号されたブロックからのデータに基づいて、現在のビデオスライスのピクチャブロックのための予測ブロック365を生成するように構成される。ビデオピクチャがインターコーディングされた(すなわち、BまたはP)スライスとしてコーディングされるとき、モード適用ユニット360のインター予測ユニット344(たとえば、動き補償ユニット)は、動きベクトルおよびエントロピー復号ユニット304から受信された他のシンタックス要素に基づいて、現在のビデオスライスのビデオブロックのための予測ブロック365を生成するように構成される。インター予測のために、予測ブロックは、参照ピクチャリストのうちの1つの中の参照ピクチャのうちの1つから生成され得

40

50

る。ビデオデコーダ30は、DPB330に記憶されている参照ピクチャに基づいて、デフォルトの構築技法を使用して、リスト0およびリスト1という参照フレームリストを構築し得る。同じことまたは同様のことが、スライス(たとえば、ビデオスライス)に加えて、またはその代わりに、タイルグループ(たとえば、ビデオタイルグループ)および/またはタイル(たとえば、ビデオタイル)に対して適用されてもよく、またはそれを使用する実施形態によって適用されてもよく、たとえば、ビデオは、I、P、またはBタイルグループおよび/もしくはタイルを使用してコーディングされてもよい。

【0148】

モード適用ユニット360は、動きベクトルまたは関連する情報および他のシンタックス要素を解析することによって、現在のビデオスライスのビデオブロックのための予測情報を決定するように構成され、復号されている現在のビデオブロックのための予測ブロックを生成するために予測情報を使用する。たとえば、モード適用ユニット360は、受信されたシンタックス要素の一部を使用して、ビデオスライスのビデオブロックをコーディングするために使用される予測モード(たとえば、イントラ予測またはインター予測)、インター予測スライスタイプ(たとえば、Bスライス、Pスライス、またはGPBスライス)、スライスのための参照ピクチャリストのうちの1つまたは複数のための構築情報、スライスの各々のインター符号化されたビデオブロックのための動きベクトル、スライスの各々のインターコーディングされたビデオブロックのためのインター予測ステータス、および現在のビデオスライスの中のビデオブロックを復号するための他の情報を決定する。同じことまたは同様のことが、スライス(たとえば、ビデオスライス)に加えて、またはその代わりに、タイルグループ(たとえば、ビデオタイルグループ)および/またはタイル(たとえば、ビデオタイル)を使用する実施形態に対して適用されてもよく、またはそれを使用する実施形態によって適用されてもよく、たとえば、ビデオは、I、P、またはBタイルグループおよび/もしくはタイルを使用してコーディングされてもよい。

【0149】

図3に示されるようなビデオデコーダ30の実施形態は、スライス(ビデオスライスとも呼ばれる)を使用することによってピクチャを区分および/または復号するように構成されてもよく、ピクチャは1つまたは複数のスライス(通常は重複しない)へと区分され、またはそれらを使用して復号されてもよく、各スライスは1つまたは複数のブロック(たとえば、CTU)を含んでもよい。

【0150】

図3に示されるようなビデオデコーダ30の実施形態は、タイルグループ(ビデオタイルグループとも呼ばれる)および/またはタイル(ビデオタイルとも呼ばれる)を使用することによってピクチャを区分および/または復号するように構成されてもよく、ピクチャは、1つまたは複数のタイルグループ(通常は重複しない)へと区分され、またはそれらを使用して復号されてもよく、各タイルグループは、たとえば1つまたは複数のブロック(たとえば、CTU)または1つまたは複数のタイルを備えてもよく、各タイルは、たとえば、長方形の形状であってもよく、1つまたは複数のブロック(たとえば、CTU)、たとえば、完全なブロックまたは部分的なブロックを備えてもよい。

【0151】

ビデオデコーダ30の他の変形が、符号化されたピクチャデータ21を復号するために使用され得る。たとえば、デコーダ30は、ループフィルタリングユニット320なしで出力ビデオストリームを生成することができる。たとえば、非変換ベースのデコーダ30は、いくつかのブロックまたはフレームに対して、逆変換処理ユニット312なしで残差信号を直接逆量子化することができる。別の実装形態では、ビデオデコーダ30は、単一のユニットへと組み合わせられる逆量子化ユニット310および逆変換処理ユニット312を有し得る。

【0152】

エンコーダ20およびデコーダ30において、現在のステップの処理結果はさらに処理されて、次いで次のステップに出力され得ることを理解されたい。たとえば、補間フィルタリング、動きベクトル導出、またはループフィルタリングの後で、切り取りまたはシフト

10

20

30

40

50

などのさらなる動作が、補間フィルタリング、動きベクトル導出、またはループフィルタリングの処理結果に対して実行され得る。

【 0 1 5 3 】

さらなる動作が、現在のブロックの導出された動きベクトル(限定はされないが、アフィンモードの制御点動きベクトル、アフィンモード、平面モード、ATMVPモードのサブブロック動きベクトル、時間動きベクトルなどを含む)に適用され得る。たとえば、動きベクトルの値は、それを表すビットに従って、あらかじめ定められた範囲に制約される。動きベクトルを表すビットがbitDepthである場合、範囲は $-2^{(bitDepth-1)} \sim 2^{(bitDepth-1)}$ -1であり、ここで「 \wedge 」は指数を意味する。たとえば、bitDepthが16に等しく設定される場合、範囲は-32768 ~ 32767であり、bitDepthが18に等しく設定される場合、範囲は-131072 ~ 131071である。たとえば、導出された動きベクトルの値(たとえば、1つの 8×8 ブロック内の4つの 4×4 サブブロックのMV)は、4つの 4×4 サブブロックMVの整数部分の間の最大の差が、1サンプルより大きくななど、Nサンプルより大きくなないように制約される。10

【 0 1 5 4 】

図4は、本開示のある実施形態によるビデオコーディングデバイス400の概略図である。ビデオコーディングデバイス400は、本明細書において説明されるような開示される実施形態を実装するのに適している。ある実施形態では、ビデオコーディングデバイス400は、図1Aのビデオデコーダ30などのデコーダ、または図1Aのビデオエンコーダ20などのエンコーダであり得る。20

【 0 1 5 5 】

ビデオコーディングデバイス400は、データを受信するための入口ポート410(または入力ポート410)および受信器ユニット(Rx)420、データを処理するためのプロセッサ、論理ユニット、または中央処理装置(CPU)430、データを送信するための送信器ユニット(Tx)440および出口ポート450(または出力ポート450)、ならびにデータを記憶するためのメモリ460を備える。ビデオコーディングデバイス400はまた、光信号または電気信号の出入るために、入口ポート410、受信器ユニット420、送信器ユニット440、および出口ポート450に結合される、光-電気(OE)コンポーネントおよび電気-光(EO)コンポーネントを備え得る。30

【 0 1 5 6 】

プロセッサ430は、ハードウェアおよびソフトウェアによって実装される。プロセッサ430は、1つまたは複数のCPUチップ、コア(たとえば、マルチコアプロセッサとして)、FPGA、ASIC、およびDSPとして実装され得る。プロセッサ430は、入口ポート410、受信器ユニット420、送信器ユニット440、出口ポート450、およびメモリ460と通信している。プロセッサ430はコーディングモジュール470を備える。コーディングモジュール470は、上で説明された開示された実施形態を実装する。たとえば、コーディングモジュール470は、様々なコーディング動作を実施し、処理し、準備し、または提供する。したがって、コーディングモジュール470を含むことは、ビデオコーディングデバイス400の機能にかなりの改善をもたらし、異なる状態へのビデオコーディングデバイス400の変換を引き起こす。代替的に、コーディングモジュール470は、メモリ460において記憶されプロセッサ430によって実行される命令として実装される。40

【 0 1 5 7 】

メモリ460は、1つまたは複数のディスク、テープドライブ、およびソリッドステートドライブを備えてもよく、プログラムが実行のために選択されるときにそのようなプログラムを記憶するために、およびプログラム実行の間に読み取られる命令とデータを記憶するために、オーバーフローデータストレージデバイスとして使用されてもよい。メモリ460は、たとえば、揮発性および/または不揮発性であってもよく、読み取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、三値連想メモリ(TCAM)、および/またはスタティックランダムアクセスメモリ(SRAM)であってもよい。

【 0 1 5 8 】

図5は、例示的な実施形態による、図1Aからソースデバイス12と宛先デバイス14のいずれかまたは両方として使用され得る、装置500の簡略化されたブロック図である。

【0159】

装置500の中のプロセッサ502は中央処理装置であり得る。代替的に、プロセッサ502は、今存在する、または今後開発される、情報を操作または処理することが可能な、任意の他のタイプのデバイス、または複数のデバイスであり得る。開示された実装形態は、示されるように単一のプロセッサ、たとえばプロセッサ502を用いて実践され得るが、1つより多くのプロセッサを使用すると、速度および効率性において利益を得ることができる。

【0160】

ある実装形態では、装置500のメモリ504は、読み取り専用メモリ(ROM)デバイスまたはランダムアクセスメモリ(RAM)デバイスであり得る。任意の他の適切なタイプのストレージデバイスが、メモリ504として使用され得る。メモリ504は、バス512を使用してプロセッサ502によってアクセスされるコードおよびデータ506を含み得る。メモリ504はさらに、オペレーティングシステム508およびアプリケーションプログラム510を含んでもよく、アプリケーションプログラム510は、ここで説明される方法をプロセッサ502が実行することを可能にする少なくとも1つのプログラムを含む。たとえば、アプリケーションプログラム510は、アプリケーション1からNを含んでもよく、これらはさらに、ここで説明される方法を実行するビデオコーディングアプリケーションを含む。

【0161】

装置500はまた、ディスプレイ518などの1つまたは複数の出力デバイスを含み得る。
ディスプレイ518は、一例では、ディスプレイを、タッチ入力を感知するように動作可能なタッチ感知素子と組み合わせる、タッチ感知ディスプレイであり得る。ディスプレイ518は、バス512を介してプロセッサ502に結合され得る。

【0162】

ここでは単一のバスとして図示されているが、装置500のバス512は複数のバスからなっていてもよい。さらに、二次ストレージ514が、装置500の他の構成要素に直接結合されてもよく、またはネットワークを介してアクセスされてもよく、メモリカードなどの単一の統合されたユニットまたは複数のメモリカードなどの複数のユニットを備えることができる。したがって、装置500は、広範囲の構成で実装され得る。

【0163】

以下はまず、本出願における概念を説明する。

【0164】

1. インター予測モード

HEVCでは、進化型動きベクトル予測(advanced motion vector prediction, AMVP)モードおよびマージ(merge)モードという、2つのインター予測モードが使用される。

【0165】

AMVPモードでは、現在のブロックの空間的または時間的に隣接する符号化されたブロック(隣接ブロックと表記される)がまず走査される。動きベクトル候補リスト(これは動き情報候補リストとも呼ばれる)が、隣接ブロックの動き情報に基づいて構築され、次いで、最適な動きベクトルが、レートひずみコストに基づいて動きベクトル候補リストから決定される。レートひずみコストが最小である動き情報候補が、現在のブロックの動きベクトル予測子(MVP)として使用される。隣接ブロックの位置とその走査順序の両方があらかじめ定められる。レートひずみコストは式(1)に従って計算され、ここで J はレートひずみコスト(RD cost)を表し、SADは、動きベクトル候補予測子を使用することによって動き推定を通じて取得される、元のサンプル値と予測されたサンプル値との絶対値差分(SAD)であり、 R はビットレートを表し、 λ はラグランジュ乗数を表す。エンコーダ側は、動きベクトル候補リストの中の選択された動きベクトル予測子のインデックス値および参照フレームインデックス値を、デコーダ側に伝える。さらに、現在のブロックの実際の動きベクトルを取得するために、動き探索がMVPにおいて近傍で実行される。エンコーダ側は、MVPと実際の動きベクトルとの差(動きベクトル差分)をデコーダ側に転送する。

10

20

30

40

50

J=SAD+ R (1)

【 0 1 6 6 】

マージモードでは、動きベクトル候補リストはまず、現在のブロックの空間的または時間的に隣接する符号化されたブロックの動き情報に基づいて構築される。次いで、レートひずみコストに基づく現在のブロックの動き情報として、最適な動き情報が動きベクトル候補リストから決定される。動きベクトル候補リストにおける最適な動き情報の位置のインデックス値(以後マージインデックスと表記される)が、デコーダ側に伝えられる。現在のブロックの空間的および時間的な動き情報候補が、図6に示される。空間的な動き情報候補は、5つの空間的に隣接するブロック(A0、A1、B0、B1、およびB2)からのものである。隣接ブロックが利用不可能である(隣接ブロックが存在しない、または隣接ブロックが符号化されない、または隣接ブロックのために使用される予測モードがインター予測モードではない)場合、この隣接ブロックの動き情報は動きベクトル候補リストに追加されない。現在のブロックの時間動き情報候補は、参照フレームおよび現在のフレームのピクチャ順序カウント(POC)に基づいて、参照フレームの中の対応する位置においてブロックのMVをスケーリングすることによって取得される。参照フレームの中のTの位置におけるブロックが利用可能であるかどうかがまず決定され、ブロックが利用不可能である場合、Cの位置におけるブロックが選択される。

【 0 1 6 7 】

AMVPモードと同様に、マージモードにおいて、隣接ブロックの位置とその走査順序の両方もあらかじめ定められる。加えて、隣接ブロックの位置およびその横断順序は、異なるモードでは異なり得る。

【 0 1 6 8 】

動きベクトル候補リスト(候補のリストとも呼ばれ、これは略して候補リストと呼ばれ得る)は、AMVPモードとマージモードの両方において維持される必要があることがわかり得る。新しい動き情報が候補リストに追加される前に毎回、同じ動き情報がリストにすでに存在するかどうかがまず確認される。同じ動き情報がすでに存在する場合、動き情報はリストに追加されない。この確認プロセスは、動きベクトル候補リストのプルーニングと呼ばれる。リストのプルーニングは、リストに同じ動き情報が含まれるのを避け、それにより、冗長なレートひずみコストの計算を避けることである。

【 0 1 6 9 】

HEVCにおけるインター予測では、コーディングブロックの中のすべてのサンプルのために同じ動き情報が使用され、そして、コーディングブロックのサンプルの予測子を取得するために、動き情報に基づいて動き補償が実行される。しかしながら、コーディングブロックにおいて、すべてのサンプルが同じ動き特性を有することは限らない。コーディングブロックのために同じ動き情報を使用すると、不正確な動き補償予測およびより多くの残差情報が生じ得る。

【 0 1 7 0 】

既存のビデオコーディング規格では、並進動きモデルに基づくブロックマッチング動き推定が適用され、ブロックの中のすべてのサンプルの動きが一貫していると仮定される。しかしながら、現実の世界では、様々な動きがある。多くの物体、たとえば、回転している物体、様々な方向に回転するローラーコースター、花火の打ち上げ、および映画における何らかのスタント、特にUser Generated Content(UGC)のシナリオにおける動いている物体が、非並進運動をしている。これらの動いている物体に対して、既存のコーディング規格における並進動きモデルに基づくブロック動き補償技術がコーディングのために使用される場合、コーディング効率は大きく影響を受けることがある。したがって、非並進動きモデル、たとえばアフィン動きモデルが、コーディング効率をさらに高めるために導入される。

【 0 1 7 1 】

これに基づいて、異なる動きモデルにより、AMVPモードは、並進モデルベースのAMVPモードおよび非並進モデルベースのAMVPモード(たとえば、アフィンモデルベースのA

10

20

30

40

50

MVPモード)へと分類されてもよく、マージモードは、並進モデルベースのマージモードおよび非並進モデルベースのマージモード(たとえば、アフィンモデルベースのマージモード)へと分類されてもよい。

【0172】

2. 非並進動きモデル

非並進動きモデルに基づく予測は、現在のブロックの中の各サブブロックの動き情報(サブ動き補償ユニットまたは基本動き補償ユニットとも呼ばれる)を導出するために、エンコーダ側とデコーダ側の両方に対して同じ動きモデルが使用され、予測ブロックを取得するために、サブブロックの動き情報に基づいて動き補償が実行され、それにより予測効率を改善することを指す。一般的な非並進動きモデルは、4パラメータのアフィン動きモデルおよび6パラメータのアフィン動きモデルを含む。

10

【0173】

本出願のこの実施形態におけるサブ動き補償ユニット(サブブロックとも呼ばれる)は、特定の区分方法に基づいて取得されるサンプルまたは $N_1 \times N_2$ サンプルブロックであってもよく、 N_1 と N_2 の両方が正の整数であり、 N_1 は N_2 に等しくてもよく、または N_2 に等しくなくてもよい。

【0174】

4パラメータのアフィン動きモデルは、式(2)として表現される。

【0175】

【数1】

20

$$\begin{cases} vx = a_1 + a_3x + a_4y \\ vy = a_2 - a_4x + a_3y \end{cases} \quad (2)$$

【0176】

4パラメータのアフィン動きモデルは、2つのサンプルの動きベクトルおよび現在のブロックの左上サンプルに対するそれらの座標によって表現され得る。動きモデルパラメータを表現するために使用されるサンプルは、制御点と呼ばれる。左上のサンプル(0,0)および右上のサンプル(W,0)が制御点として使用される場合、現在のブロックの左上の制御点および右上の制御点のそれぞれの動きベクトル(vx0,vy0)および(vx1,vy1)がまず決定される。次いで、現在のブロックの各サブ動き補償ユニットの動き情報が、式(3)に従って取得され、(x,y)は現在のブロックの左上サンプルに対する相対的な、サブ動き補償ユニットの座標(左上サンプルの座標など)であり、Wは現在のブロックの幅を表す。他の制御点が代わりに使用されてもよいことを理解されたい。たとえば、位置(2,2)および(W+2,2)、または(-2,-2)および(W-2,-2)におけるサンプルが、制御点として使用され得る。制御点の選択は、本明細書において列挙される例により限定されない。

30

【0177】

【数2】

40

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W}x - \frac{vy_1 - vy_0}{W}y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W}x + \frac{vx_1 - vx_0}{W}y + vy_0 \end{cases} \quad (3)$$

【0178】

6パラメータのアフィン動きモデルは、式(4)として表現される。

【0179】

50

【数3】

$$\begin{cases} vx = a_1 + a_3x + a_4y \\ vy = a_2 + a_5x + a_6y \end{cases} \quad (4)$$

【0180】

6パラメータのアフィン動きモデルは、3つのサンプルの動きベクトルおよび現在のブロックの左上サンプルに対するそれらの座標によって表現され得る。現在のブロックの左上のサンプル(0,0)、右上のサンプル(W,0)、および左下のサンプル(0,H)が制御点として使用される場合、現在のブロックの左上の制御点、右上の制御点、および左下の制御点のそれぞれの動きベクトル(vx0,vy0)、(vx1,vy1)、および(vx2,vy2)がまず決定される。次いで、現在のブロックの各サブ動き補償ユニットの動き情報が、式(5)に従って取得され、(x,y)は現在のブロックの左上サンプルに対する相対的な、サブ動き補償ユニットの座標であり、WおよびHはそれぞれ、現在のブロックの幅および高さを表す。他の制御点が代わりに使用されてもよいことを理解されたい。たとえば、位置(2,2)、(W+2,2)、および(2,H+2)、または(-2,-2)、(W-2,-2)、および(-2,H-2)におけるサンプルが、制御点として使用され得る。これらの例は限定するものではない。

【0181】

【数4】

$$\begin{cases} vx = \frac{vx_1 - vx_0}{w}x + \frac{vx_2 - vy_0}{H}y + vx_0 \\ vy = \frac{vy_1 - vy_0}{w}x + \frac{vy_2 - vx_0}{H}y + vy_0 \end{cases} \quad (5)$$

【0182】

アフィン動きモデルを使用することによって予測されるコーディングブロックは、アフィンコーディングされたブロックと呼ばれる。

【0183】

一般に、アフィンコーディングされたブロックの制御点の動き情報は、アフィン動きモデルベースの高度動きベクトル予測(AMVP)モードまたはアフィン動きモデルベースのマージモードを使用することによって取得され得る。

【0184】

現在のコーディングブロックの制御点の動き情報は、継承された制御点動きベクトル予測方法または構築された制御点動きベクトル予測方法を使用することによって取得され得る。

【0185】

3. 継承された制御点動きベクトル予測方法

継承された制御点動きベクトル予測方法とは、現在のブロックの制御点動きベクトル候補を決定するために、隣接する符号化されたアフィンコーディングされたブロックの動きモデルを使用することを指す。

【0186】

図7に示される現在のブロックが例として使用される。現在のブロックの隣接位置におけるブロックが位置するアフィンコーディングされたブロックを見つけて、アフィンコーディングされたブロックの制御点動き情報を取得するために、現在のブロックの周りの隣接位置におけるブロックが、指定された順序で、たとえばA1 B1 B0 A0 B2の順序で走査される。さらに、現在のブロックの制御点動きベクトル(マージモードのための)ま

10

20

30

40

50

たは制御点動きベクトル予測子(AMVPモードのための)は、アフィンコーディングされたブロックの制御点動き情報に基づいて構築される動きモデルを使用することによって導出される。上で言及された順序 A1 B1 B0 A0 B2 は例として使用されるにすぎず、限定するものとして解釈されるべきではない。別の順序も使用され得る。加えて、隣接位置におけるブロックは、A1、B1、B0、A0、およびB2に限定されず、隣接位置における様々なブロックが使用され得る。

【0187】

隣接位置におけるブロックは、特定の区分方法に基づいて取得されるあらかじめ設定されたサイズのサンプルまたはサンプルブロックでありうる。たとえば、サンプルブロックは、 4×4 サンプルブロック、 4×2 サンプルブロック、または別のサイズのサンプルブロックであり得る。これらのブロックサイズは例示が目的であり、限定するものと解釈されるべきではない。10

【0188】

以下は、A1を例として使用することにより決定プロセスを説明し、同様のプロセスが他の事例に対して利用され得る。

【0189】

図7に示されるように、A1が位置するコーディングブロックが4パラメータのアフィンコーディングされたブロックである場合、アフィンコーディングされたブロックの左上サンプル($x4, y4$)の動きベクトル($vx4, vy4$)および右上サンプル($x5, y5$)の動きベクトル($vx5, vy5$)が取得される。現在のアフィンコーディングされたブロックの左上サンプル($x0, y0$)の動きベクトル($vx0, vy0$)は式(6)に従って計算され、現在のアフィンコーディングされたブロックの右上サンプル($x1, y1$)の動きベクトル($vx1, vy1$)は式(7)に従って計算される。20

【0190】

【数5】

$$\begin{cases} vx_0 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_0 - x_4) - \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (y_0 - y_4) \\ vy_0 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_0 - x_4) + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (y_0 - y_4) \end{cases} \quad (6)$$

$$\begin{cases} vx_1 = vx_4 + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (x_1 - x_4) - \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (y_1 - y_4) \\ vy_1 = vy_4 + \frac{(vy_5 - vy_4)}{x_5 - x_4} \times (x_1 - x_4) + \frac{(vx_5 - vx_4)}{x_5 - x_4} \times (y_1 - y_4) \end{cases} \quad (7)$$
30

【0191】

A1が位置するアフィンコーディングされたブロックに基づいて取得される現在のブロックの左上サンプル($x0, y0$)の動きベクトル($vx0, vy0$)および右上サンプル($x1, y1$)の動きベクトル($vx1, vy1$)の組合せが、現在のブロックの制御点動きベクトル候補である。

【0192】

A1が位置するコーディングブロックが6パラメータのアフィンコーディングされたブロックである場合、アフィンコーディングされたブロックの左上サンプル($x4, y4$)の動きベクトル($vx4, vy4$)、右上サンプル($x5, y5$)の動きベクトル($vx5, vy5$)および左下サンプル($x6, y6$)の動きベクトル($vx6, vy6$)が取得される。現在のブロックの左上サンプル($x0, y0$)の動きベクトル($vx0, vy0$)は、式(8)に従って計算される。現在のブロックの右上サンプル($x1, y1$)の動きベクトル($vx1, vy1$)は、式(9)に従って計算される。現在のブロックの左下サンプル($x2, y2$)の動きベクトル($vx2, vy2$)は、式(10)に従って計算される。40

【0193】

【数6】

$$\begin{cases} vx_0 = vx_4 + \frac{(vx_5-vx_4)}{x_5-x_4} \times (x_0 - x_4) + \frac{(vx_6-vx_4)}{y_6-y_4} \times (y_0 - y_4) \\ vy_0 = vy_4 + \frac{(vy_5-vy_4)}{x_5-x_4} \times (x_0 - x_4) + \frac{(vy_6-vy_4)}{y_6-y_4} \times (y_0 - y_4) \end{cases} \quad (8)$$

$$\begin{cases} vx_1 = vx_4 + \frac{(vx_5-vx_4)}{x_5-x_4} \times (x_1 - x_4) + \frac{(vx_6-vx_4)}{y_6-y_4} \times (y_1 - y_4) \\ vy_1 = vy_4 + \frac{(vy_5-vy_4)}{x_5-x_4} \times (x_1 - x_4) + \frac{(vy_6-vy_4)}{y_6-y_4} \times (y_1 - y_4) \end{cases} \quad (9)$$

$$\begin{cases} vx_2 = vx_4 + \frac{(vx_5-vx_4)}{x_5-x_4} \times (x_2 - x_4) + \frac{(vx_6-vx_4)}{y_6-y_4} \times (y_2 - y_4) \\ vy_2 = vy_4 + \frac{(vy_5-vy_4)}{x_5-x_4} \times (x_2 - x_4) + \frac{(vy_6-vy_4)}{y_6-y_4} \times (y_2 - y_4) \end{cases} \quad (10)$$

10

【0194】

A1が位置するアフィンコーディングされたブロックに基づいて取得される現在のブロックの左上サンプル(x_0, y_0)の動きベクトル(vx_0, vy_0)、右上サンプル(x_1, y_1)の動きベクトル(vx_1, vy_1)、および左下サンプル(x_2, y_2)の動きベクトル(vx_2, vy_2)の組合せが、現在のブロックの制御点動きベクトル候補である。

【0195】

他の動きモデル、位置候補、および探索と走査の順序も、本出願に適用可能であることに留意されたい。本出願のこの実施形態において詳細は説明されない。

20

【0196】

他の制御点が隣接コーディングブロックおよび現在のコーディングブロックの動きモデルを表現するために使用される方法は、本出願にも適用可能であることに留意されたい。詳細は本明細書では説明されない。

【0197】**4. 構築された制御点動きベクトル予測方法1**

構築された制御点動きベクトル予測方法は、現在のブロックの制御点の周りの隣接する符号化されたブロックの動きベクトルを、それらの隣接する符号化されたブロックがアフィンコーディングされたブロックであるかどうかを考慮せずに、現在のアフィンコーディングされたブロックの制御点動きベクトルとして組み合わせることを指す。

30

【0198】

現在のブロックの左上サンプルおよび右上サンプルの動きベクトルは、現在のコーディングブロックの周りの隣接する符号化されたブロックの動き情報を使用することによって決定される。図8Aは、構築された制御点動きベクトル予測方法を説明するために例として使用される。図8Aは例にすぎず、限定するものとして解釈されるべきではないことに留意されたい。

【0199】

図8Aに示されるように、左上サンプルの隣接する符号化されたブロックA2、B2、およびB3の動きベクトルは、現在のブロックの左上サンプルの動きベクトルのための動きベクトル候補として使用され、右上サンプルの隣接する符号化されたブロックB1およびB0の動きベクトルは、現在のブロックの右上サンプルの動きベクトルのための動きベクトル候補として使用される。左上サンプルおよび右上サンプルの動きベクトル候補は、複数の2タブルを構成するように組み合わせられる。2タブルに含まれる2つの符号化されたブロックの動きベクトルは、以下の式(11A)に示されるように、現在のブロックの制御点動きベクトル候補として使用され得る。

$$\{vA2, vB1\}, \{vA2, vB0\}, \{vB2, vB1\}, \{vB2, vB0\}, \{vB3, vB1\}, \{vB3, vB0\} \quad (11A)$$

ここで、 $vA2$ はA2の動きベクトルを表し、 $vB1$ はB1の動きベクトルを表し、 $vB0$ はB0の動きベクトルを表し、 $vB2$ はB2の動きベクトルを表し、 $vB3$ はB3の動きベクトルを表す

40

50

。

【0200】

図8Aに示されるように、左上サンプルの隣接する符号化されたブロックA2、B2、およびB3の動きベクトルは、現在のブロックの左上サンプルの動きベクトルのための動きベクトル候補として使用され、右上サンプルの隣接する符号化されたブロックB1およびB0の動きベクトルは、現在のブロックの右上サンプルの動きベクトルのための動きベクトル候補として使用され、左下サンプルの隣接する符号化されたブロックA0およびA1の動きベクトルは、現在のブロックの左下サンプルの動きベクトルのための動きベクトル候補として使用される。左上サンプル、右上サンプル、および左下サンプルの動きベクトル候補は、3タブルを構成するように組み合わせられる。3タブルに含まれる3つの符号化されたブロックの動きベクトルは、以下の式(11B)および(11C)に示されるように、現在のブロックの制御点動きベクトル候補として使用され得る。

{vA2,vB1,vA0},{vA2,vB0,vA0},{vB2,vB1,vA0},{vB2,vB0,vA0},{vB3,vB1,vA0},{vB3,vB0,vA0} (11B)

{vA2,vB1,vA1},{vA2,vB0,vA1},{vB2,vB1,vA1},{vB2,vB0,vA1},{vB3,vB1,vA1},{vB3,vB0,vA1} (11C)

ここで、vA2はA2の動きベクトルを表し、vB1はB1の動きベクトルを表し、vB0はB0の動きベクトルを表し、vB2はB2の動きベクトルを表し、vB3はB3の動きベクトルを表し、vA0はA0の動きベクトルを表し、vA1はA1の動きベクトルを表す。

【0201】

制御点動きベクトルを組み合わせるための他の方法は、本出願にも適用可能であることに留意されたい。詳細は本明細書では説明されない。

【0202】

他の制御点が隣接コーディングブロックおよび現在のコーディングブロックの動きモデルを表現するために使用される方法は、本出願にも適用可能であることに留意されたい。詳細は本明細書では説明されない。

【0203】

5. 図8Bに示されるような構築された制御点動きベクトル予測方法2:

ステップ501: 現在のブロックの制御点の動き情報を取得する。

【0204】

たとえば、図8Aにおいて、 CP_k ($k=1,2,3,4$)はk番目の制御点を表す。A0、A1、A2、B0、B1、B2、およびB3は、現在のブロックの空間的に隣接する位置であり、CP1、CP2、またはCP3を予測するために使用され、Tは、現在のブロックの時間的に隣接する位置であり、CP4を予測するために使用される。

【0205】

CP1、CP2、CP3、およびCP4の座標はそれぞれ、(0,0)、(W,0)、(H,0)、および(W,H)であると想定され、WおよびHは現在のブロックの幅および高さを表す。

【0206】

各制御点に対して、その動き情報が以下の順序で取得される。

【0207】

(1)CP1に対して、確認順序はB2 A2 B3である。B2が利用可能である場合、B2の動き情報がCP1のために使用される。それ以外の場合、A2およびB3が順番に確認される。すべての3つの位置の動き情報が利用不可能である場合、CP1の動き情報を取得することができない。

【0208】

(2)CP2に対して、確認順序はB0 B1である。B0が利用可能である場合、B0の動き情報がCP2のために使用される。それ以外の場合、B1が確認される。両方の位置の動き情報が利用不可能である場合、CP2の動き情報を取得することができない。

【0209】

(3)CP3に対して、確認順序はA0 A1である。A0が利用可能である場合、A0の動き情

10

20

30

40

50

報がCP3のために使用される。それ以外の場合、A1が確認される。両方の位置の動き情報が利用不可能である場合、CP3の動き情報を取得することができない。

【0210】

(4) CP4に対して、Tの動き情報が使用される。

【0211】

本明細書では、Xが利用可能であるとは、ブロックX(たとえば、A0、A1、A2、B0、B1、B2、B3、またはT)がすでに符号化されており、インター予測モードが使用されることを意味する。それ以外の場合、Xは利用不可能である。

【0212】

制御点の動き情報を取得するための他の方法は、本出願にも適用可能であることに留意されたい。詳細は本明細書では説明されない。

10

【0213】

ステップ502: 構築された制御点動き情報を取得するために、制御点の動き情報を組み合わせる。

【0214】

4パラメータのアフィン動きモデルを構築するために、2つの制御点の動き情報が組み合わせられて、2タブルを構成する。2つの制御点の動き情報の組合せは、{CP1,CP4}、{CP2,CP3}、{CP1,CP2}、{CP2,CP4}、{CP1,CP3}、および{CP3,CP4}であり得る。たとえば、制御点CP1およびCP2の動き情報を含む2タブルを使用することによって構築される4パラメータのアフィン動きモデルは、Affine(CP1,CP2)と表記され得る。

20

【0215】

6パラメータのアフィン動きモデルを構築するために、3つの制御点の動き情報が組み合わせられて、3タブルを構成する。3つの制御点の動き情報の組合せは、{CP1,CP2,CP4}、{CP1,CP2,CP3}、{CP2,CP3,CP4}、および{CP1,CP3,CP4}であり得る。たとえば、制御点CP1、CP2、およびCP3の動き情報を含む3タブルを使用することによって構築される6パラメータのアフィン動きモデルは、Affine(CP1,CP2,CP3)と表記され得る。

【0216】

8パラメータの双線形動きモデルを構築するために、4つの制御点の動き情報が組み合わせられて、4タブルを構成する。制御点CP1、CP2、CP3、およびCP4の動き情報を含む4タブルを使用することによって構築される8パラメータの双線形動きモデルは、Bilinear(CP1,CP2,CP3,CP4)と表記され得る。

30

【0217】

本出願のこの実施形態では、説明を簡単にするために、2つの制御点(または2つの符号化されたブロック)の動き情報の組合せは単に2タブルと呼ばれ、3つの制御点(または3つの符号化されたブロック)の動き情報の組合せは単に3タブルと呼ばれ、4つの制御点(または4つの符号化されたブロック)の動き情報の組合せは単に4タブルと呼ばれる。

【0218】

これらのモデルはあらかじめ設定された順序で走査される。組合せモデルに対応する制御点の動き情報が利用不可能である場合、モデルは利用不可能であると考えられる。それ以外の場合、モデルの参照フレームインデックスが決定され、制御点動きベクトルがスケーリングされる。スケーリングの後のすべての制御点の動き情報が一貫している場合、モデルは無効である。モデルを制御する制御点のすべての動き情報が利用可能であり、モデルが有効である場合、モデルを構築する制御点の動き情報が、動き情報候補リストに追加される。

40

【0219】

制御点動きベクトルスケーリング方法が式(12)に示される。

【0220】

【数7】

50

$$\mathbf{MV}_s = \frac{\text{CurPoc}-\text{DesPoc}}{\text{CurPoc}-\text{SrcPoc}} \times \mathbf{MV} \quad (12)$$

【0221】

ここで、CurPocは現在のフレームのPOC数を表し、DesPocは現在のブロックの参照フレームのPOC数を表し、SrcPocは制御点の参照フレームのPOC数を表し、 \mathbf{MV}_s はスケーリングの後に取得される動きベクトルを表し、 \mathbf{MV} は制御点の動きベクトルを表す。

10

【0222】

異なる制御点の組合せが、同じ位置において制御点へと変換され得ることに留意されたい。

【0223】

たとえば、組合せ{CP1,CP4}、{CP2,CP3}、{CP2,CP4}、{CP1,CP3}、または{CP3,CP4}を通じて取得される4パラメータのアフィン動きモデルは、{CP1,CP2}または{CP1,CP2,CP3}による表現へと変換される。変換方法は、制御点{CP1,CP4}、{CP2,CP3}、{CP2,CP4}、{CP1,CP3}、または{CP3,CP4}の動きベクトルおよび座標情報を式(2)へと置換してモデルパラメータを取得するステップと、{CP1,CP2}の座標情報を式(3)へと置換して制御点{CP1,CP2}の動きベクトルを取得するステップとを含む。

20

【0224】

より直接的には、以下の式(13)から(21)に従って変換が実行されてもよく、ここでWは現在のブロックの幅を表し、Hは現在のブロックの高さを表す。式(13)から(21)において、 (vx_0,vy_0) はCP1の動きベクトルを表し、 (vx_1,vy_1) はCP2の動きベクトルを表し、 (vx_2,vy_2) はCP3の動きベクトルを表し、 (vx_3,vy_3) はCP4の動きベクトルを表す。

【0225】

{CP1,CP2}は、以下の式(13)を使用することによって、{CP1,CP2,CP3}へと変換され得る。言い換えると、{CP1,CP2,CP3}の中のCP3の動きベクトルは、式(13)を使用することによって決定され得る。

【0226】

30

【数8】

$$\begin{cases} vx_2 = -\frac{vy_1-vy_0}{W}H + vx_0 \\ vy_2 = +\frac{vx_1-vx_0}{W}H + vy_0 \end{cases} \quad (13)$$

【0227】

{CP1,CP3}は、以下の式(14)を使用することによって、{CP1,CP2}または{CP1,CP2,CP3}へと変換され得る:

40

【0228】

【数9】

$$\begin{cases} vx_1 = +\frac{vy_2-vy_0}{H}W + vx_0 \\ vy_1 = -\frac{vx_2-vx_0}{H}W + vy_0 \end{cases} \quad (14)$$

【0229】

{CP2,CP3}は、以下の式(15)を使用することによって、{CP1,CP2}または{CP1,CP2,CP3}へと変換され得る:

50

P3}へと変換され得る:

【0 2 3 0】

【数1 0】

$$\begin{cases} vx_0 = \frac{vx_2 - vx_1}{W*W+H*H} W * W - \frac{vy_2 - vy_1}{W*W+H*H} H * W + vx_1 \\ vy_0 = \frac{vy_2 - vy_1}{W*W+H*H} W * W + \frac{vx_2 - vx_1}{W*W+H*H} H * W + vy_1 \end{cases} \quad (15)$$

10

【0 2 3 1】

{CP1,CP4}は、以下の式(16)または(17)を使用することによって、{CP1,CP2}または{CP1,CP2,CP3}へと変換され得る:

【0 2 3 2】

【数1 1】

$$\begin{cases} vx_1 = \frac{vx_3 - vx_0}{W*W+H*H} W * W + \frac{vy_3 - vy_0}{W*W+H*H} H * W + vx_0 \\ vy_1 = \frac{vy_3 - vy_0}{W*W+H*H} W * W - \frac{vx_3 - vx_0}{W*W+H*H} H * W + vy_0 \end{cases} \quad (16)$$

20

$$\begin{cases} vx_2 = \frac{vx_3 - vx_0}{W*W+H*H} H * H - \frac{vy_3 - vy_0}{W*W+H*H} H * W + vx_0 \\ vy_2 = \frac{vy_3 - vy_0}{W*W+H*H} W * H + \frac{vx_3 - vx_0}{W*W+H*H} H * H + vy_0 \end{cases} \quad (17)$$

30

【0 2 3 3】

{CP2,CP4}は、以下の式(18)を使用することによって{CP1,CP2}へと変換されてもよく、{CP2,CP4}は、以下の式(18)および(19)を使用することによって{CP1,CP2,CP3}へと変換されてもよい。

【0 2 3 4】

【数1 2】

$$\begin{cases} vx_0 = -\frac{vy_3 - vy_1}{H} W + vx_1 \\ vy_0 = +\frac{vx_3 - vx_1}{H} W + vy_1 \end{cases} \quad (18)$$

$$\begin{cases} vx_2 = -\frac{vy_3 - vy_1}{H} W + vx_3 \\ vy_2 = +\frac{vx_3 - vx_1}{H} W + vy_3 \end{cases} \quad (19)$$

40

【0 2 3 5】

{CP3,CP4}は、以下の式(20)を使用することによって{CP1,CP2}へと変換されてもよく、{CP3,CP4}は、以下の式(20)および(21)を使用することによって{CP1,CP2,CP3}へと変換されてもよい。

【0 2 3 6】

【数1 3】

50

$$\begin{cases} vx_0 = +\frac{vy_3 - vy_2}{W} H + vx_2 \\ vy_0 = -\frac{vx_3 - vx_2}{W} H + vy_2 \end{cases} \quad (20)$$

$$\begin{cases} vx_1 = +\frac{vy_3 - vy_2}{W} H + vx_3 \\ vy_1 = -\frac{vx_3 - vx_2}{W} H + vy_3 \end{cases} \quad (21)$$

10

【0237】

たとえば、組合せ{CP1,CP2,CP4}、{CP2,CP3,CP4}、または{CP1,CP3,CP4}を通じて取得される6パラメータのアフィン動きモデルは、{CP1,CP2,CP3}による表現へと変換され得る。変換方法は、制御点{CP1,CP2,CP4}、{CP2,CP3,CP4}、または{CP1,CP3,CP4}の動きベクトルおよび座標情報を式(4)へと置換してモデルパラメータを取得するステップと、{CP1,CP2,CP3}の座標情報を式(5)へと置換して{CP1,CP2,CP3}の動きベクトルを取得するステップとを含む。

【0238】

より直接的には、以下の式(22)から(24)に従って変換が実行されてもよく、ここでWは現在のブロックの幅を表し、Hは現在のブロックの高さを表す。式(13)から(21)において、 (vx_0, vy_0) はCP1の動きベクトルを表し、 (vx_1, vy_1) はCP2の動きベクトルを表し、 (vx_2, vy_2) はCP3の動きベクトルを表し、 (vx_3, vy_3) はCP4の動きベクトルを表す。

20

【0239】

{CP1,CP2,CP4}は、以下の式(22)を使用することによって、{CP1,CP2,CP3}へと変換され得る：

【0240】

【数14】

$$\begin{cases} vx_2 = vx_3 + vx_0 - vx_1 \\ vy_2 = vy_3 + vy_0 - vy_1 \end{cases} \quad (22)$$

30

【0241】

{CP2,CP3,CP4}は、以下の式(23)を使用することによって、{CP1,CP2,CP3}へと変換され得る：

【0242】

【数15】

40

$$\begin{cases} vx_0 = vx_1 + vx_2 - vx_3 \\ vy_0 = vy_1 + vy_2 - vy_3 \end{cases} \quad (23)$$

【0243】

{CP1,CP3,CP4}は、以下の式(24)を使用することによって、{CP1,CP2,CP3}へと変換され得る：

【0244】

【数16】

50

$$\begin{cases} vx_1 = vx_3 + vx_0 - vx_2 \\ vy_1 = vy_3 + vy_0 - vy_2 \end{cases} \quad (24)$$

【0245】

6. アフィン動きモデルベースの高度動きベクトル予測モード(アフィンAMVPモード) 10

(1)動きベクトル候補リストの構築

アフィン動きモデルベースのAMVPモードのための動きベクトル候補リストは、上で説明された、継承された制御点動きベクトル予測方法および/または構築された制御点動きベクトル予測方法を使用することによって構築される。本出願のこの実施形態では、アフィン動きモデルベースのAMVPモードのための動きベクトル候補リストは、制御点動きベクトル予測子候補リストと呼ばれ得る。各制御点の動きベクトル予測子は、2つの(4パラメータアフィン動きモデル)制御点の動きベクトルまたは3つの(6パラメータアフィン動きモデル)制御点の動きベクトルを含む。

【0246】

任意選択で、制御点動きベクトル予測子候補リストは、特定の規則に従って剪定されて分類され、特定の量の制御点動きベクトル予測子候補を含むように、切り詰められ、またはパディングされ得る。 20

【0247】

(2)最適な制御点動きベクトル予測子候補の決定

エンコーダ側で、現在のコーディングブロックの中の各サブ動き補償ユニットの動きベクトルは、式(3)または(5)を使用することによって、制御点動きベクトル予測子候補リストの中の各制御点動きベクトル予測子候補(たとえば、Xタブル候補)に基づいて取得される。取得された動きベクトルは、サブ動き補償ユニットの動きベクトルが指示する参照フレームの中の対応する位置にあるサンプル値を取得するために使用され得る。このサンプル値は、アフィン動きモデルを使用して動き補償を行うための予測子として使用される。現在のコーディングブロックの中の各サンプルの元の値と予測値との平均の差分が計算される。最小の平均の差分に対応する制御点動きベクトル予測子候補が、最適な制御点動きベクトル予測子候補として選択され、現在のコーディングブロックの2つまたは3つの制御点の動きベクトル予測子として使用される。制御点動きベクトル予測子候補リストにおける最適な制御点動きベクトル予測子候補(たとえば、Xタブル候補)の位置を表すインデックス番号は、ビットストリームへと符号化され、デコーダへ送信される。 30

【0248】

デコーダ側で、インデックス番号が解析され、制御点動きベクトル予測子(CPMVP)(たとえば、Xタブル候補)が、インデックス番号に基づいて制御点動きベクトル予測子候補リストから決定される。

【0249】

(3)制御点動きベクトルの決定 40

エンコーダ側で、制御点動きベクトル予測子は、制御点動きベクトル(CPMV)を取得するために、特定の探索範囲内での動き探索のための探索開始点として使用される。それぞれの制御点動きベクトルと制御点動きベクトル予測子との差分(制御点動きベクトル差分、CPMVD)は、デコーダ側に伝えられる。

【0250】

デコーダ側で、制御点動きベクトル差分が、ビットストリームを解析することによって取得され、それぞれの制御点動きベクトルを取得するために、それぞれ制御点動きベクトル予測子に加算される。

【0251】

7. アフィンマージモード

制御点動きベクトルマージ候補リストは、上で説明された、継承された制御点動きベクトル予測方法および/または構築された制御点動きベクトル予測方法を使用することによって構築される。

【0252】

任意選択で、制御点動きベクトルマージ候補リストは、特定の規則に従って剪定されて分類され、特定の量へと切り詰められ、またはパディングされ得る。

【0253】

エンコーダ側で、現在のコーディングブロックの中の各サブ動き補償ユニット(特定の区分方法に基づいて取得されるサンプルまたは $N_1 \times N_2$ サンプルブロック)の動きベクトルは、式(3)または(5)を使用することによって、マージ候補リストの中の各制御点動きベクトル候補(たとえば、Xタブル候補)に基づいて取得される。取得された動きベクトルは、各サブ動き補償ユニットの動きベクトルが指示する参照フレームの中の位置にあるサンプル値を取得するために使用され得る。これらのサンプル値は、アフィン動き補償を行うために、予測されたサンプル値として使用される。現在のコーディングブロックの中の各サンプルの元の値と予測される値との平均の差分が計算される。最小の平均の差分に対応する制御点動きベクトル(CPMV)候補(たとえば、2タブル候補または3タブル候補)は、現在のコーディングブロックの2つまたは3つの制御点の動きベクトルとして選択される。候補リストの中の制御点動きベクトルの位置を表すインデックス番号は、ビデオのビットストリームへと符号化され、デコーダに送信される。10

【0254】

デコーダ側で、インデックス番号が解析され、制御点動きベクトル(CPMV)が、インデックス番号に基づいて制御点動きベクトルマージ候補リストから決定される。20

【0255】

加えて、本出願では、「少なくとも1つ」は1つまたは複数を意味し、「複数の」は少なくとも2つを意味することに留意されたい。「および/または」という用語は、関連する物体を記述するための相関関係を記述し、3つの関係が存在し得ることを表す。たとえば、Aおよび/またはBは以下の事例を表し得る。Aのみが存在する、AとBの両方が存在する、およびBのみが存在する。ここでAおよびBは単数形または複数形であり得る。文字「/」は一般に、関連する物体間の「または」の関係を示す。「以下[の項目]のうちの少なくとも1つ(1個)」または同様の表現は、単一の項目(断片)または複数の項目(断片)の任意の組合せを含めて、これらの項目の任意の組合せを指す。たとえば、a、b、またはcのうちの少なくとも1つ(1個)は、a、b、c、aおよびb、aおよびc、bおよびc、またはaおよびbおよびcを表してもよく、a、b、およびcは単数または複数であってもよい。30

【0256】

本出願では、現在のブロックを復号するためにインター予測モードが使用されるとき、インター予測モードをシグナリングするためにシンタックス要素が使用され得る。

【0257】

現在のブロックを解析するために使用されるインター予測モードのいくつかの現在使用されているシンタックス構造については、インター予測モードのための一部のシンタックスが列挙されているTable 1(表1)を参照されたい。代替的に、シンタックス構造の中のシンタックス要素は、他の識別子によって表され得る。40

【0258】

50

50

50

50

50

【表 1 A】

Table 1

記述子	
coding_unit(x0, y0, cbWidth, cbHeight, treeType) {	
if(cu_skip_flag[x0][y0] == 0)	
merge_flag[x0][y0]	ae(v)
if(merge_flag[x0][y0]) {	
merge_data(x0, y0, cbWidth, cbHeight)	
} else if (CuPredMode[x0][y0] == MODE_IBC) {	
mvd_coding(x0, y0, 0, 0)	
mvp_l0_flag[x0][y0]	ae(v)
if(sps_amvr_enabled_flag &&	
(MvdL0[x0][y0][0] != 0 MvdL0[x0][y0][1] != 0)) {	
amvr_precision_flag[x0][y0]	ae(v)
}	
} else {	
if(tile_group_type == B)	
inter_pred_idc[x0][y0]	ae(v)
if(sps_affine_enabled_flag && cbWidth >= 16 && cbHeight >= 16) {	
inter_affine_flag[x0][y0]	ae(v)
if(sps_affine_type_flag && inter_affine_flag[x0][y0])	
cu_affine_type_flag[x0][y0]	ae(v)
}	
if(inter_pred_idc[x0][y0] == PRED_BI && !inter_affine_flag[x0][y0] &&	
RefIdxSymL0 > -1 && RefIdxSymL1 > -1)	
sym_mvd_flag[x0][y0]	ae(v)
if(inter_pred_idc[x0][y0] != PRED_L1) {	
if(NumRefIdxActive[0] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l0[x0][y0]	ae(v)
mvd_coding(x0, y0, 0, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 0, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 0, 2)	
mvp_l0_flag[x0][y0]	ae(v)
} else {	
MvdL0[x0][y0][0] = 0	
MvdL0[x0][y0][1] = 0	
}	
if(inter_pred_idc[x0][y0] != PRED_L0) {	
if(NumRefIdxActive[1] > 1 && !sym_mvd_flag[x0][y0])	
ref_idx_l1[x0][y0]	ae(v)
if(mvd_l1_zero_flag && inter_pred_idc[x0][y0] == PRED_BI) {	

10

20

30

40

【 0 2 5 9 】

50

【表 1 B】

MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
MvdCpL1[x0][y0][0][0] = 0	
MvdCpL1[x0][y0][0][1] = 0	
MvdCpL1[x0][y0][1][0] = 0	
MvdCpL1[x0][y0][1][1] = 0	
MvdCpL1[x0][y0][2][0] = 0	
MvdCpL1[x0][y0][2][1] = 0	
}	
else {	
if(sym_mvd_flag[x0][y0]) {	
MvdL1[x0][y0][0] = -MvdL0[x0][y0][0]	
MvdL1[x0][y0][1] = -MvdL0[x0][y0][1]	
} else	
mvd_coding(x0, y0, 1, 0)	
if(MotionModelIdc[x0][y0] > 0)	
mvd_coding(x0, y0, 1, 1)	
if(MotionModelIdc[x0][y0] > 1)	
mvd_coding(x0, y0, 1, 2)	
mvp_ll_flag[x0][y0]	ae(v)
}	
} else {	
MvdL1[x0][y0][0] = 0	
MvdL1[x0][y0][1] = 0	
}	
....	
}	
....	
}	
....	
}	
....	
}	
....	
}	
....	
}	
....	
}	
....	
}	

10

20

30

【0 2 6 0】

Table 1(表1)において、1に等しいinter_affine_flag[x0][y0]は、PまたはBタイルグループを復号するとき、現在のコーディングユニットに対して、現在のコーディングユニットの予測サンプルを生成するためにアフィンモデルベースの動き補償が使用されることを指定する。0に等しいinter_affine_flag[x0][y0]は、コーディングユニットがアフィンモデルベースの動き補償によって予測されないことを指定する。inter_affine_flag[x0][y0]が存在しないとき、それは0に等しいと推測される。

【0 2 6 1】

inter_pred_idc[x0][y0]は、Table 2(表2)に従って、現在のコーディングユニットのためにlist0が使用されるか、list1が使用されるか、または双予測が使用されるかを指定する。アレイインデックスx0、y0は、ピクチャの左上ルマサンプルに対する相対的な、考慮されるコーディングブロックの左上ルマサンプルの位置(x0,y0)を指定する。

【0 2 6 2】

inter_pred_idc[x0][y0]が存在しないとき、それはPRED_L0に等しいと推測される。

【0 2 6 3】

40

50

【表 2】

Table 2

inter_pred_idc	inter_pred_idc の名称	
	(cbWidth + cbHeight) != 8	(cbWidth + cbHeight) == 8
0	PRED_L0	PRED_L0
1	PRED_L1	PRED_L1
2	PRED_BI	n.a.

10

【0 2 6 4】

sps_affine_enabled_flagは、アフィンモデルベースの動き補償がインター予測のために使用され得るかどうかを指定する。sps_affine_enabled_flagが0に等しい場合、シンタックスは、アフィンモデルベースの動き補償がCVSにおいて使用されず、inter_affine_flagおよびcu_affine_type_flagがCVSのコーディングユニットシンタックスにおいて存在しないように、制約されるものとする。それ以外(sps_affine_enabled_flagが1に等しい)の場合、アフィンモデルベースの動き補償がCVSにおいて使用され得る。

20

【0 2 6 5】

シンタックス要素inter_affine_flag[x0][y0](またはaffine_inter_flag[x0][y0])は、現在のブロックが位置するスライスがPタイプスライスまたはBタイプスライスであるときに、アフィン動きモデルベースのAMVPモードが現在のブロックのために使用されるかどうかを示すために使用され得る。このシンタックス要素がビットストリームにおいて現れないとき、シンタックス要素はデフォルトで0である。たとえば、inter_affine_flag[x0][y0]=1は、アフィン動きモデルベースのAMVPモードが現在のブロックのために使用されることを示し、inter_affine_flag[x0][y0]=0は、アフィン動きモデルベースのAMVPモードが現在のブロックのために使用されず、並進動きモデルベースのAMVPモードが使用され得ることを示す。すなわち、1に等しいinter_affine_flag[x0][y0]は、PまたはBタイルグループを復号するとき、現在のコーディングユニットに対して、現在のコーディングユニットの予測サンプルを生成するためにアフィンモデルベースの動き補償が使用されることを指定する。0に等しいinter_affine_flag[x0][y0]は、コーディングユニットがアフィンモデルベースの動き補償によって予測されないことを指定する。inter_affine_flag[x0][y0]が存在しないとき、それは0に等しいと推測される。

30

【0 2 6 6】

シンタックス要素cu_affine_type_flag[x0][y0]は、現在のブロックが位置するスライスがPタイプスライスまたはBタイプスライスであるとき、6パラメータのアフィン動きモデルが現在のブロックのための動き補償を行うために使用されるかどうかを示すために使用され得る。cu_affine_type_flag[x0][y0]=0は、現在のブロックのための動き補償を行うために6パラメータのアフィン動きモデルが使用されず、動き補償を行うために4パラメータのアフィン動きモデルだけが使用され得ることを示し、cu_affine_type_flag[x0][y0]=1は、現在のブロックのための動き補償を行うために6パラメータのアフィン動きモデルが使用されることを示す。すなわち、1に等しいcu_affine_type_flag[x0][y0]は、PまたはBタイルグループを復号するとき、現在のコーディングユニットに対して、現在のコーディングユニットの予測サンプルを生成するために6パラメータのアフィンモデルベースの動き補償が使用されることを指定する。0に等しいcu_affine_type_flag[x0][y0]

40

50

は、現在のコーディングユニットの予測サンプルを生成するために4パラメータのアフィンモデルベースの動き補償が使用されることを指定する。

【0267】

Table 3(表3)に示されるように、MotionModelIdc[x0][y0]=1は、4パラメータのアフィン動きモデルが使用されることを示し、MotionModelIdc[x0][y0]=2は、6パラメータのアフィン動きモデルが使用されることを示し、MotionModelIdc[x0][y0]=0は、並進動きモデルが使用されることを示す。

【0268】

【表3】

10

Table 3

	動き補償のための動きモデル(motion model for motion compensation)
0	並進動き(translational motion)
1	4パラメータアフィン動き(4-parameter affine motion)
2	6パラメータアフィン動き(6-parameter affine motion)

20

【0269】

最大のリスト長を表し、構築された動きベクトル候補リストの最大の長さを示すために、変数MaxNumMergeCandおよびMaxAffineNumMrgCandが使用される。予測方向を示すために、inter_pred_idc[x0][y0]が使用される。後方予測を示すために、PRED_L1が使用される。num_ref_idx_l0_active_minus1は、前方参照フレームリストの中の参照フレームの数を示し、ref_idx_l0[x0][y0]は、現在のブロックの前方参照フレームインデックス値を示す。mvd_coding(x0, y0, 0, 0)は、第1の動きベクトル差分を示す。mvp_l0_flag[x0][y0]は、前方MVP候補リストインデックス値を示す。PRED_L0は前方予測を示す。num_ref_idx_l1_active_minus1は、後方参照フレームリストの中の参照フレームの数を示す。ref_idx_l1[x0][y0]は、現在のブロックの後方参照フレームインデックス値を示し、mvp_l1_flag[x0][y0]は、後方MVP候補リストインデックス値を示す。

30

【0270】

Table 1(表1)において、ae(v)は、コンテキストベースの適応バイナリ算術コーディング(cabac)を使用することによって符号化されるシンタックス要素を表す。

【0271】

図9Aは、本出願のある実施形態による復号方法のプロセスを示すフローチャートである。プロセスは、ビデオデコーダ30のインター予測ユニット344によって実行され得る。プロセスは、一連のステップまたは動作として説明される。プロセスは、様々な順序で実行されてもよく、および/または同時に実行されてもよく、図9Aに示される実行順序に限定されないことを理解されたい。ビデオデコーダは、図9Aに示されるインター予測プロセスを含むプロセスを使用することによって、複数のビデオフレームを有するビデオデータストリームを復号するために利用されることが想定される。

40

【0272】

ステップ601: 現在のブロックのインター予測モードを決定するために、Table 1(表1)に示されるシンタックス構造に基づいてビットストリームを解析する。

【0273】

50

現在のブロックのインター予測モードがアフィン動きモデルベースのAMVPモードであると決定される場合、ステップ602aを実行する。

【0274】

たとえば、シンタックス要素merge_flag=0およびinter_affine_flag=1は、現在のブロックのインター予測モードがアフィン動きモデルベースのAMVPモードであることを示す。

【0275】

現在のブロックのインター予測モードがアフィン動きモデルベースのマージモードであると決定される場合、ステップ602bを実行する。

【0276】

たとえば、シンタックス要素merge_flag=1およびinter_affine_flag=1は、現在のブロックのインター予測モードがアフィン動きモデルベースのマージモードであることを示す。

【0277】

ステップ602a: アフィン動きモデルベースのAMVPモードに対応する動きベクトル候補リストを構築する。

【0278】

現在のブロックの1つまたは複数の制御点動きベクトル候補(たとえば、1つまたは複数のXタプル候補)は、継承された制御点動きベクトル予測方法および/または構築された制御点動きベクトル予測方法を使用することによって導出され得る。これらの制御点動きベクトル候補は、動きベクトル候補リストに追加され得る。

20

【0279】

動きベクトル候補リストは、2タブルリスト(4パラメータのアフィン動きモデルが現在のコーディングブロックのために使用される)または3タブルリストを含み得る。2タブルリストは、4パラメータのアフィン動きモデルを構築するために使用される1つまたは複数の2タブルを含む。3タブルリストは、6パラメータのアフィン動きモデルを構築するために使用される1つまたは複数の3タブルを含む。各々の2タブル候補は、現在のブロックの2つの制御点動きベクトル候補を含むことが理解され得る。

【0280】

任意選択で、動きベクトル候補の2タブル/3タブルリストは、特定の規則に従って剪定されて分類され、特定の量の2タブルまたは3タブルを含むように、切り詰められ、またはパディングされ得る。

30

【0281】

A1: 継承された制御点動きベクトル予測方法を使用することによって動きベクトル候補リストを構築するプロセスが説明される。

【0282】

図7が例として使用される。この例では、現在のブロックの隣接位置におけるブロックを含むアフィンコーディングされたブロックを見つけて、アフィンコーディングされたブロックの制御点動き情報を取得するために、現在のブロックの周りの隣接位置におけるブロックが、A1 B1 B0 A0 B2の順序で走査される。アフィンコーディングされたブロックの制御点動き情報は、動きモデルを構築して現在のブロックの制御点動き情報候補を導出するために利用され得る。このプロセスの詳細は、3における継承された制御点動きベクトル予測方法の説明において上で与えられる。

40

【0283】

一例では、現在のブロックのために使用されるアフィン動きモデルは、4パラメータのアフィン動きモデルである(すなわち、MotionModelIdc=1)。この例では、4パラメータのアフィン動きモデルが隣接アフィン復号ブロックのために使用される場合、そのアフィン復号ブロックの2つの制御点の動きベクトルである、左上の制御点(x4,y4)の動きベクトル(vx4,vy4)および右上の制御点(x5,y5)の動きベクトル(vx5,vy5)が取得される。アフィン復号ブロックは、アフィン動きモデルを使用することによって、符号化段階において予

50

測されるアフィンコーディングされたブロックである。

【 0 2 8 4 】

現在のブロックの2つの制御点の動きベクトル、すなわち、左上および右上の制御点は、隣接アフィン復号ブロックの2つの制御点を含む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、それぞれ、4パラメータのアフィン動きモデルの式(6)および(7)に従って導出される。

【 0 2 8 5 】

6パラメータのアフィン動きモデルが隣接アフィン復号ブロックのために使用される場合、隣接アフィン復号ブロックの3つの制御点の動きベクトル、たとえば、図7の左上の制御点(x_4, y_4)の動きベクトル(vx_4, vy_4)、右上の制御点(x_5, y_5)の動きベクトル(vx_5, vy_5)、および左下の制御点(x_6, y_6)の動きベクトル(vx_6, vy_6)が取得される。10

【 0 2 8 6 】

現在のブロックの2つの制御点の動きベクトル、すなわち、左上および右上の制御点は、隣接アフィン復号ブロックの3つの制御点を含む6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、それぞれ、6パラメータのアフィン動きモデルの式(8)および(9)に従って導出される。

【 0 2 8 7 】

別の例では、現在の復号ブロックのために使用されるアフィン動きモデルは、6パラメータのアフィン動きモデルである(すなわち、MotionModelDc=2)。

【 0 2 8 8 】

隣接アフィン復号ブロックのために使用されるアフィン動きモデルが6パラメータのアフィン動きモデルである場合、隣接アフィン復号ブロックの3つの制御点の動きベクトル、たとえば、図7の左上の制御点(x_4, y_4)の動きベクトル(vx_4, vy_4)、右上の制御点(x_5, y_5)の動きベクトル(vx_5, vy_5)、および左下の制御点(x_6, y_6)の動きベクトル(vx_6, vy_6)が取得される。20

【 0 2 8 9 】

現在のブロックの3つの制御点の動きベクトル、すなわち、左上、右上、および左下の制御点は、隣接アフィン復号ブロックの3つの制御点を含む6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、それぞれ、6パラメータのアフィン動きモデルに対応する式(8)、(9)、および(10)に従って導出される。30

【 0 2 9 0 】

隣接アフィン復号ブロックのために使用されるアフィン動きモデルが4パラメータのアフィン動きモデルである場合、隣接アフィン復号ブロックの2つの制御点の動きベクトルが取得され得る。これらの動きベクトルは、たとえば、左上の制御点(x_4, y_4)の動きベクトル(vx_4, vy_4)および右上の制御点(x_5, y_5)の動きベクトル(vx_5, vy_5)であり得る。

【 0 2 9 1 】

現在のブロックの左上、右上、および左下の制御点などの、3つの制御点の動きベクトルが導出され得る。たとえば、これらの動きベクトルは、隣接アフィン復号ブロックの2つの制御点に基づいて表される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、4パラメータのアフィン動きモデルの式(6)および(7)に従って導出され得る。40

【 0 2 9 2 】

他の動きモデル、位置候補、および探索の順序も、本明細書において利用され得ることに留意されたい。さらに、他の制御点に基づいて隣接するコーディングブロックおよび現在のコーディングブロックの動きモデルを表現するための方法も、使用され得る。

【 0 2 9 3 】

A2: 構築された制御点動きベクトル予測方法を使用することによって動きベクトル候補リストを構築するプロセスが説明される。

【 0 2 9 4 】

一例では、現在の復号ブロックのために使用されるアフィン動きモデルは、4パラメータのアフィン動きモデルである(すなわち、MotionModelDc=1)。この例では、現在のコ50

コーディングブロックの左上サンプルおよび右上サンプルの動きベクトルは、現在のコーディングブロックの隣接する符号化されたブロックの動き情報に基づいて決定される。具体的には、動きベクトル候補リストは、項目4に関して上で説明された、構築された制御点動きベクトル予測方法1、または項目5に関して上で説明された、構築された制御点動きベクトル予測方法2を使用することによって、構築され得る。

【0295】

別の例では、現在の復号ブロックのために使用されるアフィン動きモデルは、6パラメータのアフィン動きモデルである(すなわち、MotionModelIdc=2)。この例では、現在のコーディングブロックの左上サンプル、右上サンプル、および左下サンプルの動きベクトルは、現在のコーディングブロックの隣接する符号化されたブロックの動き情報を使用することによって決定される。具体的には、動きベクトル候補リストは、項目4に関して上で説明された、構築された制御点動きベクトル予測方法1、または項目5に関して上で説明された、構築された制御点動きベクトル予測方法2を使用することによって、構築され得る。

10

【0296】

制御点動き情報の他の組合せも利用され得ることに留意されたい。

【0297】

ステップ603a: ビットストリームを解析し、最適な制御点動きベクトル予測子(すなわち、最適な複数タブルの候補)を決定する。

20

【0298】

B1: 現在の復号ブロックのために使用されるアフィン動きモデルが4パラメータのアフィン動きモデルである場合(MotionModelIdc=1)、インデックス番号がビットストリームから解析され、2つの制御点の最適な動きベクトル予測子が、インデックス番号に基づいて動きベクトル候補リストから決定される。

【0299】

たとえば、インデックス番号はmvp_10_flagまたはmvp_11_flagである。

【0300】

B2: 現在の復号ブロックのために使用されるアフィン動きモデルが6パラメータのアフィン動きモデルである場合(MotionModelIdc=2)、インデックス番号がビットストリームから解析され、3つの制御点の最適な動きベクトル予測子が、インデックス番号に基づいて動きベクトル候補リストから決定される。

30

【0301】

ステップ604a: ビットストリームを解析して、制御点動きベクトルを決定する。

【0302】

C1: 現在の復号ブロックのために使用されるアフィン動きモデルが4パラメータのアフィン動きモデルである場合(MotionModelIdc=1)、現在のブロックの2つの制御点の動きベクトル差分はそれぞれ、ビットストリームを復号することによって取得される。次いで、2つの制御点の動きベクトル値は、制御点の動きベクトル差分および対応する動きベクトル予測子に基づいて取得される。例として前方予測を使用すると、2つの制御点の動きベクトル差分はそれぞれ、mvd_coding(x0,y0,0,0)およびmvd_coding(x0,y0,0,1)である。

40

【0303】

たとえば、左上の制御点および右上の制御点の動きベクトル差分は、ビットストリームを復号することによって取得され、現在のブロックの左上の制御点および右上の制御点の動きベクトルを取得するために、それぞれの動きベクトル予測子に加算される。

【0304】

C2: 現在の復号ブロックのために使用されるアフィン動きモデルは、6パラメータのアフィン動きモデルである(MotionModelIdc=2)。

【0305】

現在のブロックの3つの制御点の動きベクトル差分は、ビットストリームを復号するこ

50

とによって取得される。これらの制御点の動きベクトル値は、制御点の動きベクトル差分およびそれぞれの動きベクトル予測子に基づいて取得される。例として前方予測(すなわち、リスト0)を使用すると、3つの制御点の動きベクトル差分はそれぞれ、`mvd_coding(x0,y0,0,0)`、`mvd_coding(x0,y0,0,1)`、および`mvd_coding(x0,y0,0,2)`である。

【0306】

たとえば、左上の制御点、右上の制御点、および左下の制御点の動きベクトル差分は、ビットストリームを復号することによって取得される。これらの動きベクトル差分は、現在のブロックの左上の制御点、右上の制御点、および左下の制御点の動きベクトルを取得するために、それぞれの動きベクトル予測子に加算される。

【0307】

ステップ605a: 現在の復号ブロックのために使用される制御点動き情報およびアフィン動きモデルに基づいて、現在のブロックの中の各サブブロックの動きベクトルを取得する。

【0308】

現在のアフィン復号ブロックの中のサブブロックは、1つの動き補償ユニットと等価であることがあり、サブブロックの幅および高さは、現在のブロックの幅および高さ未満である。サブブロックまたは動き補償ユニットの中のあらかじめ設定された位置におけるサンプルの動き情報は、サブブロックまたは動き補償ユニットの中のすべてのサンプルの動き情報を表現するために使用され得る。動き補償ユニットのサイズが $M \times N$ であることを仮定すると、あらかじめ設定された位置におけるサンプルは、中心のサンプル($M/2, N/2$)、左上のサンプル($0, 0$)、右上のサンプル($M-1, 0$)、または動き補償ユニットの中の別の位置におけるサンプルであり得る。以下の説明は、説明のための例として、動き補償ユニットの中心のサンプルを使用する。図9Cを参照すると、 V_0 は左上の制御点の動きベクトルを表し、 V_1 は右上の制御点の動きベクトルを表す。各々の小さいボックスは1つの動き補償ユニットを表す。

【0309】

現在のアフィン復号ブロックの中の左上サンプルに対する、相対的な動き補償ユニットの中心のサンプルの座標は、式(25)に従って計算される。式(25)において、 i は水平方向(左から右)における i 番目の動き補償ユニットを示し、 j は垂直方向(上から下)における j 番目の動き補償ユニットを示し、 $(x(i,j), y(i,j))$ は現在のアフィン復号ブロックの中の左上の制御点サンプルに対する相対的な、 (i,j) 番目の動き補償ユニットの中心サンプルの座標を表す。

【0310】

現在のアフィン復号ブロックのために使用されるアフィン動きモデルが6パラメータのアフィン動きモデルである場合、各動き補償ユニット($vx(i,j), vy(i,j)$)の中心のサンプルの動きベクトルを取得するために、 $(x(i,j), y(i,j))$ は6パラメータのアフィン動きモデルの式(26)へと置換される。上で論じられたように、動き補償ユニットの中心のピクセルの動きベクトルは、動き補償ユニットの中のすべてのサンプルの動きベクトルとして使用される。

【0311】

現在のアフィン復号ブロックのために使用されるアフィン動きモデルが4アフィン動きモデルである場合、動き補償ユニットの中のすべてのサンプルの動きベクトルとして使用される各々の動き補償ユニット($vx(i,j), vy(i,j)$)の中心のサンプルの動きベクトルを取得するために、 $(x(i,j), y(i,j))$ は4パラメータのアフィン動きモデルの式(27)へと置換される。

【0312】

【数17】

10

20

30

40

50

$$\begin{cases} x_{(i,j)} = M \times i + \frac{M}{2}, i = 0,1.. \\ y_{(i,j)} = N \times j + \frac{N}{2}, j = 0,1.. \end{cases} \quad (25)$$

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x + \frac{vx_2 - vy_0}{H} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vy_2 - vx_0}{H} y + vy_0 \end{cases} \quad (26)$$

$$\begin{cases} vx = \frac{vx_1 - vx_0}{W} x - \frac{vy_1 - vy_0}{W} y + vx_0 \\ vy = \frac{vy_1 - vy_0}{W} x + \frac{vx_1 - vx_0}{W} y + vy_0 \end{cases} \quad (27)$$

10

【 0 3 1 3 】

ステップ606a: サブブロックの予測サンプル値を取得するために、サブブロックの決定された動きベクトルに基づいて、各サブブロックのための動き補償を行う。

【 0 3 1 4 】

上で論じられたように、ステップ601において、現在のブロックのインター予測モードがアフィン動きモデルベースのマージ(merge)モードであると決定される場合、ステップ602bが実行される。

20

【 0 3 1 5 】

ステップ602b: アフィン動きモデルベースのマージモードに対応する動き情報候補リストを構築する。

【 0 3 1 6 】

具体的には、アフィン動きモデルベースのマージモードに対応する動き情報候補リストは、継承された制御点動きベクトル予測方法および/または構築された制御点動きベクトル予測方法を使用することによって構築され得る。

【 0 3 1 7 】

任意選択で、動き情報候補リストは、特定の規則に従って剪定されて分類され、特定の量の動き情報を含むように、切り詰められ、またはパディングされ得る。

30

【 0 3 1 8 】

D1: 継承された制御点動きベクトル予測方法を使用することによって動きベクトル候補リストを構築するプロセスが説明される。

【 0 3 1 9 】

現在のブロックの制御点動き情報候補は、継承された制御点動きベクトル予測方法を使用することによって導出され、動き情報候補リストに追加される。

【 0 3 2 0 】

図8Aに示される例では、隣接ブロックが位置するアフィンコーディングされたブロックを見つけて、アフィンコーディングされたブロックの制御点動き情報を取得するために、現在のブロックの周りの隣接位置におけるブロックが、A1 B1 B0 A0 B2の順序に従って走査される。さらに、現在のブロックの制御点動き情報候補は、現在のブロックの動きモデルを使用することによって導出される。

40

【 0 3 2 1 】

動きベクトル候補リストが空である場合、上で取得された制御点動き情報候補が候補リストに追加される。それ以外の場合、制御点動き情報候補と同じ動き情報が動きベクトル候補リストの中に存在するかどうかを確認するために、動きベクトル候補リストの中の動き情報が逐次走査される。制御点動き情報候補と同じ動き情報が動きベクトル候補リストに存在しない場合、制御点動き情報候補が動きベクトル候補リストに追加される。

【 0 3 2 2 】

50

2つの動き情報候補が同じであるかどうかを決定することは、動き情報候補の前方(リスト0)参照フレームおよび後方(リスト1)参照フレームならびに各々の前方動きベクトルおよび後方動きベクトルの水平成分と垂直成分が同じであるかどうかを決定することによって、実行され得る。2つの動き情報候補は、すべてのこれらの要素が異なるときにのみ異なるものと見なされる。

【0323】

動きベクトル候補リストの中の動き情報の量が最大リスト長MaxAffineNumMrgCandに達する場合(MaxAffineNumMrgCandは1、2、3、4、または5などの正の整数であり、5は以下の説明において例として使用される)、候補リストの構築は完了する。それ以外の場合、次の隣接ブロックが走査される。

10

【0324】

D2: 現在のブロックの制御点動き情報候補は、構築された制御点動きベクトル予測方法を使用することによって導出され、動き情報候補リストに追加される。図9Bは、構築された制御点動きベクトル予測方法のフローチャートの例を示す。

【0325】

ステップ601c: 現在のブロックの制御点の動き情報を取得する。このステップは「5.構築された制御点動きベクトル予測方法2」におけるステップ501と同様である。詳細はここでは繰り返されない。

【0326】

ステップ602c: 構築された制御点動き情報を取得するために、制御点の動き情報を組み合わせる。このステップは図8Bのステップ501と同様であり、このステップの詳細はここでは再び説明されない。

20

【0327】

ステップ603c: 構築された制御点動き情報を動きベクトル候補リストに追加する。

【0328】

候補リストの長さが最大リスト長MaxAffineNumMrgCand未満である場合、制御点の動き情報の組合せはあらかじめ設定された順序で走査され、得られた有効な組合せは、制御点動き情報候補として使用される。この場合、動きベクトル候補リストが空である場合、制御点動き情報候補が動きベクトル候補リストに追加される。それ以外の場合、制御点動き情報候補と同じ動き情報が動きベクトル候補リストの中に存在するかどうかを確認するために、動きベクトル候補リストの中の動き情報が逐次走査される。制御点動き情報候補と同じ動き情報が動きベクトル候補リストに存在しない場合、制御点動き情報候補が動きベクトル候補リストに追加される。

30

【0329】

たとえば、あらかじめ設定された順序は次の通りである。Affine(CP1,CP2,CP3) Affine(CP1,CP2,CP4) Affine(CP1,CP3,CP4) Affine(CP2,CP3,CP4) Affine(CP1,CP2) Affine(CP1,CP3) Affine(CP2,CP3) Affine(CP1,CP4) Affine(CP2,CP4) Affine(CP3,CP4) 全体で10個の組合せがある。

【0330】

ある組合せに対応する制御点動き情報が利用不可能である場合、この組合せは利用不可能であると見なされる。組合せが利用可能である場合、その組合せの参照フレームインデックスが決定される。2つの制御点の場合、組合せの参照フレームインデックスとして、最小参照フレームインデックスが選択される。2つより多くの制御点の場合、組合せの参照フレームインデックスとして、存在頻度が最大の参照フレームインデックスが選択される。複数の参照フレームインデックスの存在頻度が同じである場合、最小参照フレームインデックスが参照フレームインデックスとして選択される。制御点動きベクトルがさらにスケーリングされる。スケーリングの後のすべての制御点の動き情報が一貫している場合、組合せは無効である。

40

【0331】

任意選択で、本出願のこの実施形態では、動きベクトル候補リストはパディングされ得

50

る。たとえば、前述の走査プロセスの後で、動きベクトル候補リストの長さが最大リスト長MaxAffineNumMrgCand未満である場合、動きベクトル候補リストは、リスト長がMaxAffineNumMrgCandに等しくなるまでパディングされ得る。

【0332】

ゼロ動きベクトルパディング方法を使用することによって、または、既存のリストの中の既存の動き情報候補を組み合わせる(たとえば、加重平均する)ことによって、パディングが実行され得る。動きベクトル候補リストをパディングするための他の方法は、本出願にも適用可能であることに留意されたい。

【0333】

ステップ603b: ビットストリームを解析して、最適な制御点動き情報を決定する。

10

【0334】

インデックス番号が解析され、最適な制御点動き情報が、インデックス番号に基づいて動きベクトル候補リストから決定される。

【0335】

ステップ604b: 現在の復号ブロックのために使用される最適な制御点動き情報およびアフィン動きモデルに基づいて、現在のブロックの中の各サブブロックの動きベクトルを取得する。

【0336】

このステップはステップ605aと同様である。

【0337】

ステップ605b: サブブロックの予測サンプル値を取得するために、サブブロックの決定された動きベクトルに基づいて、各サブブロックのための動き補償を行う。

20

【0338】

前に説明されたように、各サブブロックの動きベクトルがステップ605aおよび604bにおいて取得された後で、サブブロックのための動き補償が、それぞれステップ606aおよび605bにおいて実行される。すなわち、アフィンコーディングされたブロックの現在のサブブロックの予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックのためのサブブロックベースのアフィン動き補償を行うことの詳細が、上で説明される。従来の設計では、サブブロックのサイズは 4×4 に設定され、すなわち、それぞれの/異なる動きベクトルを使用することによって、各々の 4×4 ユニットに対して動き補償が実行される。一般に、より小さいサブブロックのサイズは、より高い動き補償計算の複雑さおよびより良好な予測効果につながる。動き補償計算の複雑さと予測の正確さの両方を考慮するために、サブブロックレベルの動き補償の後で、オプティカルフローを用いた予測信号洗練(PROP)のためのプロセスが提供される。プロセスの例示的なステップは次の通りである。

30

【0339】

(1)各サブブロックの動きベクトルがステップ605aおよび604bを使用することによって取得された後で、サブブロックの予測信号 $I(i,j)$ を取得するために、ステップ606aおよび605bを使用することによってサブブロックのための動き補償を行う。ステップ(1)はPROPプロセスに含まれないことに留意することができる。

40

【0340】

(2)サブブロックの予測信号の水平勾配値 $g_x(i,j)$ および垂直勾配値 $g_y(i,j)$ を計算し、計算方法は次の通りである。

$$g_x(i,j) = I(i+1,j) - I(i-1,j)$$

$$g_y(i,j) = I(i,j+1) - I(i,j-1)$$

【0341】

図9Dに示されるように、 4×4 ブロックに対する勾配値(4×4 勾配値)を取得するために、 6×6 の予測信号ウィンドウ900が必要とされることが、式からわかり得る。

【0342】

これは、以下の異なる方法を使用することによって実施され得る。

50

a)サブブロックの予測行列がサブブロックの動き情報(たとえば、動きベクトル)に基づいて取得された後で、サブブロックの水平勾配行列および垂直勾配行列を取得する。言い換えると、 $M \times N$ サブブロックの動きベクトルに基づく補間を通じて、 $(M+2) \times (N+2)$ 予測ブロックが取得される。たとえば、 6×6 予測信号を取得し、 4×4 勾配値(すなわち、 4×4 勾配行列)を計算するために、サブブロックの動きベクトルに基づいて、補間が直接実行される。

b) 4×4 予測信号(すなわち、第1の予測行列)を取得するために、サブブロックの動きベクトルに基づいて補間を実行し、次いで、 6×6 予測信号(すなわち、第2の予測行列)を取得して 4×4 勾配値(すなわち、 4×4 勾配行列)を計算するために、予測信号に対してエッジ延長を実行する。

c) 各々の 4×4 予測信号(すなわち、第1の予測行列)を取得し、組合せを通じて $w \times h$ 予測信号を取得するために、各サブブロックの動きベクトルに基づいて補間を実行する。次いで、 $(w+2) \times (h+2)$ 予測信号を取得するために、 $w \times h$ 予測信号に対してエッジ延長を実行し、各々の 4×4 勾配値(すなわち、 4×4 勾配行列)を取得するために、 $w \times h$ 勾配値(すなわち、 $w \times h$ 勾配行列)を計算する。

【 0 3 4 3 】

$M \times N$ サブブロックの動きベクトルに基づく補間を通じて直接 $(M+2) \times (N+2)$ 予測ブロックを取得することは、以下の実装形態を含むことに留意されたい。

【 0 3 4 4 】

a1)周りの領域(図13の白いサンプル)に対して、動きベクトルが指し示す位置の左上サンプルの整数サンプルが取得される。内側の領域(図13の灰色のサンプル)に対して、動きベクトルが指し示す位置のサンプルが取得される。サンプルが分数サンプルである場合、サンプルは、補間フィルタを使用することによって補間を通じて取得される。

【 0 3 4 5 】

図14に示されるように、A、B、C、およびDは整数サンプルであり、 $M \times N$ サブブロックの動きベクトルは $1/16$ サンプル精度であり、 $dx/16$ は左上サンプルの分数サンプルと整数サンプルとの間の水平距離であり、 $dy/16$ は左上サンプルの分数サンプルと整数サンプルとの間の垂直距離である。周りの領域に対して、Aのサンプル値は、サンプル位置の予測サンプル値として使用される。内側の領域に対して、サンプル位置の予測サンプル値は、補間フィルタを使用することによって補間を通じて取得される。

【 0 3 4 6 】

a2)周りの領域(図13の白いサンプル)に対して、動きベクトルが指し示す位置に最も近い整数サンプルが取得される。内側の領域(図13の灰色のサンプル)に対して、動きベクトルが指し示す位置のサンプルが取得される。サンプルが分数サンプルである場合、サンプルは、補間フィルタを使用することによって補間を通じて取得される。

【 0 3 4 7 】

図14に示されるように、周りの領域に対して、動きベクトルが指し示す位置に最も近い整数サンプルは、 dx および dy に基づいて選択される。

【 0 3 4 8 】

a3)周りの領域と内側の領域の両方に対して、動きベクトルが指し示す位置のサンプルが取得される。サンプルが分数サンプルである場合、サンプルは、補間フィルタを使用することによって補間を通じて取得される。

【 0 3 4 9 】

a)、b)、およびc)は3つの異なる実装形態であることが理解されるべきである。

【 0 3 5 0 】

(3)デルタ予測値を計算し、計算方法は次の通りである。

$$I(i,j) = g_x(i,j)^* v_x(i,j) + g_y(i,j)^* v_y(i,j)$$

【 0 3 5 1 】

(i,j) はサブブロックの現在のサンプルを表し、 $v(i,j)$ は、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの差分(図

10

20

30

40

50

10に示されるような)であり、前述の式に従って計算されてもよく、 $v_x(i,j)$ および $v_y(i,j)$ は、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの間の差分の、水平オフセット値および垂直オフセット値である。代替的に、簡略化された方法において、現在のサンプルが属する各々の 2×2 サンプルブロックの動きベクトルと、サブブロックの中心のサンプルの動きベクトルとの間の動きベクトル差分が、計算され得る。比較すると、 $v(i,j)$:動きベクトル差分は、各ピクセルまたはサンプルに対して計算される必要がある(たとえば、 4×4 サブブロックでは計算は16回実行される必要がある)。しかしながら、簡略化された方法では、動きベクトル差分は、各々の 2×2 サブブロックに対して計算される(たとえば、 4×4 サブブロックでは計算は4回実行される)。ここでのサブブロックは、 4×4 サブブロックまたは $M \times N$ サブブロックであり得ることに留意されたい。たとえば、ここでのmは4以上であり、またはここでのnは4以上である。

【0 3 5 2】

【数18】

10

$$\begin{cases} \Delta v_x(x, y) = c * x + d * y \\ \Delta v_y(x, y) = e * x + f * y \end{cases}$$

20

【0 3 5 3】

4パラメータアフィンモデルに対して:

【0 3 5 4】

【数19】

$$\begin{cases} c = f = \frac{v_{1x} - v_{0x}}{w} \\ e = -d = \frac{v_{1y} - v_{0y}}{w} \end{cases}$$

30

【0 3 5 5】

6パラメータアフィンモデルに対して:

【0 3 5 6】

【数20】

$$\begin{cases} c = \frac{v_{1x} - v_{0x}}{w} \\ d = \frac{v_{2x} - v_{0x}}{h} \\ e = \frac{v_{1y} - v_{0y}}{w} \\ f = \frac{v_{2y} - v_{0y}}{h} \end{cases}$$

40

【0 3 5 7】

50

ここで、 (v_{0x}, v_{0y}) 、 (v_{1x}, v_{1y}) 、および (v_{2x}, v_{2y}) は、左上、右上、および左下の制御点の動きベクトルであり、 w および h はアフィンコーディングされたブロック(CU)の幅と高さである。

【0358】

(4)予測洗練化を実行する：

$$I'(i,j) = I(i,j) + \Delta I(i,j)$$

ここで $I(i,j)$ は、サブブロックのサンプル (i,j) の予測値(すなわち、サブブロックの中の位置 (i,j) における予測サンプル値)であり、 $\Delta I(i,j)$ はサブブロックのサンプル (i,j) のデルタ予測値であり、 $I'(i,j)$ はサブブロックのサンプル (i,j) の洗練化された予測サンプル値である。

本開示の実施形態による、オプティカルフローを用いたサブブロックベースのアフィン動き補償された予測を洗練化するために、オプティカルフローを用いた予測洗練化(PROF)プロセスが条件的に実行されるは、次のように説明される。

【0359】

実施形態1

オプティカルフローを用いてサブブロックのデルタ予測値(これは具体的にはサブブロックの各サンプルのデルタ予測値である)を取得するための方法は、単方向のアフィンコーディングされたブロックに適用されてもよく、または双方向のアフィンコーディングされたブロックに適用されてもよい。方法が双方向のアフィン予測ブロックに適用される場合、上で説明されたステップ(1)から(4)は2回実行される必要があり、比較的計算の複雑さが高くなる。方法の複雑さを下げるために、本発明は、PROFを適用するための制約を提供する。具体的には、予測サンプル値は、アフィンコーディングされたブロックが单方向のアフィンコーディングされたブロックであるときにのみ、この方法を使用することによって洗練化される。

【0360】

デコーダ側において、ビットストリームを解析することによって取得されるシンタックス要素は、单予測または双予測を示す。このシンタックス要素は、アフィンコーディングされたブロックが单方向のアフィンコーディングされたブロックであるかどうかを決定するために使用され得る。

【0361】

エンコーダ側において、BフレームおよびPフレームの構造が、異なる使用事例によって決定され、Bフレームにおいて单予測が使用されるか、または双予測が使用されるかが、RDOによって決定される。言い換えると、Bフレームに対して、エンコーダ側は、RDOコストに基づいて、現在のアフィンピクチャブロックのために单予測が使用されるか、または双予測が使用されるかを決定し得る。たとえば、エンコーダ側は、前方予測、後方予測、および双方向予測の中から、RDOを最小にする機構を選択することを試みる。

【0362】

実施形態2

オプティカルフローを用いた予測信号洗練の複雑さを下げるために、オプティカルフローを用いてサブブロックの予測オフセット値を取得するための方法は、サブブロックのサイズが比較的大きいときにのみ使用され得る。たとえば、单方向のアフィンコーディングされたブロックのサブブロックサイズは 4×4 に設定されることがあり、双方向のアフィンコーディングされたブロックのサブブロックサイズは、 8×8 、 8×4 、または 4×8 に設定されることがある。この例では、サブブロックサイズが 4×4 より大きいときにのみ、この方法が使用される。別の例では、サブブロックサイズは、アフィンコーディングされたブロックの制御点の動きベクトル、アフィンコーディングされたブロックの幅および高さなどの情報に基づいて、適応的に選択され得る。サブブロックサイズが 4×4 より大きいときにのみ、この方法が使用される。

【0363】

加えて、ステップ(2)において、方法a)とb)はともに、アフィンコーディングされたブロ

10

20

30

40

50

ックの各々の 4×4 サブブロックの予測に依存関係がなく、同時に実行され得ることを、確実にすることができる。しかしながら、方法a)は補間計算の複雑さを増大させる。方法b)は複雑さを増大させないが、境界の勾配値が延長されたサンプルを使用することによる計算を通じて取得され、正確さは高くない。方法c)は勾配計算の正確さを改善することができるが、各々の 4×4 サブブロックには依存性があり、すなわち、オプティカルフローに基づく洗練化は、CU全体の補間が完了したときにのみ実行され得る。

【0364】

図9Eに示されるように、勾配計算の同時性と正確さの両方を考慮するために、本開示は、 16×16 の粒度に基づく勾配値計算を提案する。 $\text{size_w}=\min(w, 16)$ であり、 $\text{size_h}=\min(h, 16)$ であると仮定すると、アフィンコーディングされたブロックの中の各々の $\text{size_w} * \text{size_h}$ に対して、アフィンコーディングされたブロックの中の各々の 4×4 サブブロックの予測子(予測サンプル値)が計算され、 $\text{size_w} * \text{size_h}$ 予測信号が組合せを通じて取得される。次いで、 $(\text{size_w}+2) * (\text{size_h}+2)$ 予測信号を取得するために、 $\text{size_w} * \text{size_h}$ 予測信号に対してエッジ延長が実行される(たとえば、パディングなどによって、2サンプル分外側に延長される)。各々の 4×4 勾配値を取得するために、 $\text{size_w} * \text{size_h}$ 勾配値が計算される。本出願における外側に延長されるサンプルの量は、2サンプルには限らず、勾配計算に關係することを理解されたい。勾配の分解能が3タップである場合、外側に2サンプル延長される。言い換えると、これは勾配計算のためのフィルタに關係する。フィルタのタップの量がTであると仮定すると、追加される領域またはサポートされる周りの領域は、 $T/2$ (割り切れる) $* 2$ である。

10

【0365】

図11Aは、一実施形態による、アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PROP)のための方法を示し、この方法はコーディング装置(復号装置またはデコーダなど)によって実行され得る。この方法は以下のステップを含む。

20

【0366】

S1101. 複数のオプティカルフロー決定条件が満たされていると決定する。

【0367】

ここで、オプティカルフロー決定条件は、PROPの適用を許容するための条件とも呼ばれ得る。オプティカルフロー決定条件のすべてが満たされている場合、PROPは、アフィンコーディングされたブロックの現在のサブブロックのために適用される。オプティカルフロー決定条件の例が以下で説明される。いくつかの例では、オプティカルフロー決定条件は、PROPを適用するための制約条件と置き換えられてもよく、またはそのように言い換えられてもよい。PROPを適用するための制約条件が満たされている場合、PROPは、アフィンコーディングされたブロックの現在のサブブロックに適用されない。それらの例では、ステップS1101は、PROPを適用するための複数の制約条件のいずれもが満たされていないと決定することに変更される。

30

【0368】

S1102. アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するPROPプロセスを行い、複数のオプティカルフロー決定条件が、アフィンコーディングされたブロックに対してすべて満たされている。ここで、現在のサブブロックの洗練化された予測サンプル値は、予測洗練化が加えられた後の現在のサブブロックの最終的な予測サンプル値として理解され得る。

40

【0369】

ステップS1102において、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、現在のサブブロックまたは各サブブロック)のデルタ予測値(たとえば、 $I(i,j)$)を取得するために、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、現在のサブブロックまたは各サブブロック)に対するオプティカルフロー(オプティカルフローを用いた予測洗練化、PROP)処理を実行する。

【0370】

50

ステップS1102は、サブプロックのデルタ予測値(たとえば、 $\Delta I(i,j)$)およびサブプロックの予測サンプル値(たとえば、予測信号 $I(i,j)$)に基づいて、サブプロックの洗練化された予測サンプル値(たとえば、予測信号 $I'(i,j)$)を取得するステップを伴う。

【0371】

具体的には、ステップS1102は、サブプロックの中の現在のサンプルのデルタ予測値(たとえば、 $\Delta I(i,j)$)およびサブプロックの中の現在のサンプルの予測子(たとえば、予測信号 $I(i,j)$)に基づいて、サブプロックの中の現在のサンプルの洗練化された予測子(たとえば、予測信号 $I'(i,j)$)を取得するステップを伴う。

【0372】

ある可能な設計では、複数のオプティカルフロー決定条件は、以下のうちの1つまたは複数を含む。

【0373】

(a) 解析または導出を通じて取得される指示情報(たとえば、`sps_prof_enabled_flag`または`sps_bdoe_enabled_flag`)は、PROFが現在のシーケンス、ピクチャ、スライス、またはタイルグループに対して有効であることを示す。たとえば、`sps_prof_enabled_flag`または`sps_bdoe_enabled_flag=1`である。PROFを適用するための制約条件がオプティカルフロー決定条件の代わりにS1101において使用される場合、この条件は、次のようにPROFを適用するための制約条件に変換され得ることが理解され得る。(a) PROFが現在のシーケンス、ピクチャ、スライス、またはタイルグループに対して無効であることを指示情報が示す。たとえば、`sps_prof_disabled_flag`または`sps_bdoe_disabled_flag=1`である。

【0374】

SPS、PPS、スライスヘッダ、またはタイルグループヘッダなどのパラメータセットを解析することによって取得される指示情報は、現在のシーケンス、ピクチャ、スライス、またはタイルグループに対してPROFが有効であるかどうかを示す。

【0375】

具体的には、`sps_prof_enabled_flag`が制御のために使用されてもよく、`sps_prof_enabled_flag`のシンタックスおよびセマンティクスは次の通りである。

【0376】

【表4】

10

20

30

40

<code>seq_parameter_set_rbsp()</code>	記述子
...	
<code>sps_affine_enabled_flag</code>	<code>u(1)</code>
<code>if(sps_affine_enabled_flag) {</code>	
<code>sps_affine_type_flag</code>	<code>u(1)</code>
<code>sps_prof_enabled_flag</code>	<code>u(1)</code>
}	
...	

【0377】

0に等しい`sps_prof_enabled_flag`は、アフィンベースの動き補償のための予測洗練化オプティカルフローが無効であることを指定する。1に等しい`sps_prof_enabled_flag`は、アフィンベースの動き補償のための予測洗練化オプティカルフローが有効であることを指定する。

50

【0378】

代替的に、`sps_bdot_enabled_flag`が制御のために再使用される。

【0379】

この実施形態では、前述の条件が満たされている(たとえば、メインスイッチがPROFを有効にすると決定する)という前提で、別の条件がさらに導出されることを理解されたい。言い換えると、PROFが現在のシーケンス、ピクチャ、スライス、またはタイルグループに対して有効である場合、現在のアフィンピクチャブロックが以下で説明される別のオプティカルフロー決定条件を満たすかどうかがさらに決定される。PROFが現在のシーケンス、ピクチャ、スライス、またはタイルグループに対して有効ではない場合、現在のアフィンピクチャブロックが以下で提供される別のオプティカルフロー決定条件を満たすかどうかを決定することは不要である。

10

【0380】

(b)現在のアフィンコーディングされたブロックが区分されるべきであることを導出された指示情報(たとえば、変数`fallbackModeTriggered`)が示す。たとえば、`fallbackModeTriggered=0`である。PROFを適用するための制約条件がオプティカルフロー決定条件の代わりにS1101において使用される場合、条件(b)は、次のようにPROFを適用するための制約条件に変換され得ることが理解され得る。(b)現在のアフィンコーディングされたブロックが区分されないことを導出された指示情報が示す。たとえば、`fallbackModeTriggered=1`である。

20

【0381】

変数`fallbackModeTriggered`はアフィンパラメータに基づいて導出され、PROFを使用するかどうかは`fallbackModeTriggered`に依存する。`fallbackModeTriggered`が1であるとき、それは、現在のアフィンコーディングされたブロックが区分されるべきではないことを示す。`fallbackModeTriggered`が0であるとき、それは、アフィンコーディングされたブロックが区分されるべきである(たとえば、アフィンコーディングされたブロックがサブブロック、たとえば 4×4 サブブロックへと区分される)ことを示す。現在のアフィンコーディングされたブロックが区分されるべきであるとき、PROFが使用されることになる。

【0382】

具体的には、変数`fallbackModeTriggered`は次のプロセスを使用することによって導出され得る。

30

【0383】

変数`fallbackModeTriggered`は最初に1に設定され、次のようにさらに導出される。

- 変数`bxWX4`、`bxHX4`、`bxWXh`、`bxHXh`、`bxWXv`および`bxHXv`が次のように導出される。

`maxW4=Max(0,Max(4*(2048+dHorX),Max(4*dHorY,4*(2048+dHorX)+4*dHorY)))` (8-775)

`minW4=Min(0,Min(4*(2048+dHorX),Min(4*dHorY,4*(2048+dHorX)+4*dHorY)))` (8-775)

`maxH4=Max(0,Max(4*dVerX,Max(4*(2048+dVerY),4*dVerX+4*(2048+dVerY))))` (8-775)

`minH4=Min(0,Min(4*dVerX,Min(4*(2048+dVerY),4*dVerX+4*(2048+dVerY))))` (8-775)

`bxWX4=((maxW4-minW4) 11)+9` (8-775)

`bxHX4=((maxH4-minH4) 11)+9` (8-775)

`bxWXh=((Max(0,4*(2048+dHorX))-Min(0,4*(2048+dHorX))) 11)+9` (8-775)

`bxHXh=((Max(0,4*dVerX)-Min(0,4*dVerX)) 11)+9` (8-775)

`bxWXv=((Max(0,4*dVerY)-Min(0,4*dVerY)) 11)+9` (8-775)

`bxHXv=((Max(0,4*(2048+dHorY))-Min(0,4*(2048+dHorY))) 11)+9` (8-775)

- `inter_pred_idc[xCb][yCb]`が`PRED_BI`に等しく、`bxWX4*bxHX4`が225以下である

40

50

場合、`fallbackModeTriggered`は0に等しく設定される。

- そうではなく、 $b_{\text{X}}W_{\text{X}_h} * b_{\text{X}}H_{\text{X}_h}$ が165以下であり、かつ $b_{\text{X}}W_{\text{X}_v} * b_{\text{X}}H_{\text{X}_v}$ が165以下である場合、`fallbackModeTriggered`は0に等しく設定される。

【0384】

(c)現在のアフィンピクチャブロックは、単予測アフィンピクチャブロックである。

【0385】

- (d)アフィンピクチャブロックの中のサブブロックのサイズは $N \times N$ より大きく、 $N=4$ である。

【0386】

- (e)現在のアフィンピクチャブロックは単予測アフィンピクチャブロックであり、アフィンピクチャブロックの中のサブブロックのサイズは $N \times N$ に等しく、 $N=4$ である。

10

【0387】

- (f)現在のアフィンピクチャブロックは双予測アフィンピクチャブロックであり、アフィンピクチャブロックの中のサブブロックのサイズは $N \times N$ より大きく、 $N=4$ である。

11

【0388】

現在のアフィンピクチャブロックは現在のアフィン符号化されたブロックであり、現在のアフィンピクチャブロックが単予測アフィンピクチャブロックであることは、以下の方法を使用することによって決定される。

エンコーダ側において、レートひずみ基準RDOに基づいて、単予測が現在のアフィンピクチャブロックのために使用されることが決定される。

20

【0389】

現在のアフィンピクチャブロックは現在のアフィン復号ブロックであり、現在のアフィンピクチャブロックが単予測アフィンピクチャブロックであることは、以下の方法を使用することによって決定される。

デコーダ側では、AMVPモードにおいて、単予測方向(たとえば、前方予測のみまたは後方予測のみ)を示すために予測方向指示情報が使用され、ビットストリームを解析することによって、もしくは導出を通じて、予測方向指示情報が取得され、または、

デコーダ側では、マージモードにおいて、候補リストの中の候補インデックスに対応する動き情報候補が第1の参照フレームリストに対応する第1の動き情報を含み、または、候補リストの中の候補インデックスに対応する動き情報候補が第2の参照フレームリストに対応する第2の動き情報を含む。

30

【0390】

ある可能な設計において、予測方向指示情報はシンタックス要素`inter_pred_idc[x0][y0]`を含み、

`inter_pred_idc[x0][y0]=PRED_L0`であり、これは前方予測を示すために使用され、`inter_pred_idc[x0][y0]=PRED_L1`であり、これは後方予測を示すために使用され、または、

予測方向指示情報は`predFlagL0`および/または`predFlagL1`を含み、

`predFlagL0=1`、`predFlagL1=0`であり、これは前方予測を示すために使用され、`predFlagL1=1`、`predFlagL0=0`であり、これは後方予測を示すために使用される。

40

【0391】

オプティカルフロー決定条件(またはPROFを適用するための制約条件)は上の例には限らず、追加のまたは異なるオプティカルフロー決定条件(またはPROFを適用するための制約条件)は、異なる適用シナリオに基づいて設定されてもよいことに留意されたい。たとえば、上の条件(c)から(f)は、オプティカルフロー決定条件などの他の条件で置き換えられてもよい。PROFは、アフィンコーディングされたブロックのすべての制御点MVが互いに異なる場合、またはPROFを適用するための制約条件と異なる場合、アフィンコーディングされたブロックに対して適用され得る。PROFは、アフィンコーディングされたブロックのすべての制御点MVが同じである場合、アフィンコーディングされたブロックに対して適用されない。オプティカルフロー決定条件など:PROFは、現在のピクチャの解像度

50

およびアフィンコーディングされたブロックの参照ピクチャの解像度が同じである場合、たとえば、RprConstraintsActive[X][refIdxLX]が0またはPROFを適用するための制約条件に等しい場合、アフィンコーディングされたブロックに対して適用され得る。PROFは、現在のピクチャの解像度およびアフィンコーディングされたブロックの参照ピクチャの解像度が互いに異なる場合、たとえば、RprConstraintsActive[X][refIdxLX]が1に等しい場合、アフィンコーディングされたブロックに対して適用されない。

【0392】

ある可能な設計では、ステップS1102において、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロックまたは現在のサブブロック)のデルタ予測値(たとえば、 $I(i,j)$)を取得するために、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロックまたは現在のサブブロック)に対するオプティカルフロー(オプティカルフローを用いた予測洗練化、PROF)処理を実行することは、以下のステップを含み得る。

10

【0393】

ステップ1. 現在のアフィンピクチャブロックの中の現在のサブブロックの動き情報(たとえば、動きベクトル)に基づいて、第2の予測行列を取得する。

【0394】

たとえば、 $M \times N$ サブブロックの動きベクトルに基づく補間を通じて、 $(M+2) \times (N+2)$ 予測ブロック(すなわち、第2の予測行列)が取得される。様々な実装形態が上で与えられる。

20

【0395】

ステップ2. 第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を計算し、第2の予測行列のサイズは水平予測勾配行列および垂直予測勾配行列のサイズ以上である。

【0396】

ステップ3. 水平予測勾配行列の中のサブブロックの中の現在のサンプルの水平予測勾配値、垂直予測勾配行列の中の現在のサンプルの垂直予測勾配値、および現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの差分に基づいて、サブブロックの中の現在のサンプルのデルタ予測値($I(i,j)$)を計算する。
。

【0397】

30

それに対応して、ステップS1102において、サブブロックのデルタ予測値(たとえば、 $I(i,j)$)およびサブブロックの予測サンプル値(たとえば、予測信号 $I(i,j)$)に基づいて、サブブロックの洗練化された予測サンプル値(たとえば、予測信号 $I'(i,j)$)を取得することは、サブブロックの中の現在のサンプルのデルタ予測値(たとえば、 $I(i,j)$)および現在のサンプルの予測サンプル値(たとえば、予測信号 $I(i,j)$)に基づいて、現在のサンプルの洗練化された予測サンプル値(たとえば、予測信号 $I'(i,j)$)を取得するステップを含み得る。

【0398】

サブブロックの予測サンプル値(たとえば、予測信号 $I(i,j)$)は、 $(M+2) \times (N+2)$ 予測ブロックの中の $M \times N$ 予測ブロックであり得ることが理解されるべきである。

【0399】

40

ステップ3について、ある実装形態では、現在のサブブロックの中の異なるサンプルの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分は異なる。別の実装形態では、現在のサンプルを含む現在のサンプルユニット(たとえば、 2×2 サンプルブロック)の動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分が、現在のサブブロックの現在のサンプルの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分として使用される。言い換えると、処理オーバーヘッドと予測の正確さのバランスを取るために、サンプルAとBの両方が現在のサンプルユニットに含まれると仮定して、現在のサンプルユニット(たとえば、 2×2 サンプルブロック)の動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分が、サブブロックの中のサンプルAの動きベクトルとサブブロックの中

50

心のサンプルの動きベクトルとの動きベクトル差分として使用され得る。また、現在のサンプルユニットの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分が、サブブロックの中のサンプルBの動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分として使用され得る。

【0400】

ある実装形態では、上記のステップ1における第2の予測行列は $I_1(p,q)$ によって表され、pの値の範囲は $[-1, sbW]$ であり、qの値の範囲は $[-1, sbH]$ であり、

水平予測勾配行列は $X(i,j)$ によって表され、iの値の範囲は $[0, sbW-1]$ であり、jの値の範囲は $[0, sbH-1]$ であり、

垂直予測勾配行列は $Y(i,j)$ によって表され、iの値の範囲は $[0, sbW-1]$ であり、jの値の範囲は $[0, sbH-1]$ であり、

sbW は現在のアフィンピクチャブロックの中の現在のサブブロックの幅を表し、 sbH は現在のアフィンピクチャブロックの中の現在のサブブロックの高さを表し、 (x,y) は現在のアフィンピクチャブロックの中の現在のサブブロックの中の各サンプル(サンプルとも呼ばれる)の位置座標を表し、 (x,y) に位置する要素は (i,j) に位置する要素に対応し得る。

【0401】

別の可能な設計では、ステップ1102において、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロック)のデルタ予測値(予測子オフセット値、たとえば $I(i,j)$ とも呼ばれる)を取得するために、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロック)に対するオプティカルフロー(オプティカルフローを用いた予測洗練化、PROF)処理を実行することは、図12に示されるように、以下を含む。

S1202. 第1の予測行列に基づいて第2の予測行列を取得または生成し、サブブロック(たとえば、各サブブロック)の第1の予測行列(たとえば、第1の予測信号 $I(i,j)$ または 4×4 予測)は現在のサブブロックの予測サンプル値に対応する。図9Aに示されるように、アフィンコーディングされたブロックの現在のサブブロックに対するサブブロックベースのアフィン動き補償が、アフィンコーディングされたブロックの現在のサブブロックの予測サンプル値を取得するために実行される。

S1203. 第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を計算し、第2の予測行列のサイズは第1の予測行列のサイズ以上であり、第2の予測行列のサイズは水平予測勾配行列および垂直予測勾配行列のサイズ以上である。

【0402】

S1204. 水平予測勾配行列、垂直予測勾配行列、およびサブブロックの現在のサンプルユニット(たとえば、 2×2 サンプルブロックなどの、現在のサンプルまたは現在のサンプルブロック)の動きベクトルとサブブロックの中心のサンプルの動きベクトルとの動きベクトル差分に基づいて、サブブロックのデルタ予測値行列(たとえば、予測信号の $I(i,j)$)を計算する。

サブブロックのデルタ予測値(たとえば、 $I(i,j)$)およびサブブロックの予測サンプル値(たとえば、予測信号 $I(i,j)$)に基づいて、サブブロックの洗練化された予測サンプル値(たとえば、予測信号 $I'(i,j)$)を取得するステップは以下を含む。

S1205. デルタ予測値行列(たとえば、 $I(i,j)$)および第1の予測行列(たとえば、予測信号 $I(i,j)$)に基づいてサブブロックの洗練化された第3の予測行列(たとえば、予測信号 $I'(i,j)$)を取得する。

【0403】

本明細書の $I(i,j)$ は現在のサブブロックの中の現在のサンプルの予測サンプル値(たとえば、動き補償を通じて取得された元の予測)を表し、 $I'(i,j)$ は現在のサブブロックの中の現在のサンプルのデルタ予測値を表し、 $I'(i,j)$ は現在のサブブロックの中の現在のサンプルの洗練化された予測サンプル値を表すことを理解されたい。たとえば、元の予測サンプル値+デルタ予測値=洗練化された予測サンプル値である。現在のサブブロックの中の複数のサンプル(たとえば、すべてのサンプル)の洗練化された予測サンプル値を取得すること

10

20

30

40

50

は、現在のサブブロックの洗練化された予測サンプル値を取得することと等価であることを理解されたい。

【0404】

異なる可能な実装形態では、勾配値はサンプルごとに計算されてもよく、デルタ予測値はサンプルごとに計算されてもよい。代替的に、勾配行列が取得されてもよく、次いでデルタ予測値が計算されてもよい。これは本出願では限定されない。ある代替的な実装形態では、第1の予測行列および第2の予測行列は同じ予測行列を表す。

【0405】

第2の予測行列のサイズが第1の予測行列のサイズに等しく、第2の予測行列のサイズが水平予測勾配行列および垂直予測勾配行列のサイズに等しい場合、ある可能な実装形態では、 $(w-2) \times (h-2)$ 勾配行列が $w \times h$ 予測行列を使用することによって計算され、その勾配行列は $w \times h$ のサイズを得るためにパディングされ、 $w \times h$ は現在のサブブロックのサイズを表す。たとえば、第1の予測行列と第2の予測行列の両方が、サイズが $w \times h$ である予測行列であり、または、第1の予測行列および第2の予測行列は同じ予測行列を表す。

10

【0406】

図11Bに示されるように、本出願の別の実施形態は、以下のステップを含む、アフィンコーディングされたプロックに対するオプティカルフローを用いた予測洗練化(PROP)のための別の方針を提供する。

【0407】

S1110. 複数のオプティカルフロー決定条件が満たされるかどうか、または満足されるかどうかが決定される。ここで、オプティカルフロー決定条件は、PROPの適用を許容するための条件を指す。

20

【0408】

S1111. 複数のオプティカルフロー決定条件が満たされている場合、第1のインジケータ(たとえば、applyProfFlag)は真に等しく設定され、アフィンコーディングされたプロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、アフィンコーディングされたプロックの現在のサブブロックに対するオプティカルフローを用いた予測洗練化(PROP)プロセスを行う。ステップS1111において、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロック)のデルタ予測値(予測子オフセット値、たとえば、 $I(i,j)$ とも呼ばれる)を取得するために、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロック)に対して、オプティカルフロー(オプティカルフローを用いた予測洗練化、PROP)処理が実行される。

30

【0409】

ステップS1111において、サブブロックのデルタ予測値(たとえば、 $I(i,j)$)およびサブブロックの予測サンプル値(たとえば、予測信号 $I(i,j)$)に基づいて、サブブロックの洗練化された予測サンプル値(たとえば、予測信号 $I'(i,j)$)が取得される。

【0410】

本明細書の $I(i,j)$ は現在のサブブロックの中の現在のサンプルの予測サンプル値(たとえば、動き補償を通じて取得された元の予測サンプル値)を表し、 $I'(i,j)$ は現在のサブブロックの中の現在のサンプルのデルタ予測値を表し、 $I'(i,j)$ は現在のサブブロックの中の現在のサンプルの洗練化された予測サンプル値を表すことを理解されたい。たとえば、元の予測サンプル値+デルタ予測値=洗練化された予測サンプル値である。現在のサブブロックの中の複数のサンプル(たとえば、すべてのサンプル)の洗練化された予測サンプル値を取得することは、現在のサブブロックの洗練化された予測サンプル値を取得することと等価であることを理解されたい。

40

【0411】

アフィンコーディングされたプロックの各サブブロックの洗練化された予測サンプル値が生成されるとき、アフィンコーディングされたプロックの洗練化された予測サンプル値は自然に生成されることが理解され得る。S1113、複数のオプティカルフロー決定条件の

50

うちの少なくとも1つが満たされていない、または満足されないとき、第1のインジケータ(たとえば、applyProfFlag)が偽に等しく設定され、PROFプロセスがスキップされる。

【0412】

PROFを適用するための制約条件が、PROFを適用するかどうかを決定するために使用される場合、ステップS1110は、PROFを適用するための複数の制約条件のいずれもが満たされていないかどうかを決定することに変更されることが理解され得る。この場合、ステップS1111は、PROFを適用するための複数の制約条件のいずれもが満たされていない、または満足されない場合、第1のインジケータ(たとえば、applyProfFlag)は真に等しく設定され、アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するオプティカルフローを用いた予測洗練化(PROF)プロセスを行うことに変更される。したがって、ステップS1113は、PROFを適用するための複数の制約条件のうちの少なくとも1つが満たされている場合、第1のインジケータ(たとえば、applyProfFlag)が偽に等しく設定され、PROFプロセスがスキップされることに変更される。

10

【0413】

ある実装形態では、第1のインジケータ(たとえば、applyProfFlag)が第1の値(たとえば、1)である場合、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロック)に対するオプティカルフロー(たとえば、PROF)処理を実行し、または、

そうではなく、第1のインジケータ(たとえば、applyProfFlag)が第2の値(たとえば、0)である場合、現在のアフィンピクチャブロックの中の1つまたは複数のサブブロック(たとえば、各サブブロック)に対するオプティカルフロー(たとえば、PROF)処理の実行をスキップする。

20

【0414】

ある実装形態では、第1のインジケータの値は、オプティカルフロー決定条件が満たされているかどうかに依存し、オプティカルフロー決定条件は以下のうちの1つまたは複数を含む。

PROFが現在のピクチャユニットに対して有効であることを示すために、第1の指示情報(たとえば、sps_prof_enabled_flagまたはsps_bdof_enabled_flag)が使用される。本明細書において現在のピクチャユニットは、たとえば、現在のシーケンス、現在のピクチャ、現在のスライス、または現在のタイルグループであり得ることに留意されたい。現在のピクチャユニットのこれらの例は限定するものではない。

30

現在のアフィンピクチャブロックを区分することを示すために、第2の指示情報(たとえば、fallbackModeTriggered)が使用される。

現在のアフィンピクチャブロックは、単予測アフィンピクチャブロックである。

アフィンピクチャブロックの中のサブブロックのサイズが $N \times N$ より大きく、 $N=4$ である。

現在のアフィンピクチャブロックが单予測アフィンピクチャブロックであり、アフィンピクチャブロックの中のサブブロックのサイズが $N \times N$ に等しく、 $N=4$ である。または、

現在のアフィンピクチャブロックが双予測アフィンピクチャブロックであり、アフィンピクチャブロックの中のサブブロックのサイズが $N \times N$ より大きく、 $N=4$ である。

40

【0415】

本出願は、限定はされないが、前述のオプティカルフロー決定条件を含み、追加のまたは異なるオプティカルフロー決定条件が、異なる適用シナリオに基づいて設定され得ることに留意されたい。

【0416】

本出願のこの実施形態では、たとえば、すべての以下の条件が満たされているとき、applyProfFlagは1に設定される。

- ssp_prof_enabled_flag==1
- fallbackModeTriggered==0

50

- `inter_pred_idc[x0][y0]=PRED_L0` または `PRED_L1` (または `predFlagL0=1`、`predFlagL1=0`; または `predFlagL1=1`、`predFlagL0=0`)

- 他の条件

【 0 4 1 7 】

別の実施形態では、たとえば、すべての以下の条件が満たされているとき、`applyProfFlag`は1に設定される。

- `sps_prof_enabled_flag==1`
- `fallbackModeTriggered==0`
- 他の条件

【 0 4 1 8 】

`PROF`を適用するための制約条件がオプティカルフロー決定条件の代わりに`S1110`において使用される場合、すべての以下の制約条件のいずれもが満たされていないとき、`applyProfFlag`は1に設定されることが理解され得る。

- `sps_prof_disabled_flag==1`
- `fallbackModeTriggered==1`
- 他の条件

【 0 4 1 9 】

本出願のこの実施形態において提供される予測方法におけるステップの実行エンティティ、ならびにこれらのステップの拡張および変形の詳細については、対応する方法の前述の説明を参照することを理解されたい。簡潔にするために、本明細書では詳細は再び説明されない。

【 0 4 2 0 】

本出願の別の実施形態はさらに、

現在のアフィンピクチャブロックの中の複数のサブブロックの動き情報(たとえば、動きベクトル)に基づいて $M \times N$ ブロックの第1の予測行列を取得するステップであって、たとえば、 $M \times N$ ブロックは図9Eに示されるように 16×16 であり、たとえば、 16×16 ブロック(または 16×16 ウィンドウ)は16個の 4×4 サブブロックを含む、ステップと、

第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を計算するステップであって、第2の予測行列のサイズは第1の予測行列のサイズ以上であり、第2の予測行列のサイズは水平予測勾配行列および垂直予測勾配行列のサイズ以上である、ステップと、

水平予測勾配行列、垂直予測勾配行列、および $M \times N$ ブロックの中の現在のピクセルユニット(たとえば、 2×2 ピクセルブロックなどの、現在のピクセルまたは現在のピクセルブロック)の動きベクトルと $M \times N$ ブロックの中心のピクセルの動きベクトルとの動きベクトル差分に基づいて、 $M \times N$ ブロックのデルタ予測値行列(たとえば、予測信号の $I(i,j)$)を計算するステップと、

デルタ予測値行列(たとえば、 $I(i,j)$)および第1の予測行列(たとえば、予測信号 $I(i,j)$)に基づいて $M \times N$ ブロックの洗練化された第3の予測行列(たとえば、予測信号 $I'(i,j)$)を取得するステップとを含む、別の`PROF`プロセスを提供する。

【 0 4 2 1 】

別の可能な設計では、第1の予測行列は $I_1(i,j)$ によって表され、 i の値の範囲は $[0, size_w-1]$ であり、 j の値の範囲は $[0, size_h-1]$ であり、

第2の予測行列は $I_2(i,j)$ によって表され、 i の値の範囲は $[-1, size_w]$ であり、 j の値の範囲は $[-1, size_h]$ であり、 $size_w=\min(W,m)$ 、 $size_h=\min(H,m)$ 、および $m=16$ であり、

水平予測勾配行列は $X(i,j)$ によって表され、 i の値の範囲は $[0, size_w-1]$ であり、 j の値の範囲は $[0, size_h-1]$ であり、

垂直予測勾配行列は $Y(i,j)$ によって表され、 i の値の範囲は $[0, size_w-1]$ であり、 j の値の範囲は $[0, size_h-1]$ であり、

W は現在のアフィンピクチャブロックの幅を表し、 H は現在のアフィンピクチャブロックの高さを表し、 (x,y) は現在のアフィンピクチャブロックの中の各サンプルの位置座標を表す。

10

20

30

40

50

【 0 4 2 2 】

図9Eに示されるように、いくつかの例では、アフィンピクチャブロックが 16×16 ブロックへと暗黙的に区分され、勾配行列が各々の 16×16 ブロックのために計算されることを理解されたい。それに対応して、第2の予測行列は $I_2(i,j)$ によって表され、 i の値の範囲は $[-1, size_w]$ であり、 j の値の範囲は $[-1, size_h]$ であり、 $size_w = \min(w, m)$ 、 $size_h = \min(h, m)$ 、および $m = 16$ である。水平予測勾配行列は $X(i,j)$ によって表され、 i の値の範囲は $[0, size_w - 1]$ であり、 j の値の範囲は $[0, size_h - 1]$ である。垂直予測勾配行列は $Y(i,j)$ によって表され、 i の値の範囲は $[0, size_w - 1]$ であり、 j の値の範囲は $[0, size_h - 1]$ である。

【 0 4 2 3 】

別の可能な設計では、第2の予測行列(たとえば、 $(size_w + 2) * (size_h + 2)$ 予測信号)に基づいて水平予測勾配行列および垂直予測勾配行列(たとえば、 $size_w * size_h$ 勾配値)を計算することは、10

第2の予測行列に基づいて水平予測勾配行列および垂直予測勾配行列を計算することを含み、水平予測勾配行列および垂直予測勾配行列は、それぞれ、サブブロックの水平予測勾配行列および垂直予測勾配行列を含み、

第2の予測行列は $I_2(i,j)$ によって表され、 i の値の範囲は $[-1, size_w]$ であり、 j の値の範囲は $[-1, size_h]$ であり、 $size_w = \min(W, m)$ 、 $size_h = \min(H, m)$ 、および $m = 16$ であり、

水平予測勾配行列は $X(i,j)$ によって表され、 i の値の範囲は $[0, size_w - 1]$ であり、 j の値の範囲は $[0, size_h - 1]$ であり、

垂直予測勾配行列は $Y(i,j)$ によって表され、 i の値の範囲は $[0, size_w - 1]$ であり、 j の値の範囲は $[0, size_h - 1]$ であり、20

W は現在のアフィンピクチャブロックの幅を表し、 H は現在のアフィンピクチャブロックの高さを表し、 (i, j) は現在のアフィンピクチャブロックの中の各サンプルの位置座標を表す。

【 0 4 2 4 】

現在のアフィンピクチャブロックが 16×16 ブロックへと暗黙的に区分され、勾配行列が各々の 16×16 ブロックのために計算されることが、前述の説明からわかり得る。 $m = 16$ は本明細書では例として使用されるにすぎず、限定するものとして解釈されるべきではないことを理解されたい。 $m = 32$ などの、 m の様々な他の値が使用され得る。

【 0 4 2 5 】

ある可能な設計では、方法は単予測のために使用され、動き情報は、第1の参照フレームリストに対応する第1の動き情報、または第2の参照フレームリストに対応する第2の動き情報を含み、30

第1の予測行列は第1の初期予測行列または第2の初期予測行列を含み(であり)、第1の初期予測行列は第1の動き情報に基づいて取得され、第2の初期予測行列は第2の動き情報に基づいて取得され、

水平予測勾配行列は第1の水平予測勾配行列または第2の水平予測勾配行列を含み(であり)、第1の水平予測勾配行列は延長された第1の初期予測行列に基づく計算を通じて取得され、第2の水平予測勾配行列は延長された第2の初期予測行列に基づく計算を通じて取得され、40

垂直予測勾配行列は第1の垂直予測勾配行列または第2の垂直予測勾配行列を含み(であり)、第1の垂直予測勾配行列は延長された第1の初期予測行列に基づく計算を通じて取得され、第2の垂直予測勾配行列は延長された第2の初期予測行列に基づく計算を通じて取得され、

デルタ予測値行列は、第1の参照フレームリストに対応する第1のデルタ予測値行列または第2の参照フレームリストに対応する第2のデルタ予測値行列を含み(であり)、第1のデルタ予測値行列は、第1の水平予測勾配行列、第1の垂直予測勾配行列、およびサブブロックの中心のサンプルに対する相対的なサブブロックの中の各サンプルユニットの第1の動きベクトル差分(たとえば、前方動きベクトル差分)に基づく計算を通じて取得され、第2のデルタ予測値行列は、第2の水平予測勾配行列、第2の垂直予測勾配行列、およびサブブロ

ックの中心のサンプルに対する相対的なサブブロックの中の各サンプルユニットの第2の動きベクトル差分(たとえば、後方動きベクトル差分)に基づく計算を通じて取得される。

【 0 4 2 6 】

ある可能な設計では、方法は双予測のために使用され、動き情報は、第1の参照フレームリストに対応する第1の動き情報および第2の参照フレームリストに対応する第2の動き情報を含み、

第1の予測行列は第1の初期予測行列および第2の初期予測行列を含み、第1の初期予測行列は第1の動き情報に基づいて取得され、第2の初期予測行列は第2の動き情報に基づいて取得され、

水平予測勾配行列は第1の水平予測勾配行列および第2の水平予測勾配行列を含み、第1の水平予測勾配行列は延長された第1の初期予測行列に基づく計算を通じて取得され、第2の水平予測勾配行列は延長された第2の初期予測行列に基づく計算を通じて取得され、

垂直予測勾配行列は第1の垂直予測勾配行列および第2の垂直予測勾配行列を含み、第1の垂直予測勾配行列は延長された第1の初期予測行列に基づく計算を通じて取得され、第2の垂直予測勾配行列は延長された第2の初期予測行列に基づく計算を通じて取得され、

デルタ予測値行列は、第1の参照フレームリストに対応する第1のデルタ予測値行列および第2の参照フレームリストに対応する第2のデルタ予測値行列を含み、第1のデルタ予測値行列は、第1の水平予測勾配行列、第1の垂直予測勾配行列、およびサブブロックの中心のサンプルに対する相対的なサブブロックの中の各サンプルユニットの第1の動きベクトル差分(たとえば、前方動きベクトル差分)に基づく計算を通じて取得され、第2のデルタ予測値行列は、第2の水平予測勾配行列、第2の垂直予測勾配行列、およびサブブロックの中心のサンプルに対する相対的なサブブロックの中の各サンプルユニットの第2の動きベクトル差分(たとえば、後方動きベクトル差分)に基づく計算を通じて取得される。

【 0 4 2 7 】

ある可能な設計では、方法は単予測のために使用される。

動き情報は、第1の参照フレームリストに対応する第1の動き情報、または第2の参照フレームリストに対応する第2の動き情報を含み、

第1の予測行列は第1の初期予測行列または第2の初期予測行列を含み(であり)、第1の初期予測行列は第1の動き情報に基づいて取得され、第2の初期予測行列は第2の動き情報に基づいて取得される。

【 0 4 2 8 】

ある可能な設計では、方法は双予測のために使用される。

動き情報は、第1の参照フレームリストに対応する第1の動き情報および第2の参照フレームリストに対応する第2の動き情報を含み、

第1の予測行列は第1の初期予測行列および第2の初期予測行列を含み、第1の初期予測行列は第1の動き情報に基づいて取得され、第2の初期予測行列は第2の動き情報に基づいて取得され、

サブブロックの動き情報に基づいてサブブロックの予測行列を取得するステップは、

サブブロックの予測行列を取得するために、第1の初期予測行列および第2の初期予測行列の中の同じ位置にあるサンプル値に対して加重加算を実行するステップを含む。加重加算がここで実行される前に、第1の初期予測行列および第2の初期予測行列の中のサンプル値は別々に洗練化され得ることを理解されたい。

【 0 4 2 9 】

別の可能な設計では、PROFプロセスは以下の4つのステップとして説明される。

【 0 4 3 0 】

ステップ1) サブブロック予測 $I(i,j)$ を生成するために、サブブロックベースのアフィン動き補償が実行される。たとえば、 i は $[0, subW+1]$ または $[-1, subW]$ からの値を有し、 j は $[0, subH+1]$ または $[-1, subH]$ からの値を有する。 i は $[0, subW+1]$ からの値を有し j は $[0, subH+1]$ からの値を有するので、左上サンプル(または座標の原点)は $(1, 1)$ に位置し、一方、 i は $[-1, subW]$ からの値を有し j は $[-1, subH]$ からの値を有するので、左上サンプルは $(0,$

10

20

30

40

50

0)に位置することが理解され得る。

【0 4 3 1】

ステップ2) サブブロック予測の空間勾配 $g_x(i,j)$ および $g_y(i,j)$ が、3タップフィルタ[-1, 0, 1]を使用して各サンプル位置において計算される。

$$g_x(i,j) = l(i+1,j) - l(i-1,j)$$

$$g_y(i,j) = l(i,j+1) - l(i,j-1)$$

【0 4 3 2】

サブブロック予測は、勾配計算のために両側で1サンプルだけ延長される。メモリの帯域幅と複雑さを下げるために、延長された境界上のサンプルは、参照ピクチャの中の最も近い整数サンプル位置からコピーされる。したがって、パディング領域のための追加の補間は避けられる。

10

【0 4 3 3】

ステップ3) ルマ予測洗練化がオプティカルフローの式によって計算される。

$$l(i,j) = g_x(i,j) * v_x(i,j) + g_y(i,j) * v_y(i,j)$$

ここで、図10に示されるように、 $v(i,j)$ は、 $v(i,j)$ によって表記されるサンプル位置(i, j)に対して計算されるサンプルMVと、サンプル(i, j)が属するサブブロックのサブブロックMVとの差分である。

【0 4 3 4】

言い換えると、各々の 4×4 の中心のサンプルのMVが計算され、次いでサブブロックの各サンプルのMVが計算される。各サンプルのMVと中心のサンプルのMVとの差分 $v(i,j)$ が取得され得る。

20

【0 4 3 5】

アフィンモデルパラメータおよびサブブロック中心に対する相対的なサンプル位置は、サブブロックごとに変化しないので、 $v(i,j)$ を第1のサブブロックのために計算することができ、同じCUの中の他のサブブロックのために再使用することができる。 x および y をサブブロックの中心に対するサンプル位置からの水平オフセットおよび垂直オフセットとし、 $v(x,y)$ は以下の式によって導出され得る。

【0 4 3 6】

【数2 1】

30

$$\begin{cases} \Delta v_x(x,y) = c * x + d * y \\ \Delta v_y(x,y) = e * x + f * y \end{cases}$$

【0 4 3 7】

4パラメータアフィンモデルに対して、

【0 4 3 8】

【数2 2】

40

$$\begin{cases} c = f = \frac{v_{1x} - v_{0x}}{w} \\ e = -d = \frac{v_{1y} - v_{0y}}{w} \end{cases}$$

【0 4 3 9】

6パラメータアフィンモデルに対して、

【0 4 4 0】

50

【数23】

$$\begin{cases} c = \frac{v_{1x} - v_{0x}}{w} \\ d = \frac{v_{2x} - v_{0x}}{h} \\ e = \frac{v_{1y} - v_{0y}}{w} \\ f = \frac{v_{2y} - v_{0y}}{h} \end{cases}$$

10

【0441】

ここで、 (v_{0x}, v_{0y}) 、 (v_{1x}, v_{1y}) 、 (v_{2x}, v_{2y}) は、左上、右上、および左下の制御点の動きベクトルであり、 w および h はCUの幅と高さである。

【0442】

ステップ4) 最後に、ルマ予測洗練化がサブブロック予測 $I(i,j)$ に加算される。最終的な予測 I' は、以下の式に示されるように生成される。

$$I'(i,j) = I(i,j) + I(i,j)$$

20

【0443】

図15は、本開示の別の態様による、アフィンコーディングされたブロックに対するオプティカルフローを用いた予測洗練化(PROP)のための装置1500を示す。ある例では、装置1500は、

PROPを適用するための複数の制約条件のいずれもが満たされていないと決定するために構成される決定ユニット1501と、

アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、オプティカルフローを用いた予測洗練化、アフィンコーディングされたブロックの現在のサブブロックに対するPROPプロセスを行うために構成される予測処理ユニット1503とを備える。アフィンコーディングされたブロックの各サブブロックの洗練化された予測サンプル値が生成されるとき、アフィンコーディングされたブロックの洗練化された予測サンプル値が自然に生成されることが理解され得る。

30

【0444】

別の例では、装置1500は、

複数のオプティカルフロー決定条件が満たされていると決定するために構成される決定ユニット1501であって、ここで、複数のオプティカルフロー決定条件はPROPの適用を許容する条件を指す、決定ユニットと、

アフィンコーディングされたブロックの現在のサブブロックの洗練化された予測サンプル値を取得するために、アフィンコーディングされたブロックの現在のサブブロックに対するPROPプロセスを行うために構成される予測処理ユニット1503とを備える。アフィンコーディングされたブロックの各サブブロックの洗練化された予測サンプル値が生成されるとき、アフィンコーディングされたブロックの洗練化された予測サンプル値が自然に生成されることが理解され得る。

40

【0445】

それに対応して、ある例では、装置1500の例示的な構造は、図2のエンコーダ20に対応していてよい。別の例では、装置1500の例示的な構造は、図3のデコーダ30に対応していてよい。

【0446】

別の例では、装置1500の例示的な構造は、図2のインター予測ユニット244に対応していてよい。別の例では、装置1500の例示的な構造は、図3のインター予測ユニット344

50

に対応していてもよい。

【0447】

本出願のこの実施形態において提供されるエンコーダ20またはデコーダ30の中の決定ユニットおよび予測処理ユニット(インター予測モジュールに対応する)は、前述の対応する方法に含まれる様々な実行ステップを実施するための機能エンティティであり、すなわち、本出願の方法のステップならびにこれらのステップの拡張および変形を完全に実施するための機能エンティティを有することが理解され得る。詳細については、対応する方法の前述の説明を参照されたい。簡潔するために、本明細書では詳細は再び説明されない。

【0448】

以下は、上で言及された実施形態において示されるような符号化方法と復号方法、およびそれらを使用するシステムの適用の説明である。

10

【0449】

図16は、コンテンツ配信サービスを実現するためのコンテンツ供給システム3100を示すブロック図である。このコンテンツ供給システム3100は、キャプチャデバイス3102、端末デバイス3106を含み、任意選択でディスプレイ3126を含む。キャプチャデバイス3102は、通信リンク3104を介して端末デバイス3106と通信する。通信リンクは、上で説明された通信チャネル13を含み得る。通信リンク3104は、限定はされないが、WIFI、イーサネット、ケーブル、ワイヤレス(3G/4G/5G)、USB、またはこれらの任意の種類の組合せなどを含む。

【0450】

キャプチャデバイス3102は、データを生成し、上の実施形態において示されるような符号化方法によってデータを符号化し得る。代替的に、キャプチャデバイス3102がデータをストリーミングサーバ(図には示されていない)に配信してもよく、サーバがデータを符号化して符号化されたデータを端末デバイス3106に送信する。キャプチャデバイス3102は、限定はされないが、カメラ、スマートフォンもしくはパッド、コンピュータもしくはラップトップ、ビデオ会議システム、PDA、車載デバイス、またはこれらの任意の組合せなどを含む。たとえば、キャプチャデバイス3102は、上で説明されたようなソースデバイス12を含み得る。データがビデオを含むとき、キャプチャデバイス3102に含まれるビデオエンコーダ20は、実際にビデオ符号化処理を実行し得る。データがオーディオ(すなわち、音声)を含むとき、キャプチャデバイス3102に含まれるオーディオエンコーダは、実際にオーディオ符号化処理を実行し得る。いくつかの現実的なシナリオでは、キャプチャデバイス3102は、符号化されたビデオデータとオーディオデータを、それらと一緒に多重化することによって配信する。他の現実的なシナリオでは、たとえばビデオ会議システムにおいて、符号化されたオーディオデータおよび符号化されたビデオデータは多重化されない。キャプチャデバイス3102は、符号化されたオーディオデータおよび符号化されたビデオデータを別々に端末デバイス3106に配信する。

20

30

【0451】

コンテンツ供給システム3100において、端末デバイス3106は符号化されたデータを受信して再生する。端末デバイス3106は、スマートフォンもしくはパッド3108、コンピュータもしくはラップトップ3110、ネットワークビデオレコーダ(NVR)/デジタルビデオレコーダ(DVR)3112、TV3114、セットトップボックス(STB)3116、ビデオ会議システム3118、ビデオ監視システム3120、携帯情報端末(PDA)3122、車載デバイス3124、または上で言及された符号化されたデータを復号することが可能な、これらの任意の組合せなどの、データ受信および復元能力のあるデバイスであり得る。たとえば、端末デバイス3106は、上で説明されたような宛先デバイス14を含み得る。符号化されたデータがビデオを含むとき、端末デバイスに含まれるビデオデコーダ30が、優先的にビデオ復号を実行する。符号化されたデータがオーディオを含むとき、端末デバイスに含まれるオーディオデコーダが、優先的にオーディオ復号処理を実行する。

40

【0452】

ディスプレイがある端末デバイス、たとえば、スマートフォンもしくはパッド3108、

50

コンピュータもしくはラップトップ3110、ネットワークビデオレコーダ(NVR)/デジタルビデオレコーダ(DVR)3112、TV3114、携帯情報端末(PDA)3122、または車載デバイス3124では、端末デバイスは、復号されたデータをそのディスプレイに供給することができる。ディスプレイがない端末デバイス、たとえばSTB3116、ビデオ会議システム3118、またはビデオ監視システム3120では、復号されたデータを受信して示すために、外部ディスプレイ3126が接続される。

【0453】

このシステムの中の各デバイスが符号化または復号を実行するとき、上で言及された実施形態において示されるような、ピクチャ符号化デバイスまたはピクチャ復号デバイスが使用され得る。

10

【0454】

図17は、端末デバイス3106の例の構造を示す図である。端末デバイス3106がキャプチャデバイス3102からストリームを受信した後、プロトコル進行ユニット3202がストリームの送信プロトコルを分析する。プロトコルは、限定はされないが、リアルタイムストリーミングプロトコル(RTSP)、ハイパーテキストトランスファープロトコル(HTTP)、HTTPライブストリーミングプロトコル(HLS)、MPEG-DASH、リアルタイムトранSPORTプロトコル(RTP)、リアルタイムメッセージングプロトコル(RTMP)、またはこれらの任意の種類の組合せなどを含む。

【0455】

プロトコル進行ユニット3202がストリームを処理した後、ストリームファイルが生成される。このファイルは逆多重化ユニット3204に出力される。逆多重化ユニット3204は、多重化されたデータを符号化されたオーディオデータおよび符号化されたビデオデータへと分離することができる。上で説明されたように、いくつかの現実的なシナリオでは、たとえばビデオ会議システムにおいて、符号化されたオーディオデータおよび符号化されたビデオデータは多重化されない。この状況では、符号化されたデータは、逆多重化ユニット3204を介さずに、ビデオデコーダ3206およびオーディオデコーダ3208に送信される。

20

【0456】

逆多重化処理を介して、ビデオエレメンタリストリーム(ES)、オーディオES、および任意選択で字幕が生成される。上で言及された実施形態において説明されるようなビデオデコーダ30を含むビデオデコーダ3206は、ビデオフレームを生成するために上で言及された実施形態に示されるような復号方法によってビデオESを復号し、このデータを同期ユニット3212に供給する。オーディオデコーダ3208は、オーディオフレームを生成するためにオーディオESを復号し、このデータを同期ユニット3212に供給する。代替的に、ビデオフレームは、それを同期ユニット3212に供給する前に、バッファ(図Yには示されていない)に記憶してもよい。同様に、オーディオフレームは、それを同期ユニット3212に供給する前に、バッファ(図Yには示されていない)に記憶してもよい。

30

【0457】

同期ユニット3212は、ビデオフレームおよびオーディオフレームを同期し、ビデオ/オーディオをビデオ/オーディオディスプレイ3214に供給する。たとえば、同期ユニット3212は、ビデオ情報およびオーディオ情報の提示を同期する。情報は、コーディングされるオーディオデータとビジュアルデータの提示に関するタイムスタンプと、データストリーム自体の配信に関するタイムスタンプとを使用して、シンタックスにおいてコーディングされ得る。

40

【0458】

字幕がストリームに含まれる場合、字幕デコーダ3210は、字幕を復号し、それをビデオフレームおよびオーディオフレームと同期し、ビデオ/オーディオ/字幕をビデオ/オーディオ/字幕ディスプレイ3216に供給する。

【0459】

本発明は上で言及されたシステムに限定されず、上で言及された実施形態におけるピク

50

チャ符号化デバイスまたはピクチャ復号デバイスのいずれかが他のシステム、たとえばカーシステムに組み込まれ得る。

【0460】

本出願において使用される数学演算子は、Cプログラミング言語において使用されるものと同様である。しかしながら、整数除算および算術シフト演算の結果はより正確に定義され、指数および実数値の除算などの、追加の演算が定義される。

【0461】

この実施形態における関連する内容、関連するステップの実装形態、有益な効果などの説明については、前述の対応する部分を参照し、または前述の対応する部分に基づいて簡単な修正が行われてもよい。詳細はここでは再び説明されない。

10

【0462】

矛盾が発生しない場合、前述の実施形態の任意の2つ以上におけるいくつかの特徴が、新しい実施形態を形成するために組み合わせられてもよいことに留意されたい。加えて、前述の実施形態のいずれか1つにおけるいくつかの特徴は、ある実施形態として独立に使用されてもよい。

【0463】

上記は主に、方法の観点から本出願の実施形態において提供される解決法を説明する。前述の機能を実装するために、機能を実行するための対応するハードウェア構造および/またはソフトウェアモジュールが含まれる。本明細書において開示される実施形態において説明される例と組み合わせて、ユニットおよびアルゴリズムステップは、本出願においてハードウェアまたはハードウェアとコンピュータソフトウェアの組合せによって実装され得ることを、当業者は容易に認識されたい。機能がハードウェアによって実行されるか、またはコンピュータソフトウェアにより駆動されるハードウェアによって実行されるかは、技術的な解決法の具体的な適用形態と設計制約に依存する。各々の特定の適用形態に対して、当業者は異なる方法を使用して説明される機能を実装することがあるが、その実装形態は本出願の範囲外であると見なされるべきではない。

20

【0464】

本出願の実施形態における機能モジュールへのエンコーダ/デコーダの分割は、前述の方法の例に基づいて実行され得る。たとえば、各機能モジュールは各機能に対応した分割を通じて取得されてもよく、または、少なくとも2つの機能が1つの処理モジュールに統合されてもよい。統合されたモジュールは、ハードウェアの形式で実装されてもよく、または、ソフトウェア機能モジュールの形式で実装されてもよい。本出願の実施形態では、モジュールの分割は例であり、論理的な機能の分割にすぎないことに留意されたい。実際の実装形態では、別の分割方式が使用され得る。

30

【0465】

本発明の実施形態は、ビデオコーディングに基づいて主に説明されたが、コーディングシステム10、エンコーダ20、およびデコーダ30(およびそれに対応してシステム10)、ならびに本明細書において説明される他の実施形態は、静止ピクチャの処理またはコーディング、すなわち、ビデオコーディングにおけるような先行ピクチャまたは連続するピクチャとは無関係な個々のピクチャの処理またはコーディングのためにも構成され得ることに留意されたい。一般に、ピクチャ処理コーディングが単一のピクチャ17に限定される場合、インター予測ユニット244(エンコーダ)および344(デコーダ)だけが、利用可能ではないことがある。ビデオエンコーダ20およびビデオデコーダ30のすべての他の機能(ツールまたは技術とも呼ばれる)が、静止ピクチャ処理、たとえば、残差計算204/304、変換206、量子化208、逆量子化210/310、(逆)変換212/312、区分化262/362、イントラ予測254/354、および/またはループフィルタリング220、320、ならびにエントロピーコーディング270およびエントロピー復号304のために等しく使用され得る。

40

【0466】

たとえばエンコーダ20およびデコーダ30の実施形態、ならびに、たとえばエンコーダ20およびデコーダ30を参照して本明細書において説明される機能は、ハードウェア、ソフ

50

トウェア、ファームウェア、またはこれらの任意の組合せにおいて実装され得る。ソフトウェアにおいて実装される場合、機能は、コンピュータ可読媒体に記憶され、または1つまたは複数の命令もしくはコードとして通信媒体を介して送信され、ハードウェアベースの処理ユニットによって実行され得る。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対応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つて、ある場所から別の場所へのコンピュータプログラムの移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)信号波もしくは搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示において説明される技法の実施のために命令、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つまたは複数のプロセッサ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。

【0467】

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もしくは他の光学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ストレージデバイス、フラッシュメモリ、または命令もしくはデータ構造の形で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含むことができる。また、あらゆる接続が適切にコンピュータ可読媒体と呼ばれる。たとえば、命令が、同軸ケーブル、光ファイバケーブル、ツイストペア、デジタル加入者線(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が、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、代わりに非一時的な有形記憶媒体を対象とすることを理解されたい。本明細書において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(CD)、レーザーディスク(登録商標)、光ディスク、デジタル多用途ディスク(DVD)、フロッピーディスク、およびBlu-ray(登録商標)ディスクを含み、ディスク(disk)は通常データを磁気的に再生し、ディスク(disc)はレーザーでデータを光学的に再生する。上記の組合せも、コンピュータ可読媒体の範囲内に含まれるべきである。

【0468】

命令は、1つまたは複数のデジタルシグナルプロセッサ(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ゲートアレイ(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によって実行され得る。したがって、本明細書において使用されるような「プロセッサ」という用語は、本明細書において説明される技法の実施に適した前述の構造または任意の他の構造のいずれをも指し得る。加えて、いくつかの態様では、本明細書において説明される機能は、符号化および復号のために構成される、または合成されたコーデックに組み込まれる、専用のハードウェアおよび/またはソフトウェアモジュール内で提供され得る。また、技法は、1つまたは複数の回路または論理要素において完全に実施され得る。

【0469】

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るデバイスの機能的な態様を強調するために、様々なコンポーネント、モジュール、またはユニットが本開示において説明されるが、それらは異なるハードウェアユニットによる実現を必ずしも必要としない。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せられてもよく、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連携して、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運用可能なハードウェアユニット

10

20

30

40

50

の集合体によって提供されてもよい。

【符号の説明】

【0 4 7 0】

10	ビデオコーディングシステム	
12	ソースデバイス	
13	通信チャネル	
14	宛先デバイス	
16	ピクチャソース	10
17	ピクチャデータ	
18	プリプロセッサ	
19	前処理されたピクチャデータ	
20	エンコーダ	
21	符号化されたピクチャデータ	
22	通信インターフェース	
28	通信インターフェース	
30	デコーダ	
31	復号されたピクチャデータ	
32	ポストプロセッサ	
33	後処理されたピクチャデータ	
34	表示デバイス	20
40	ビデオコーディングシステム	
41	イメージングデバイス	
42	アンテナ	
43	プロセッサ	
44	メモリストア	
45	表示デバイス	
46	処理回路	
201	入力	
203	ピクチャブロック	
204	残差計算ユニット	30
205	残差ブロック	
206	変換処理ユニット	
207	変換係数	
208	量子化ユニット	
209	量子化された係数	
210	逆量子化ユニット	
211	逆量子化された係数	
212	逆変換処理ユニット	
213	再構築された残差ブロック	
214	再構築ユニット	40
215	再構築されたブロック	
220	ループフィルタユニット	
221	フィルタリングされたブロック	
230	復号ピクチャバッファ	
231	復号されたピクチャ	
244	インター予測ユニット	
254	イントラ予測ユニット	
260	モード選択ユニット	
262	区分ユニット	
265	予測ブロック	50

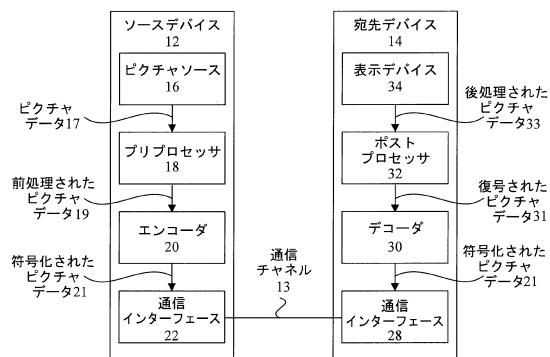
266	シンタックス要素	
270	エントロピー符号化ユニット	
272	出力	
304	エントロピー復号ユニット	
309	量子化された係数	
310	逆量子化ユニット	
311	逆量子化された係数	
312	逆変換処理ユニット	
313	再構築された残差ブロック	10
314	再構築ユニット	
315	再構築されたブロック	
320	ループフィルタ	
321	フィルタリングされたブロック	
330	復号ピクチャバッファ	
331	復号されたピクチャ	
332	出力	
344	インター予測ユニット	
354	イントラ予測ユニット	
360	モード適用ユニット	
365	予測ブロック	20
366	シンタックス要素	
400	ビデオコーディングデバイス	
410	入力ポート	
420	受信器ユニット	
430	プロセッサ	
440	送信器ユニット	
450	出力ポート	
460	メモリ	
470	コーディングモジュール	
500	装置	30
502	プロセッサ	
504	メモリ	
506	データ	
508	オペレーティングシステム	
510	アプリケーションプログラム	
512	バス	
518	ディスプレイ	
900	予測信号ウィンドウ	
1500	装置	
1501	決定ユニット	40
1503	予測処理ユニット	
3100	コンテンツ供給システム	
3102	キャプチャデバイス	
3104	通信リンク	
3106	端末デバイス	
3108	スマートフォン/パッド	
3110	コンピュータ/ラップトップ	
3112	ネットワークビデオレコーダ(NVR)/デジタルビデオレコーダ(DVR)	
3114	TV	
3116	セットトップボックス(STB)	50

- 3118 ビデオ会議システム
 3120 ビデオ監視システム
 3122 携帯情報端末(PDA)
 3124 車載デバイス
 3126 ディスプレイ
 3202 プロトコル進行ユニット
 3204 逆多重化ユニット
 3206 ビデオデコーダ
 3208 オーディオデコーダ
 3210 字幕デコーダ
 3212 同期ユニット
 3214 ビデオ/オーディオディスプレイ
 3216 ビデオ/オーディオ/字幕ディスプレイ

10

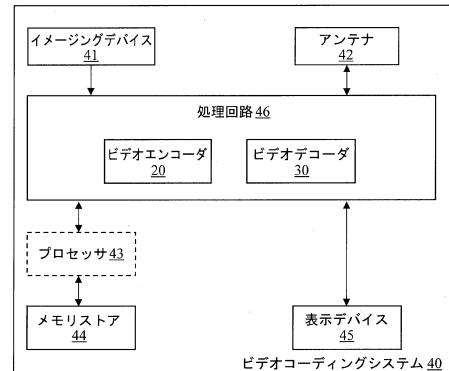
【図面】

【図 1 A】



10

【図 1 B】



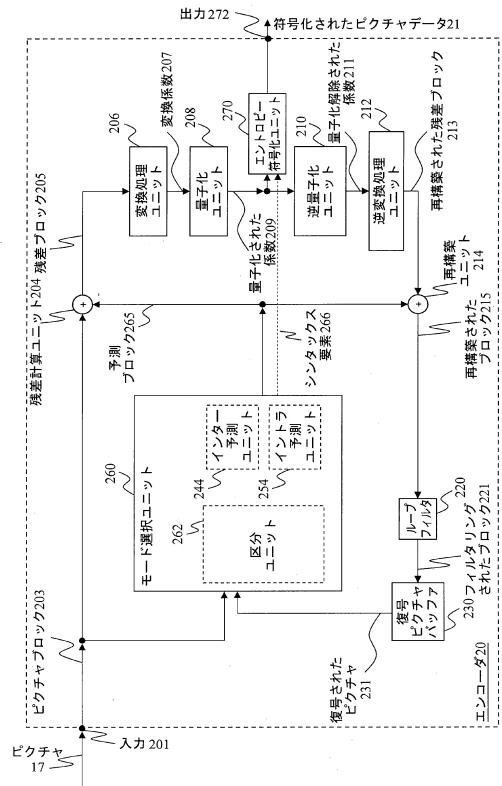
20

30

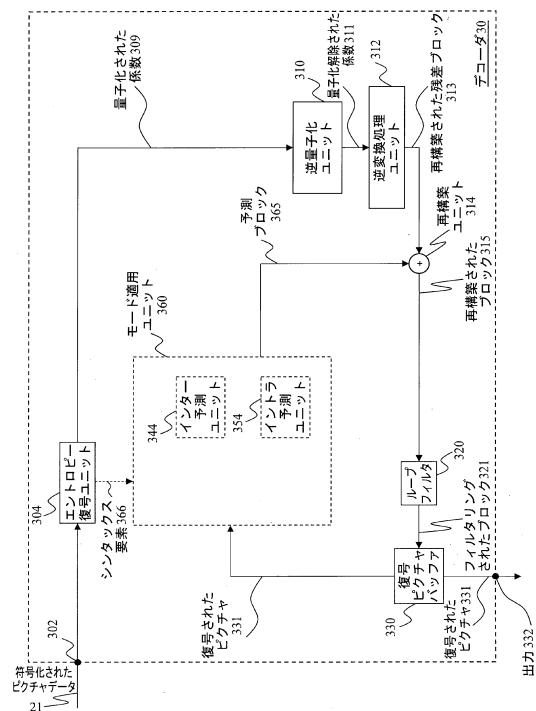
40

50

【図2】



【図3】

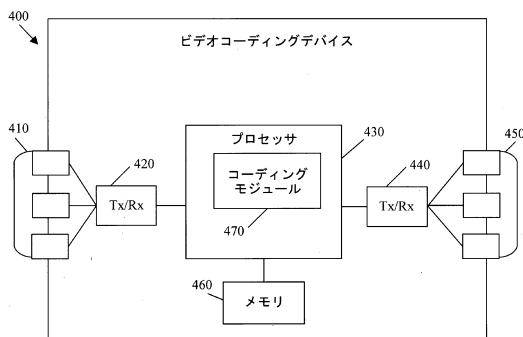


10

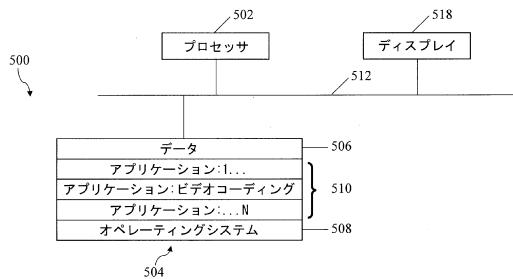
20

30

【図4】



【図5】



40

50

【図 6】

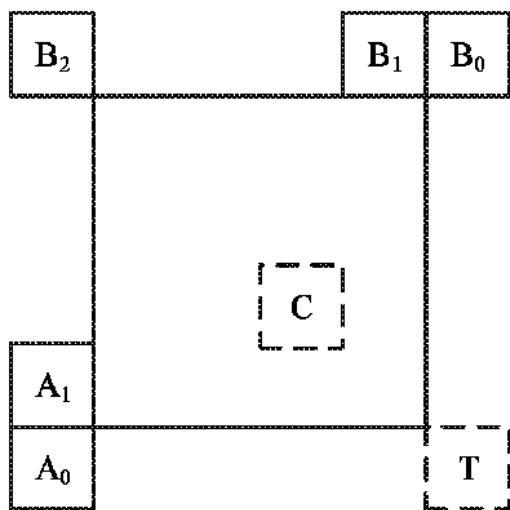


FIG. 6

10

【図 7】

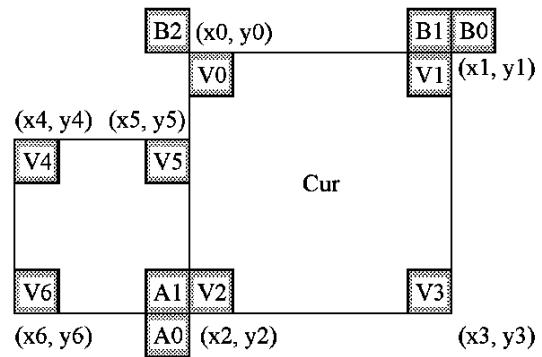
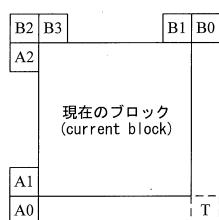


FIG. 7

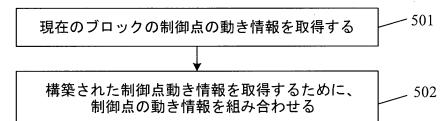
20

【図 8 A】



30

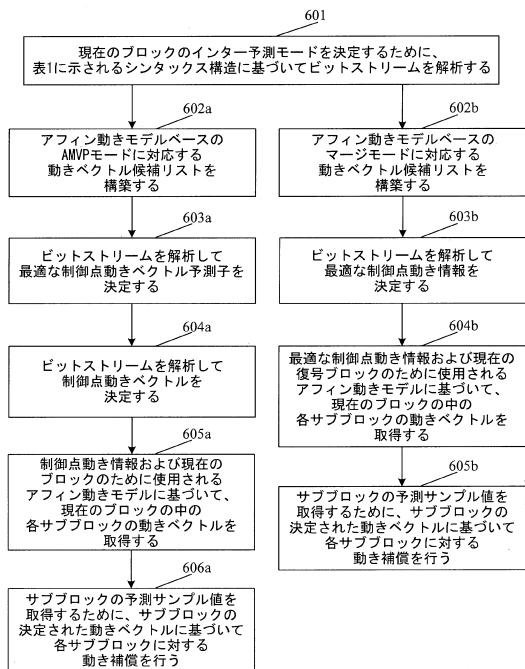
【図 8 B】



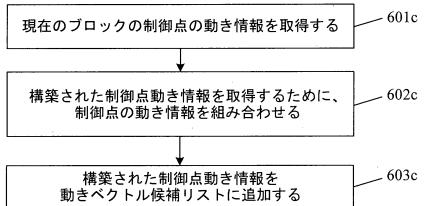
40

50

【図 9 A】

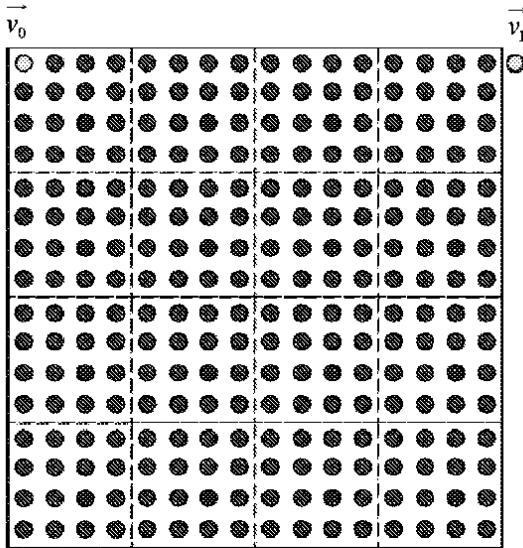


10

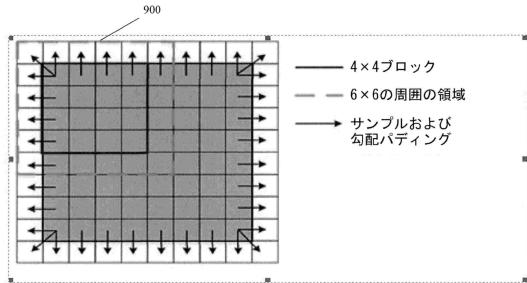


20

【図 9 C】



【図 9 D】



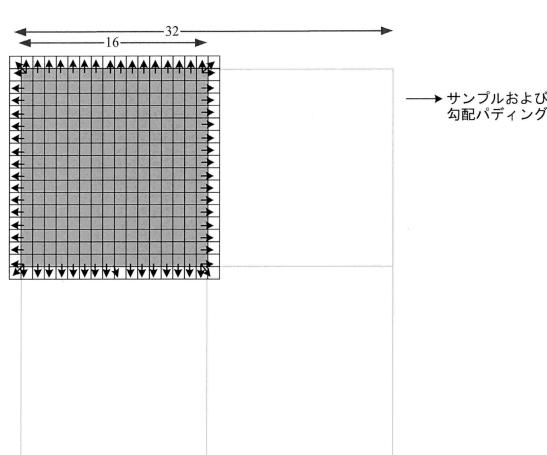
30

40

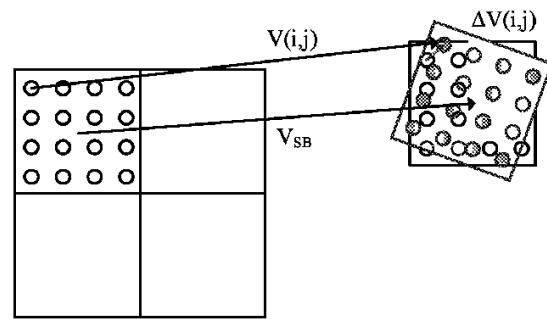
FIG. 9C

50

【図 9 E】



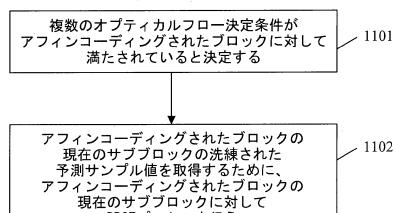
【図 10】



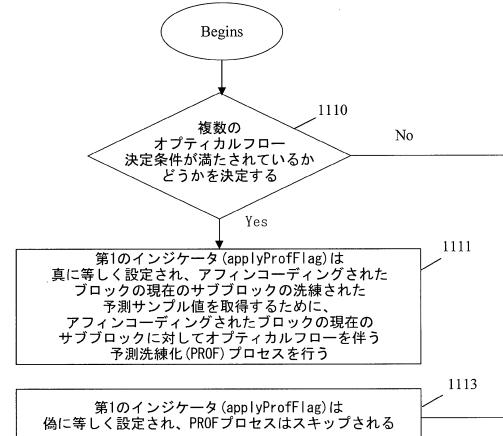
10

FIG. 10

【図 11 A】



【図 11 B】



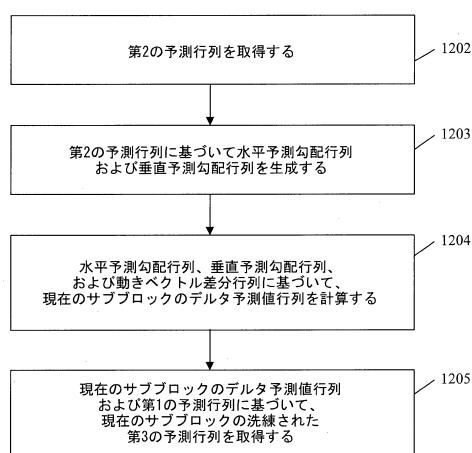
20

30

40

50

【図 1 2】



【図 1 3】

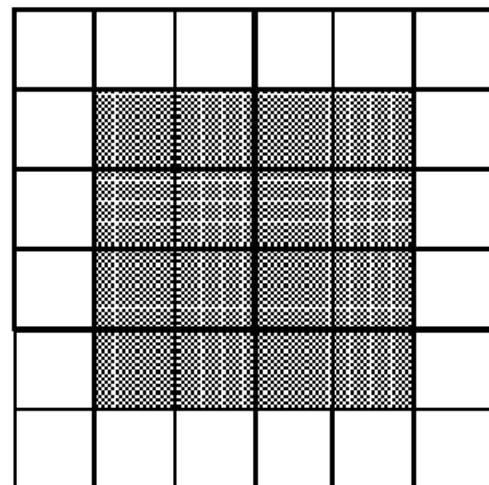


FIG. 13

20

【図 1 4】

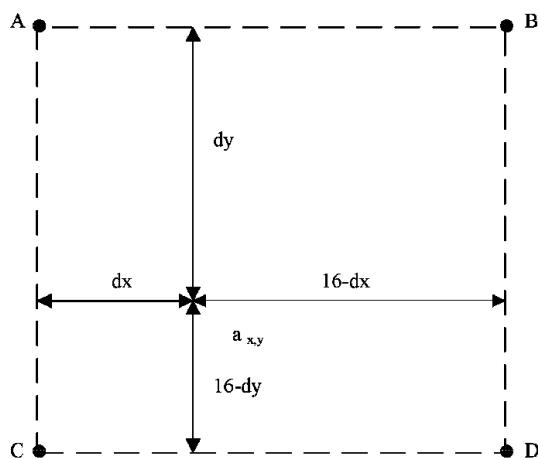
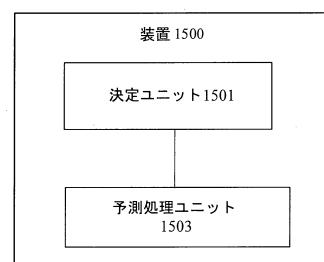


FIG. 14

【図 1 5】

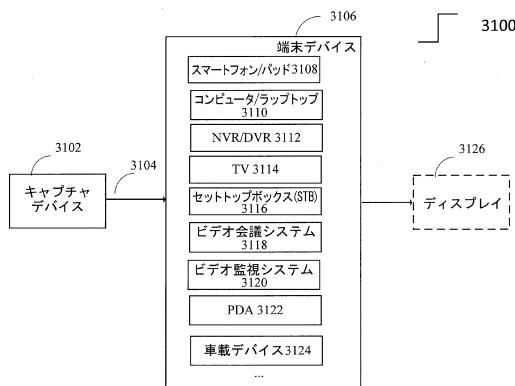


30

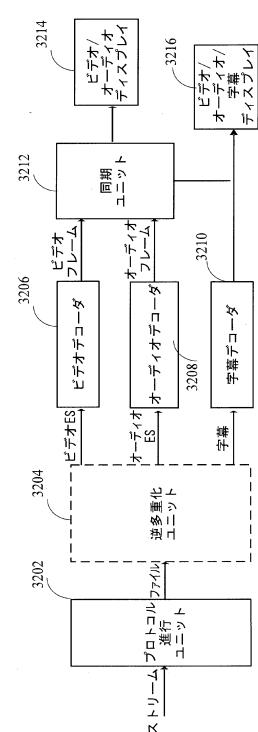
40

50

【図16】



【図17】



10

20

30

40

50

フロントページの続き

米国(US)

(31)優先権主張番号 62/839,765

(32)優先日 平成31年4月28日(2019.4.28)

(33)優先権主張国・地域又は機関

米国(US)

トリクト・パンティアン・(番地なし)・ホアウェイ・アドミニストレーション・ビルディング

(72)発明者 ハイタオ・ヤン

中華人民共和国・518129・グァンドン・シェンツエン・ロンガン・ディストリクト・パンティアン・(番地なし)・ホアウェイ・アドミニストレーション・ビルディング

(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・95050・サンタ・クララ・セントラル・エキスプレスウェイ・2330

合議体

審判長 畑中 高行

審判官 伊藤 隆夫

審判官 坂本 聰生

(56)参考文献 Jiancong (Daniel) Luo et al.、"CE2-related: Prediction refinement with optical flow for a fine mode"、Joint Video Experts Team (JVET)、2019.03.19 [JVET-N0236-r2] (version 3)

(58)調査した分野 (Int.Cl., DB名)

H04N 19/00 - 19/98