

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7270725号
(P7270725)

(45)発行日 令和5年5月10日(2023.5.10)

(24)登録日 令和5年4月27日(2023.4.27)

(51)国際特許分類 F I
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70
H 0 4 N 19/503 (2014.01) H 0 4 N 19/503

請求項の数 12 (全49頁)

(21)出願番号	特願2021-514086(P2021-514086)	(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・グァン ドン・シェンツェン・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
(86)(22)出願日	令和1年9月12日(2019.9.12)	(74)代理人	100110364 弁理士 実広 信哉
(65)公表番号	特表2022-500919(P2022-500919 A)	(74)代理人	100133569 弁理士 野村 進
(43)公表日	令和4年1月4日(2022.1.4)	(72)発明者	フヌ・ヘンドリー アメリカ合衆国・カリフォルニア・92 129・サン・ディエゴ・フォックスク 最終頁に続く
(86)国際出願番号	PCT/US2019/050853		
(87)国際公開番号	WO2020/056164		
(87)国際公開日	令和2年3月19日(2020.3.19)		
審査請求日	令和3年5月12日(2021.5.12)		
(31)優先権主張番号	62/730,172		
(32)優先日	平成30年9月12日(2018.9.12)		
(33)優先権主張国・地域又は機関	米国(US)		
(31)優先権主張番号	62/848,147		
(32)優先日	令和1年5月15日(2019.5.15)		
	最終頁に続く		

(54)【発明の名称】 ビデオエンコーダ、ビデオデコーダ、および対応する方法

(57)【特許請求の範囲】

【請求項1】

ビデオデコーダによって実施されるコーディングされたビデオビットストリームを復号する方法であって、前記コーディングされたビデオビットストリームが、複数の参照ピクチャエントリを各々含む複数の参照ピクチャリストシンタックス構造を備え、

前記コーディングされたビデオビットストリームから、前記複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウンタ(POC)の符号値を指定するフラグを構文解析するステップと、

前記コーディングされたビデオビットストリームから、前記短期参照ピクチャエントリと関連付けられる前記デルタPOCの絶対値を構文解析するステップであって、前記デルタPOCの前記絶対値が、符号なし整数0次指数ゴロム符号化に基づいて復号される、ステップと、

前記フラグと前記デルタPOCの前記絶対値が互いに別々に構文解析され、前記フラグが構文解析される前に前記デルタPOCの前記絶対値が構文解析されるものであり、

前記デルタPOCの前記符号値および前記デルタPOCの前記絶対値に基づいて参照ピクチャリストを生成するステップであって、前記短期参照ピクチャエントリと関連付けられる前記デルタPOCが、前記参照ピクチャリストの中の対応する参照ピクチャのPOC値を導出するために使用される、ステップと、

再構築されたブロックを生成するために前記参照ピクチャリストに基づいてインター予測を実行するステップとを備える、方法。

【請求項 2】

前記フラグが、前記複数の参照ピクチャリストシンタックス構造のうちの対応する1つの中のすべてのデルタPOC値の前記符号値が同じであることを示す、請求項1に記載の方法。

【請求項 3】

前記フラグが、前記複数の参照ピクチャリストシンタックス構造のうちの対応する1つの中のすべてのデルタPOC値の前記符号値が正であることを示す、請求項1に記載の方法。

【請求項 4】

前記フラグが、前記複数の参照ピクチャリストシンタックス構造のうちの対応する1つの中のすべてのデルタPOC値の前記符号値が負であることを示す、請求項1に記載の方法。

10

【請求項 5】

前記フラグおよび前記参照ピクチャリストシンタックス構造が同じパラメータセットにおいてシグナリングされる、請求項1から4のいずれか一項に記載の方法。

【請求項 6】

前記フラグが1に等しいとき、対応する参照ピクチャリストシンタックス構造の中のすべてのエントリの前記符号値が、前記参照ピクチャリストシンタックス構造においてシグナリングされる、請求項1から5のいずれか一項に記載の方法。

【請求項 7】

前記フラグが0に等しいとき、前記符号値が、参照ピクチャリストシンタックス構造の中の前記短期参照ピクチャエントリと関連付けられる各エントリのためにシグナリングされる、請求項1から6のいずれか一項に記載の方法。

20

【請求項 8】

ビデオエンコーダによって実施されるビデオビットストリームを符号化する方法であって、

再構築されたブロックを生成するために参照ピクチャリストを生成するステップと、

前記参照ピクチャリストに基づいて参照ピクチャリストシンタックス構造を前記ビデオビットストリームへと符号化するステップとを備え、前記参照ピクチャリストシンタックス構造が複数の参照ピクチャエントリを備え、

前記複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウンタ(POC)の符号値を指定するフラグおよび前記短期参照ピクチャエントリと関連付けられる前記デルタPOCの絶対値が、前記参照ピクチャリストシンタックス構造に含まれ、前記短期参照ピクチャエントリと関連付けられる前記デルタPOCが、前記参照ピクチャリストの中の対応する参照ピクチャのPOC値に基づいて取得され、前記デルタPOCの前記絶対値が、符号なし整数0次指数ゴロム符号化に基づいて符号化される、方法であって、

30

前記フラグと前記デルタPOCの前記絶対値が互いに別々に構文解析され、前記フラグが構文解析される前に前記デルタPOCの前記絶対値が構文解析されるものである、方法。

【請求項 9】

請求項8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。

【請求項 10】

請求項1から7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。

40

【請求項 11】

コンピュータまたはプロセッサ上で実行されると、請求項1から8のいずれか一項に記載の方法を実行するためのプログラムコードを備える、コンピュータプログラム。

【請求項 12】

コンピュータデバイスによって実行されると、請求項1から8のいずれか一項に記載の方法を前記コンピュータデバイスに実行さ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。

【発明の詳細な説明】

【技術分野】

50

【 0 0 0 2 】

全般に、本開示は、ビデオコーディングにおける参照ピクチャ管理に対するシグナリング効率の改善のための技法を説明する。より具体的には、本開示は、参照ピクチャリストの構築および参照ピクチャリストに直接基づく参照ピクチャのマーキングのための、改善されたシグナリングのための技法を説明する。

【 背景技術 】**【 0 0 0 3 】**

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なビデオデータの量は相当なものであることがあり、このことは、限られた帯域幅容量でデータがストリーミングされる時、または通信ネットワークを介して別様に通信されるときに、困難さをもたらし得る。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、ビデオが記憶デバイスに記憶されるときにも、ビデオのサイズは問題となり得る。ビデオ圧縮デバイスはしばしば、送信または記憶の前にビデオデータをコーディングするためにソースにおけるソフトウェアおよび/またはハードウェアを使用し、それにより、デジタルビデオ画像を表現するために必要なデータの量を減らす。次いで、圧縮されたデータが、ビデオデータを復号するビデオ展開デバイスによってデスティネーションにおいて受信される。限られたネットワークリソースと、より高品質なビデオの増え続ける需要により、画像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高める、改善された圧縮および展開技法が望ましい。

【 発明の概要 】**【 課題を解決するための手段 】****【 0 0 0 4 】**

第1の態様は、ビデオデコーダによって実施されるコーディングされたビデオビットストリームを復号する方法に関し、コーディングされたビデオビットストリームは、複数の参照ピクチャエントリを各々含む複数の参照ピクチャリスト構造を備える。方法は、コーディングされたビデオビットストリームから、複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウンタ(POC)の絶対値の符号値を指定するフラグを構文解析するステップと、コーディングされたビデオビットストリームから、短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値を構文解析するステップと、デルタPOCの絶対値の符号値およびデルタPOCの絶対値に基づいて参照ピクチャリストを生成するステップと、再構築されたブロックを生成するために参照ピクチャリストに基づいてインター予測を実行するステップとを備える。

【 0 0 0 5 】

方法は、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値と絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信されるとき、受信されるとき、および/または見られるときに、より良いユーザ体験をユーザに提供する。

【 0 0 0 6 】

第1の態様自体による方法の第1の実装形態では、フラグは、複数の参照ピクチャリスト構造のうちの対応する1つの中のすべてのデルタPOC値の符号値が同じであることを示す。

【 0 0 0 7 】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第2の実装形態では、フラグは、複数の参照ピクチャリスト構造のうちの対応する1つの中のすべてのデルタPOC値の符号値が正であることを示す。

【 0 0 0 8 】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第3の実装形態

10

20

30

40

50

では、フラグは、複数の参照ピクチャリスト構造のうちの対応する1つの中のすべてのデルタPOC値の符号値が負であることを示す。

【0009】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第4の実装形態では、フラグはall_rpl_entries_same_sign_flagと指定される。

【0010】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第5の実装形態では、フラグはシーケンスパラメータセット(SPS)においてシグナリングされる。

【0011】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第6の実装形態では、フラグおよび参照ピクチャリスト構造は同じパラメータセットにおいてシグナリングされる。

10

【0012】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第7の実装形態では、フラグが1に等しいとき、対応する参照ピクチャリスト構造の中のすべてのエントリの符号値は、参照ピクチャリスト構造においてシグナリングされる。

【0013】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第8の実装形態では、フラグが0に等しいとき、符号値は、参照ピクチャリスト構造の中の短期参照ピクチャエントリと関連付けられる各エントリのためにシグナリングされる。

20

【0014】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第9の実装形態では、デルタPOCの絶対値は、符号なし整数0次指数ゴロム符号化に基づいて復号される。

【0015】

第1の態様自体または第1の態様の任意の先行する実装形態による方法の第10の実装形態では、フラグおよび絶対値は互いに別々に構文解析される。

【0016】

第2の態様は、ビデオエンコーダによって実施されるビデオビットストリームを符号化する方法に関する。方法は、複数の参照ピクチャエントリを各々含む複数の参照ピクチャリスト構造をビデオビットストリームへと符号化するステップと、複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウント(POC)の絶対値の符号値を指定するフラグをビデオビットストリームへと符号化するステップと、短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値をビデオビットストリームへと符号化するステップと、ビデオビットストリームをビデオデコーダに向かって送信するステップとを備える。

30

【0017】

方法は、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値と絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信されるとき、受信されるとき、および/または見られるときに、より良いユーザ体験をユーザに提供する。

40

【0018】

第2の態様自体による方法の第1の実装形態では、フラグは、複数の参照ピクチャリスト構造のうちの対応する1つの中のすべてのデルタPOC値の符号値が同じであることを示す。

【0019】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第2の実装形態では、フラグは、複数の参照ピクチャリスト構造のうちの対応する1つの中のすべてのデルタPOC値の符号値が正であることを示す。

50

【 0 0 2 0 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第3の実装形態では、フラグは、複数の参照ピクチャリスト構造のうちの対応する1つの中のすべてのデルタPOC値の符号値が負であることを示す。

【 0 0 2 1 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第4の実装形態では、フラグはall_rpl_entries_same_sign_flagと指定される。

【 0 0 2 2 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第5の実装形態では、フラグはビデオビットストリームのシーケンスパラメータセット(SPS)において符号化される。

10

【 0 0 2 3 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第6の実装形態では、フラグおよび参照ピクチャリスト構造は同じパラメータセットにおいて符号化される。

【 0 0 2 4 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第7の実装形態では、フラグが1に等しいとき、参照ピクチャリスト構造の中のすべてのエントリの符号値は、参照ピクチャリスト構造において符号化される。

【 0 0 2 5 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第8の実装形態では、フラグが0に等しいとき、符号値は、参照ピクチャリスト構造の中の短期参照ピクチャエントリと関連付けられる各エントリのために符号化される。

20

【 0 0 2 6 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第9の実装形態では、デルタPOCの絶対値は、符号なし整数0次指数ゴロム符号化に基づいて復号される。

【 0 0 2 7 】

第2の態様自体または第2の態様の任意の先行する実装形態による方法の第10の実装形態では、フラグおよび絶対値は互いに別々に符号化される。

【 0 0 2 8 】

第3の態様は、ビデオデコーダによって実施されるコーディングされたビデオビットストリームを復号する方法に関する。方法は、コーディングされたビデオビットストリームにおいて表される参照ピクチャリスト構造を構文解析するステップであって、参照ピクチャリスト構造がシンタックス要素のセットを備え、シンタックス要素のセットが参照ピクチャを記述してピクチャ順序カウンタのデルタ値および符号フラグを含み、ピクチャ順序カウンタのデルタ値が参照ピクチャのピクチャ順序カウンタを取得するために使用され、符号フラグがピクチャ順序カウンタのデルタ値の符号値を指定する、ステップと、参照ピクチャリスト構造に基づいて、現在のスライスの参照ピクチャリストを導出するステップであって、参照ピクチャリストが参照ピクチャを備える、ステップと、参照ピクチャリストに基づいて、現在のスライスの少なくとも1つの再構築されたブロックを取得するステップとを備える。

30

40

【 0 0 2 9 】

方法は、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信されるとき、受信されるとき、および/または見られるときに、より良いユーザ体験をユーザに提供する。

【 0 0 3 0 】

50

第3の態様自体による方法の第1の実装形態では、ピクチャ順序カウンタのデルタ値および符号フラグが別々に復号される。

【0031】

第3の態様自体または第3の態様の任意の先行する実装形態による方法の第2の実装形態では、ピクチャ順序カウンタのデルタ値は絶対値である。

【0032】

第4の態様は、複数の参照ピクチャエントリを各々含む複数の参照ピクチャリスト構造を備えるコーディングされたビデオビットストリームを受信するように構成される受信機と、受信機に結合されるメモリであって、命令を記憶する、メモリと、メモリに結合されるプロセッサとを備える復号デバイスに関し、プロセッサは、プロセッサに、複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウンタ(POC)の絶対値の符号値を指定するフラグを構文解析させ、短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値を構文解析させ、デルタPOCの絶対値の符号値およびデルタPOCの絶対値に基づいて参照ピクチャリストを生成させ、再構築されたブロックを生成するために参照ピクチャリストに基づいてインター予測を実行させるために、メモリに記憶されている命令を実行するように構成される。

10

【0033】

復号デバイスは、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信される時、受信される時、および/または見られるときに、より良いユーザ体験をユーザに提供する。

20

【0034】

第4の態様自体による復号デバイスの第1の実装形態では、復号デバイスは、再構築されたブロックを使用して生成される画像を表示するように構成されるディスプレイを含む。

【0035】

第5の態様は、複数の参照ピクチャエントリを各々含む複数の参照ピクチャリスト構造をビデオビットストリームへと符号化し、複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウンタ(POC)の絶対値の符号値を指定するフラグをビデオビットストリームへと符号化し、短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値をビデオビットストリームへと符号化するように構成される、プロセッサと、プロセッサに結合される送信機であって、ビデオビットストリームをデコーダに向かって送信するように構成される、送信機とを備える、符号化デバイスに関する。

30

【0036】

符号化デバイスは、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信される時、受信される時、および/または見られるときに、より良いユーザ体験をユーザに提供する。

40

【0037】

第5の態様自体による符号化デバイスの第1の実装形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値の符号値が同じであることを示す。

【0038】

第5の態様自体による符号化デバイスの第2の実装形態では、フラグはall_rpl_entries_

50

same_sign_flagと指定される。

【0039】

第6の態様は、復号すべきビットストリームを受信するように構成される受信機と、受信機に結合される送信機であって、復号された画像をディスプレイに送信するように構成される、送信機と、受信機または送信機のうちの少なくとも1つに結合されるメモリであって、メモリが命令を記憶するように構成される、メモリと、メモリに結合されるプロセッサであって、本明細書において開示される実施形態のいずれかにおける方法を実行するためにメモリに記憶されている命令を実行するように構成される、プロセッサとを備える、コーディング装置に関する。

【0040】

コーディング装置は、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信される時、受信される時、および/または見られるときに、より良いユーザ体験をユーザに提供する。

【0041】

第7の態様は、エンコーダと、エンコーダと通信しているデコーダとを備えるシステムに関し、エンコーダまたはデコーダは、復号デバイス、符号化デバイス、または本明細書において開示される実施形態のいずれかにおけるコーディング装置を含む。

【0042】

システムは、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信される時、受信される時、および/または見られるときに、より良いユーザ体験をユーザに提供する。

【0043】

第8の態様は、復号すべきビットストリームを受信するように構成される受信手段と、受信手段に結合される送信手段であって、復号された画像を表示手段に送信するように構成される、送信手段と、受信手段または送信手段のうちの少なくとも1つに結合される記憶手段であって、命令を記憶するように構成される、記憶手段と、記憶手段に結合される処理手段であって、本明細書において開示される実施形態のいずれかにおける方法を実行するために記憶手段に記憶されている命令を実行するように構成される、処理手段とを備える、コーディングのための手段に関する。

【0044】

コーディングのための手段は、コーディングプロセスを簡単にしてより効率的にする技法を提供する。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーダ/デコーダ(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信される時、受信される時、および/または見られるときに、より良いユーザ体験をユーザに提供する。

【0045】

本開示のより完全な理解のために、添付の図面および詳細な説明に関連して、ここで以下の簡単な説明に対する参照が行われ、同様の参照番号は同様の部分を表す。

【図面の簡単な説明】

【0046】

10

20

30

40

50

【図1】両側予測技法を利用し得る例示的なコーディングシステムを示すブロック図である。

【図2】両側予測技法を実施し得る例示的なビデオエンコーダを示すブロック図である。

【図3】両側予測技法を実施し得るビデオデコーダの例を示すブロック図である。

【図4】参照ピクチャセット(RPS)のすべてのサブセットの中のエントリを伴う現在のピクチャを有するRPSを示す概略図である。

【図5】ビデオビットストリームの実施形態の概略図である。

【図6】コーディングされたビデオビットストリームを復号する方法の実施形態である。

【図7】ビデオビットストリームを符号化する方法の実施形態の図である。

【図8】コーディングされたビデオビットストリームを復号する方法の実施形態の図である。

10

【図9】ビデオコーディングデバイスの概略図である。

【図10】コーディングのための手段の実施形態の概略図である。

【発明を実施するための形態】

【0047】

以下は、本明細書において利用される様々な頭字語である。復号ピクチャバッファ(DPB)、瞬時復号リフレッシュ(IDR)、イントラランダムアクセスポイント(IRAP)、最下位ビット(LSB)、最上位ビット(MSB)、ネットワーク抽象化レイヤ(NAL)、ピクチャ順序カウンタ(POC)、ローバイトシーケンスペイロード(RBSP)、シーケンスパラメータセット(PPS)、およびワーキングドラフト(WD)。

20

【0048】

図1は、本明細書で説明されるようなビデオコーディング技法を利用し得る例示的なコーディングシステム10を示すブロック図である。図1に示されるように、コーディングシステム10は、デスティネーションデバイス14によって後で復号されることになる符号化されたビデオデータを提供するソースデバイス12を含む。具体的には、ソースデバイス12は、コンピュータ可読媒体16を介して、ビデオデータをデスティネーションデバイス14に提供し得る。ソースデバイス12およびデスティネーションデバイス14は、デスクトップコンピュータ、ノートブック(たとえば、ラップトップ)コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、いわゆる「スマート」パッド、テレビジョン、カメラ、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などを含む、広い範囲のデバイスのいずれかを備え得る。いくつかの場合、ソースデバイス12およびデスティネーションデバイス14は、ワイヤレス通信に対応し得る。

30

【0049】

デスティネーションデバイス14は、コンピュータ可読媒体16を介して復号されることになる、符号化されたビデオデータを受信し得る。コンピュータ可読媒体16は、符号化されたビデオデータをソースデバイス12からデスティネーションデバイス14に移動することが可能な、任意のタイプの媒体またはデバイスを備え得る。一例では、コンピュータ可読媒体16は、ソースデバイス12が符号化されたビデオデータをリアルタイムでデスティネーションデバイス14に直接送信することを可能にするための、通信媒体を備え得る。符号化されたビデオデータは、ワイヤレス通信プロトコルなどの通信規格に従って変調され、デスティネーションデバイス14に送信され得る。通信媒体は、無線周波数(RF)スペクトルまたは1つまたは複数の物理送信線などの、任意のワイヤレス通信媒体または有線通信媒体を備え得る。通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークなどの、パケットベースのネットワークの一部を形成し得る。通信媒体は、ソースデバイス12からデスティネーションデバイス14への通信を容易にするのに有用であり得る、ルータ、スイッチ、基地局、または任意の他の機器を含み得る。

40

【0050】

いくつかの例では、符号化されたデータは、出力インターフェース22から記憶デバイス

50

へ出力され得る。同様に、符号化されたデータは、入力インターフェースによって記憶デバイスからアクセスされ得る。記憶デバイスは、ハードドライブ、Blu-ray(登録商標)ディスク、デジタルビデオディスク(DVD)、コンパクトディスク読取り専用メモリ(CD-ROM)、フラッシュメモリ、揮発性もしくは不揮発性メモリ、または符号化されたビデオデータを記憶するための任意の他の適切なデジタル記憶媒体などの、多様な分散型データ記憶媒体またはローカルでアクセスされるデータ記憶媒体のいずれかを含み得る。さらなる例では、記憶デバイスは、ソースデバイス12によって生成される符号化されたビデオを記憶し得る、ファイルサーバまたは別の中間記憶デバイスに対応し得る。デスティネーションデバイス14は、ストリーミングまたはダウンロードを介して記憶デバイスからの記憶されているビデオデータにアクセスし得る。ファイルサーバは、符号化されたビデオデータを記憶することと、符号化されたビデオデータをデスティネーションデバイス14に送信することとが可能な、任意のタイプのサーバであり得る。例示的なファイルサーバは、ウェブサーバ(たとえば、ウェブサイトのための)、ファイル転送プロトコル(FTP)サーバ、ネットワークアタッチトストレージ(NAS)デバイス、またはローカルディスクドライブを含む。デスティネーションデバイス14は、インターネット接続を含む任意の標準的なデータ接続を通じて、符号化されたビデオデータにアクセスし得る。これは、ワイヤレスチャネル(たとえば、Wi-Fi接続)、有線接続(たとえば、デジタル加入者線(DSL)、ケーブルモデムなど)、または、ファイルサーバに記憶されている符号化されたビデオデータにアクセスするのに適した両方の組合せを含み得る。記憶デバイスからの符号化されたビデオデータの送信は、ストリーミング送信、ダウンロード送信、またはこれらの組合せであり得る。

10

20

【0051】

本開示の技法は、ワイヤレスの適用例または設定には必ずしも限定されない。この技法は、over-the-airテレビ放送、ケーブルテレビ送信、衛星テレビ送信、dynamic adaptive streaming over HTTP(DASH)などのインターネットストリーミングビデオ送信、データ記憶媒体に符号化されるデジタルビデオ、データ記憶媒体に記憶されるデジタルビデオの復号、または他の適用例などの、多様なマルチメディア適用例のいずれをサポートするビデオコーディングにも適用され得る。いくつかの例では、コーディングシステム10は、ビデオストリーミング、ビデオ再生、ビデオ放送、および/またはビデオ電話などの適用例をサポートするために、片方向または双方向のビデオ送信をサポートするように構成され得る。

30

【0052】

図1の例では、ソースデバイス12は、ビデオソース18、ビデオエンコーダ20、および出力インターフェース22を含む。デスティネーションデバイス14は、入力インターフェース28、ビデオデコーダ30、および表示デバイス32を含む。本開示によれば、ソースデバイス12のビデオエンコーダ20および/またはデスティネーションデバイス14のビデオデコーダ30は、ビデオコーディングのための技法を適用するように構成され得る。他の例では、ソースデバイスおよびデスティネーションデバイスは、他のコンポーネントまたは構成を含み得る。たとえば、ソースデバイス12は、外部カメラなどの外部のビデオソースからビデオデータを受信してもよい。同様に、デスティネーションデバイス14は、統合された表示デバイスを含むのではなく、外部の表示デバイスとインターフェースしてもよい。

40

【0053】

図1の示されるコーディングシステム10は一例にすぎない。ビデオコーディングのための技法は、任意のデジタルビデオ符号化および/または復号デバイスによって実行され得る。本開示の技法は全般にビデオコーディングデバイスによって実行されるが、この技法は、通常は「コーデック」と呼ばれる、ビデオエンコーダ/デコーダによっても実行され得る。その上、本開示の技法は、ビデオプリプロセッサによっても実行され得る。ビデオエンコーダおよび/またはデコーダは、グラフィクス処理装置(GPU)または同様のデバイスであり得る。

【0054】

ソースデバイス12およびデスティネーションデバイス14は、ソースデバイス12がデス

50

ディネーションデバイス14への送信のためにコーディングされたビデオデータを生成するような、コーディングデバイスの例にすぎない。いくつかの例では、ソースデバイス12およびデスティネーションデバイス14は、ソースデバイス12とデスティネーションデバイス14の各々がビデオ符号化コンポーネントとビデオ復号コンポーネントを含むように、実質的に対称的に動作してもよい。したがって、コーディングシステム10は、たとえばビデオストリーミング、ビデオ再生、ビデオブロードキャスト、またはビデオ電話のために、ビデオデバイス12と14との間の片方向または双方向のビデオ送信をサポートし得る。

【0055】

ソースデバイス12のビデオソース18は、ビデオカメラなどのビデオキャプチャデバイス、以前にキャプチャされたビデオを含むビデオアーカイブ、および/または、ビデオコンテンツプロバイダからビデオを受信するためのビデオフィードインターフェースを含み得る。さらなる代替として、ビデオソース18は、コンピュータグラフィクススペースのデータをソースビデオとして生成し、または、ライブビデオ、アーカイブされたビデオ、およびコンピュータで生成されたビデオの組合せを生成し得る。

10

【0056】

いくつかの場合、ビデオソース18がビデオカメラであるとき、ソースデバイス12およびデスティネーションデバイス14は、いわゆるカメラ電話またはビデオ電話を形成し得る。しかしながら、上で言及されたように、本開示において説明される技法は、ビデオコーディング全般に適用可能であってもよく、ワイヤレスおよび/または有線の適用例に適用されてもよい。各々の場合において、キャプチャされた、事前にキャプチャされた、またはコンピュータで生成されたビデオが、ビデオエンコーダ20によって符号化され得る。符号化されたビデオ情報は次いで、出力インターフェース22によってコンピュータ可読媒体16へと出力され得る。

20

【0057】

コンピュータ可読媒体16は、ワイヤレスブロードキャストもしくは有線ネットワーク送信などの一時的媒体、または、ハードディスク、フラッシュドライブ、コンパクトディスク、デジタルビデオディスク、Blu-ray(登録商標)ディスク、もしくは他のコンピュータ可読媒体などの記憶媒体(すなわち、非一時的記憶媒体)を含み得る。いくつかの例では、ネットワークサーバ(図示せず)は、ソースデバイス12から符号化されたビデオデータを受信し、たとえばネットワーク送信を介して、符号化されたビデオデータをデスティネーションデバイス14に提供し得る。同様に、ディスクスタンピング施設などの媒体生産施設のコンピューティングデバイスが、ソースデバイス12から符号化されたビデオデータを受信し、符号化されたビデオデータを含むディスクを生産してもよい。したがって、コンピュータ可読媒体16は、様々な例において、様々な形態の1つまたは複数のコンピュータ可読媒体を含むものとして理解され得る。

30

【0058】

デスティネーションデバイス14の入力インターフェース28は、コンピュータ可読媒体16から情報を受信する。コンピュータ可読媒体16の情報は、ビデオデコーダ30によっても使用される、ブロックおよび他のコーディングされたユニット、たとえばピクチャグループ(GOP)の特性ならびに/または処理を記述するシンタックス要素を含む、ビデオエンコーダ20により定義されるシンタックス情報を含み得る。表示デバイス32は、復号されたビデオデータをユーザに表示し、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)ディスプレイ、または別のタイプの表示デバイスなどの、多様な表示デバイスのいずれかを備え得る。

40

【0059】

ビデオエンコーダ20およびビデオデコーダ30は、現在開発中のHigh Efficiency Video Coding(HEVC)規格などの、ビデオコーディング規格に従って動作してもよく、HEVC Test Model(HM)に準拠してもよい。代替として、ビデオエンコーダ20およびビデオデコーダ30は、代替的にはMoving Picture Expert Group(MPEG)-4, Part 10, Advanced Video Coding(AVC)とも呼ばれる、国際電気通信連合電気通信規格化部門(ITU-T)H.264

50

規格、H.265/HEVC、またはそのような規格の拡張などの、他のプロプライエタリ規格または業界規格に従って動作し得る。しかしながら、本開示の技法は、どのような特定のコーディング規格にも限定されない。ビデオコーディング規格の他の例には、MPEG-2およびITU-T H.263がある。図1には示されていないが、いくつかの態様では、ビデオエンコーダ20およびビデオデコーダ30は各々、オーディオエンコーダおよびデコーダと統合されてもよく、共通のデータストリームおよび別々のデータストリームにおいてオーディオとデータの両方の符号化を扱うための、適切なマルチプレクサ-デマルチプレクサ(MUX-DEMUX)ユニット、または他のハードウェアおよびソフトウェアを含んでもよい。適用可能な場合、MUX-DEMUXユニットは、ITU H.223マルチプレクサプロトコル、またはユーザデータグラムプロトコル(UDP)などの他のプロトコルに準拠し得る。

10

【0060】

ビデオエンコーダ20およびビデオデコーダ30は各々、1つまたは複数のマイクロプロセッサ、デジタル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、ソフトウェア、ハードウェア、ファームウェア、またはこれらの任意の組合せなどの、多様な適切なエンコーダ回路のいずれとしても実装され得る。技法がソフトウェアで部分的に実装されるとき、デバイスは、適切な非一時的コンピュータ可読媒体にソフトウェアのための命令を記憶し、本開示の技法を実行するために1つまたは複数のプロセッサを使用してハードウェアで命令を実行し得る。ビデオエンコーダ20およびビデオデコーダ30の各々が、1つまたは複数のエンコーダまたはデコーダに含まれてもよく、それらのいずれもが、それぞれのデバイスにおいて合成エンコーダ/デコーダ(コーデック)の一部として統合されてもよい。ビデオエンコーダ20および/またはビデオデコーダ30を含むデバイスは、集積回路、マイクロプロセッサ、および/または携帯電話などのワイヤレス通信デバイスを備え得る。

20

【0061】

図2は、ビデオコーディング技法を実施し得るビデオエンコーダ20の例を示すブロック図である。ビデオエンコーダ20は、ビデオスライス内のビデオブロックのイントラコーディングおよびインターコーディングを実行し得る。イントラコーディングは、所与のビデオフレームまたはピクチャ内のビデオにおける空間的な冗長性を低減または除去するために、空間予測に依拠する。インターコーディングは、ビデオシーケンスの隣接するフレームまたはピクチャ内のビデオにおける時間的な冗長性を低減または除去するために、時間予測に依拠する。イントラモード(Iモード)は、いくつかの空間ベースのコーディングモードのいずれをも指し得る。単方向(単予測としても知られている)予測(Pモード)または双予測(bi-prediction)(双予測(bi prediction)としても知られている)(Bモード)などのインターモードは、いくつかの時間ベースのコーディングモードのいずれをも指し得る。

30

【0062】

図2に示されるように、ビデオエンコーダ20は、符号化されるべきビデオフレーム内の現在のビデオブロックを受信する。図2の例では、ビデオエンコーダ20は、モード選択ユニット40、参照フレームメモリ64、加算器50、変換処理ユニット52、量子化ユニット54、およびエントロピーコーディングユニット56を含む。そして、モード選択ユニット40は、動き補償ユニット44、動き推定ユニット42、イントラ予測(intra-prediction)(イントラ予測(intra prediction)としても知られている)ユニット46、および区分ユニット48を含む。ビデオブロック再構築のために、ビデオエンコーダ20は、逆量子化ユニット58、逆変換ユニット60、および加算器62も含む。デブロッキングフィルタ(図2に示されない)も、ブロック境界をフィルタリングして再構築されたビデオからブロッキングネスアーティファクトを除去するために含まれ得る。望まれる場合、デブロッキングフィルタは通常、加算器62の出力をフィルタリングする。追加のフィルタ(ループ内またはループ後の)も、デブロッキングフィルタに加えて使用され得る。簡潔にするためにそのようなフィルタは示されていないが、望まれる場合、加算器50の出力をフィルタリングし得る(ループ内フィルタとして)。

40

【0063】

50

符号化プロセスの間、ビデオエンコーダ20は、コーディングされることになるビデオフレームまたはスライスを受信する。フレームまたはスライスは複数のビデオブロックに分割され得る。動き推定ユニット42および動き補償ユニット44は、1つまたは複数の参照フレームの中の1つまたは複数のブロックに対して相対的な、受信されたビデオブロックのインタラ予測コーディングを実行して、時間予測を行う。イントラ予測ユニット46は代替として、コーディングされることになるブロックと同じフレームまたはスライスの中にある1つまたは複数の近隣ブロックに対して相対的な、受信されたビデオブロックのイントラ予測コーディングを実行して、空間予測を行い得る。ビデオエンコーダ20は、たとえばビデオデータの各ブロックに対する適切なコーディングモードを選択するために、複数のコーディングパスを実行し得る。

10

【0064】

その上、区分ユニット48は、以前のコーディングパスにおける以前の区分方式の評価に基づいて、ビデオデータのブロックをサブブロックへと区分し得る。たとえば、区分ユニット48は、最初にフレームまたはスライスを最大コーディングユニット(LCU)へと区分し、レート歪み分析(たとえば、レート歪み最適化)に基づいて、LCUの各々をサブコーディングユニット(sub-CU)へと区分し得る。モード選択ユニット40はさらに、サブCUへのLCUの区分を示す四分木データ構造を生み出し得る。四分木のリーフノードCUは、1つまたは複数の予測ユニット(PU)および1つまたは複数の変換ユニット(TU)を含み得る。

【0065】

本開示は、HEVCの文脈におけるCU、PU、もしくはTU、または、他の規格の文脈における類似のデータ構造(たとえば、H.264/AVCにおけるマクロブロックおよびマクロブロックのサブブロック)のいずれかを指すために、「ブロック」という用語を使用する。CUは、コーディングノード、PU、およびコーディングノードと関連付けられるTUを含む。CUのサイズは、コーディングノードのサイズに対応し、形状が正方形である。CUのサイズは、8×8ピクセルから、最大で64×64ピクセル以上のツリーブロックのサイズにまでわたり得る。各CUは、1つまたは複数のPUおよび1つまたは複数のTUを含み得る。CUと関連付けられるシンタックスデータは、たとえば、1つまたは複数のPUへのCUの区分を記述し得る。区分モードは、CUがスキップモードもしくはダイレクトモードで符号化されるか、イントラ予測モードで符号化されるか、またはインタラ予測(inter-prediction)(インタラ予測(inter prediction)としても知られている)モードで符号化されるかで異なり得る。PUは形状が非正方形となるように区分され得る。CUと関連付けられるシンタックスデータはまた、たとえば、四分木に従った1つまたは複数のTUへのCUの区分を記述し得る。TUは、形状が正方形または非正方形(たとえば、長方形)であり得る。

20

30

【0066】

モード選択ユニット40は、たとえば誤差の結果に基づいて、コーディングモード、すなわちイントラコーディングモードまたはインタラコーディングモードのうちの1つを選択し、得られたイントラコーディングされたブロックまたはインタラコーディングされたブロックを加算器50に提供して残差ブロックデータを生成し、加算器62に提供して参照フレームとして使用するための符号化されたブロックを再構築する。モード選択ユニット40はまた、動きベクトル、イントラモードインジケータ、区分情報、および他のそのようなシンタックス情報などのシンタックス要素を、エントロピーコーディングユニット56に提供する。

40

【0067】

動き推定ユニット42および動き補償ユニット44は高度に統合され得るが、概念上の目的で別々に示されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックの動きを推定する動きベクトルを生成するプロセスである。動きベクトルは、たとえば、現在のフレーム(または他のコーディングされるユニット)内でコーディングされている現在のブロックに対する参照フレーム(または他のコーディングされるユニット)内の予測ブロックに対する現在のビデオフレームまたはピクチャ内のビデオブロックのPUのずれを示し得る。予測ブロックは、ピクセル差分に関して、コーディングされるべきプロ

50

ックとよく一致することが見いだされるブロックであり、これは、絶対差分和(SAD)、平方差分和(SSD)、または他の差分尺度によって決定され得る。いくつかの例では、ビデオエンコーダ20は、参照フレームメモリ64に記憶されている参照ピクチャのサブ整数ピクセル位置の値を計算し得る。たとえば、ビデオエンコーダ20は、4分の1ピクセル位置、8分の1ピクセル位置、または参照ピクチャの他の分数ピクセル位置の値を補間し得る。したがって、動き推定ユニット42は、全ピクセル位置および分数ピクセル位置に対して動き探索を実行し、分数ピクセル精度で動きベクトルを出力し得る。

【0068】

動き推定ユニット42は、PUの位置を参照ピクチャの予測ブロックの位置と比較することによって、インターコーディングされたスライスの中のビデオブロックのPUに対する動きベクトルを計算する。参照ピクチャは、第1の参照ピクチャリスト(リスト0)または第2の参照ピクチャリスト(リスト1)から選択されてもよく、それらの各々が、参照フレームメモリ64に記憶されている1つまたは複数の参照ピクチャを特定する。動き推定ユニット42は、計算された動きベクトルをエントロピー符号化ユニット56および動き補償ユニット44に送信する。

10

【0069】

動き補償ユニット44によって実行される動き補償は、動き推定ユニット42によって決定される動き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を伴い得る。やはり、いくつかの例では、動き推定ユニット42および動き補償ユニット44は機能的に統合され得る。現在のビデオブロックのPUの動きベクトルを受信すると、動き補償ユニット44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し示す予測ブロックを見つけ得る。以下で論じられるように、加算器50は、ピクセル差分値を形成する、コーディングされている現在のビデオブロックのピクセル値から、予測ブロックのピクセル値を差し引くことによって、残差ビデオブロックを形成する。一般に、動き推定ユニット42は、ルマ成分に対する動き推定を実行し、動き補償ユニット44は、クロマ成分とルマ成分の両方に対して、ルマ成分に基づいて計算された動きベクトルを使用する。モード選択ユニット40はまた、ビデオスライスのビデオ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、ビデオブロックおよびビデオスライスと関連付けられるシンタックス要素を生成し得る。

20

【0070】

上で説明されたように、イントラ予測ユニット46は、動き推定ユニット42および動き補償ユニット44によって実行されるインター予測の代替として、現在のブロックをイントラ予測し得る。具体的には、イントラ予測ユニット46は、現在のブロックを符号化するために使用すべきイントラ予測モードを決定し得る。いくつかの例では、イントラ予測ユニット46は、たとえば、別々の符号化パスの間に、様々なイントラ予測モードを使用して現在のブロックを符号化してもよく、イントラ予測ユニット46(または、いくつかの例ではモード選択ユニット40)は、試験されたモードから使用すべき適切なイントラ予測モードを選択してもよい。

30

【0071】

たとえば、イントラ予測ユニット46は、様々な試験されるイントラ予測モードに対するレート歪み分析を使用してレート歪み値を計算し、試験されたモードの中から最良のレート歪み特性を有するイントラ予測モードを選択し得る。レート歪み分析は一般に、符号化されたブロックと、符号化されたブロックを生み出すために符号化された元の符号化されていないブロックとの間の歪み(または誤差)の量、ならびに、符号化されたブロックを生み出すために使用されたビットレート(すなわち、ビットの数)を決定する。イントラ予測ユニット46は、どのイントラ予測モードがブロックについて最良のレート歪み値を呈するかを決定するために、様々な符号化されたブロックに対する歪みおよびレートから比を計算し得る。

40

【0072】

加えて、イントラ予測ユニット46は、深度モデリングモード(DMM)を使用して深度マ

50

ップの深度ブロックをコーディングするように構成され得る。モード選択ユニット40は、たとえばレート歪み最適化(RDO)を使用して、利用可能なDMMモードがイントラ予測モードおよび他のDMMモードより良いコーディング結果を生み出すかどうかを決定し得る。深度マップに対応するテクスチャ画像に対するデータは、参照フレームメモリ64に記憶され得る。動き推定ユニット42および動き補償ユニット44はまた、深度マップの深度ブロックをインター予測するように構成され得る。

【0073】

ブロックに対するイントラ予測モード(たとえば、従来のイントラ予測モードまたはDMMモードのうちの1つ)を選択した後、イントラ予測ユニット46は、ブロックに対する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コーディングユニット56に提供し得る。エントロピーコーディングユニット56は、選択されたイントラ予測モードを示す情報を符号化し得る。ビデオエンコーダ20は、複数のイントラ予測モードインデックステーブルおよび複数の修正されたイントラ予測モードインデックステーブル(符号語マッピングテーブルとも呼ばれる)を含み得る、送信されるビットストリーム構成データに、様々なブロックに対する符号化コンテキストの定義、最も確率の高いイントラ予測モードの指示、イントラ予測モードインデックステーブル、および、コンテキストの各々に対して使用すべき修正されたイントラ予測モードインデックステーブルを含め得る。

10

【0074】

ビデオエンコーダ20は、コーディングされている元のビデオブロックから、モード選択ユニット40からの予測データを差し引くことによって、残差ビデオブロックを形成する。加算器50は、この減算演算を実行する1つまたは複数のコンポーネントを表す。

20

【0075】

変換処理ユニット52は、離散コサイン変換(DCT)または概念的に同様の変換などの変換を残差ブロックに適用して、残差変換係数値を備えるビデオブロックを生み出す。変換処理ユニット52は、DCTと概念的に同様の他の変換を実行し得る。ウェーブレット変換、整数変換、サブバンド変換、または他のタイプの変換も使用され得る。

【0076】

変換処理ユニット52は、残差ブロックに変換を適用し、残差変換係数のブロックを生み出す。この変換は、ピクセル値領域から、周波数領域などの変換領域に、残差情報を変換し得る。変換処理ユニット52は、得られた変換係数を量子化ユニット54に送信し得る。量子化ユニット54は、変換係数を量子化してビットレートをさらに低減する。量子化プロセスは、係数の一部またはすべてと関連付けられるビット深度を低減し得る。量子化の程度は、量子化パラメータを調整することによって修正され得る。いくつかの例では、量子化ユニット54は次いで、量子化された変換係数を含む行列の走査を実行し得る。代替的に、エントロピー符号化ユニット56が走査を実行してもよい。

30

【0077】

量子化に続いて、エントロピーコーディングユニット56が、量子化された変換係数をコーディングする。たとえば、エントロピーコーディングユニット56は、コンテキスト適応可変長コーディング(CALVC)、コンテキスト適応バイナリ算術コーディング(CABAC)、シンタックススペースコンテキスト適応バイナリ算術コーディング(SBAC)、確率間隔区分エントロピー(PIPE)コーディング、または別のエントロピーコーディング技法を実行し得る。コンテキストベースのエントロピーコーディングの場合、コンテキストは近隣ブロックに基づき得る。エントロピーコーディングユニット56によるエントロピーコーディングに続いて、符号化されたビットストリームが、別のデバイス(たとえば、ビデオデコーダ30)に送信され、または、より後の送信もしくは取り出しのために保管され得る。

40

【0078】

逆量子化ユニット58および逆変換ユニット60は、逆量子化および逆変換をそれぞれ適用して、たとえば参照ブロックとして後で使用するために、ピクセル領域において残差ブロックを再構築する。動き補償ユニット44は、参照フレームメモリ64のフレームのうちの1つの予測ブロックに残差ブロックを追加することによって、参照ブロックを計算し得

50

る。動き補償ユニット44はまた、1つまたは複数の補間フィルタを再構築された残差ブロックに適用して、動き推定において使用するためのサブ整数ピクセル値を計算し得る。加算器62は、再構築された残差ブロックを動き補償ユニット44によって生み出される動き補償された予測ブロックに加算して、参照フレームメモリ64に記憶するための再構築されたビデオブロックを生み出す。再構築されたビデオブロックは、後続のビデオフレームの中のブロックをインターコーディングするために、参照ブロックとして動き推定ユニット42および動き補償ユニット44によって使用され得る。

【0079】

図3は、ビデオコーディング技法を実施し得るビデオデコーダ30の例を示すブロック図である。図3の例では、ビデオデコーダ30は、エン트로ピー復号ユニット70、動き補償ユニット72、イントラ予測ユニット74、逆量子化ユニット76、逆変換ユニット78、参照フレームメモリ82、および加算器80を含む。ビデオデコーダ30は、いくつかの例では、ビデオエンコーダ20(図2)に関して説明された符号化パスと全般に逆の復号パスを実行する。動き補償ユニット72は、エン트로ピー復号ユニット70から受信された動きベクトルに基づいて予測データを生成し得るが、イントラ予測ユニット74は、エン트로ピー復号ユニット70から受信されたイントラ予測モードインジケータに基づいて予測データを生成し得る。

【0080】

復号プロセスの間、ビデオデコーダ30は、符号化されたビデオスライスのビデオブロックおよび関連するシンタックス要素を表す符号化されたビデオビットストリームを、ビデオエンコーダ20から受信する。ビデオデコーダ30のエン트로ピー復号ユニット70は、ビットストリームをエン트로ピー復号して、量子化された係数、動きベクトルまたはイントラ予測モードインジケータ、および他のシンタックス要素を生成する。エン트로ピー復号ユニット70は、動きベクトルおよび他のシンタックス要素を動き補償ユニット72に転送する。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でシンタックス要素を受信し得る。

【0081】

ビデオスライスがイントラコーディングされる(I)スライスとしてコーディングされる時、イントラ予測ユニット74は、シグナリングされたイントラ予測モードと、現在のフレームまたはピクチャの以前に復号されたブロックからのデータとに基づいて、現在のビデオスライスのビデオブロックに対する予測データを生成し得る。ビデオフレームがインターコーディングされる(たとえば、B、P、またはGPB)スライスとしてコーディングされる時、動き補償ユニット72は、エン트로ピー復号ユニット70から受信された動きベクトルおよび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に対する予測ブロックを生み出す。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つから生み出され得る。ビデオデコーダ30は、参照フレームメモリ82に記憶されている参照ピクチャに基づく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。

【0082】

動き補償ユニット72は、動きベクトル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に対する予測情報を決定し、予測情報を使用して、復号されている現在のビデオブロックに対する予測ブロックを生み出す。たとえば、動き補償ユニット72は、受信されたシンタックス要素のいくつかを使用して、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つまたは複数に対する構築情報、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に対する動きベクトル、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に対する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。

10

20

30

40

50

【 0 0 8 3 】

動き補償ユニット72はまた、補間フィルタに基づいて補間を実行し得る。動き補償ユニット72は、ビデオブロックの符号化の間にビデオエンコーダ20によって使用されるような補間フィルタを使用して、参照ブロックのサブ整数ピクセルに対する補間された値を計算し得る。この場合、動き補償ユニット72は、受信されたシンタックス要素からビデオエンコーダ20によって使用される補間フィルタを決定し、補間フィルタを使用して予測ブロックを生み出し得る。

【 0 0 8 4 】

深度マップに対応するテクスチャ画像に対するデータは、参照フレームメモリ82に記憶され得る。動き補償ユニット72はまた、深度マップの深度ブロックをインター予測するように構成され得る。

10

【 0 0 8 5 】

画像およびビデオ圧縮は急速な成長をとげており、様々なコーディング規格につながった。そのようなビデオコーディング規格には、International Telecommunication Union (ITU) Telecommunications Standardization Sector (ITU-T) H.261、ISO/IEC Motion Picture Experts Group (MPEG)-1 Part 2、ITU-T H.262またはISO/IEC MPEG-2 Part 2、ITU-T H.263、ITU-T H.264またはISO/IEC MPEG-4 Part 10としても知られているISO/IEC MPEG-4 Part 2、Advanced Video Coding (AVC)、および、ITU-T H.265またはMPEG-H Part 2としても知られているHigh Efficiency Video Coding (HEVC)がある。AVCは、Scalable Video Coding(SVC)、Multiview Video Coding (MVC)、およびMultiview Video Coding plus Depth(MVC+D)、ならびに3D AVC(3D-AVC)などの拡張を含む。HEVCは、Scalable HEVC(SHVC)、Multiview HEVC(MV-HEVC)、および3D HEVC(3D-HEVC)などの拡張を含む。

20

【 0 0 8 6 】

Versatile Video Coding(VVC)は、ITU-TおよびISO/IECのjoint video expert team(JVET)によって開発中の新しいビデオコーディング規格である。執筆の時点で、VVCの最新のワーキングドラフト(WD)は、JVET-K1001-v1に含まれた。JVET文書JVET-K0325-v3は、VVCの高水準シンタックスに対する更新を含む。

【 0 0 8 7 】

本開示は、VVC規格の開発中に対処することを意図した技法を説明する。しかしながら、この技法はまた、他のビデオ/メディアコーデック仕様にも適用され得る。

30

【 0 0 8 8 】

ビデオ圧縮技法は、ビデオシーケンスに内在する冗長性を低減または除去するために、空間(イントラピクチャ)予測および/または時間(インターピクチャ)予測を実行する。ブロックベースのビデオコーディングでは、ビデオスライス(たとえば、ビデオピクチャまたはビデオピクチャの一部)がビデオブロックへと区分されてもよく、このビデオブロックは、ツリーブロック、コーディングツリーブロック(CTB)、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)、および/またはコーディングノードとも呼ばれ得る。ピクチャのイントラコーディングされた(I)スライスの中のビデオブロックは、同じピクチャの中の近隣ブロックの中の参照サンプルに関する空間予測を使用して符号化される。ピクチャのインターコーディングされた(PまたはB)スライスの中のビデオブロックは、同じピクチャの中の近隣ブロックの中の参照サンプルに関する空間予測、または他の参照ピクチャの中の参照サンプルに関する時間予測を使用し得る。ピクチャはフレームと呼ばれることがあり、参照ピクチャは参照フレームと呼ばれることがある。

40

【 0 0 8 9 】

空間予測または時間予測は、コーディングされることになるブロックに対する予測ブロックをもたらす。残差データは、コーディングされることになる元のブロックと予測ブロックとの間のピクセル差分を表す。インターコーディングされたブロックは、予測ブロックを形成する参照サンプルのブロックを指し示す動きベクトルと、コーディングされたブロックと予測ブロックとの差を示す残差データとに従って符号化される。イントラコーデ

50

イングされたブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データはピクセル領域から変換領域に変換されてもよく、残差変換係数をもたらし、これは次いで量子化され得る。最初に2次元アレイに並べられる量子化された変換係数は、変換係数の1次元ベクトルを生み出すために走査されてもよく、さらなる圧縮を達成するために、エントロピーコーディングが適用されてもよい。

【0090】

ビデオコーデック仕様において、ピクチャは、インター予測における参照ピクチャとしての使用、復号ピクチャバッファ(DPB)からのピクチャの出力、動きベクトルのスケーリング、加重予測などを含めた、複数の目的で特定される。AVCおよびHEVCにおいて、ピクチャはピクチャ順序カウンタ(POC)によって特定され得る。AVCおよびHEVCにおいて、DPBの中のピクチャは、「短期参照のために使用される」、「長期参照のために使用される」、または「参照のために使用されない」としてマークされ得る。ピクチャが「参照のために使用されない」とマークされると、もはや予測のために使用することはできず、出力にもはや必要とされないとき、DPBから削除することができる。

10

【0091】

AVCにおいて、短期および長期という2つのタイプの参照ピクチャがある。参照ピクチャは、予測参照のためにもはや必要とされなくなると、「参照のために使用されない」とマークされ得る。これらの3つのステータス(短期、長期、および参照のために使用されない)の変換は、復号参照ピクチャマーキングプロセスによって制御される。暗黙的スライディングウィンドウプロセスおよび明示的メモリ管理制御操作(MMCO)プロセスという、2つの代替的な復号参照ピクチャマーキング機構がある。スライディングウィンドウプロセスは、参照フレームの数が所与の最大の数(SPSにおけるmax_num_ref_frames)に等しいとき、「参照のために使用されない」と短期参照ピクチャをマークする。短期参照ピクチャは、直近の復号された短期ピクチャがDPBに保持されるように、first-in、first-out方式で記憶される。

20

【0092】

明示的MMCOプロセスは、複数のMMCOコマンドを含み得る。MMCOコマンドは、1つまたは複数の短期または長期参照ピクチャを「参照のために使用されない」とマークしてもよく、すべてのピクチャを「参照のために使用されない」とマークしてもよく、または、現在の参照ピクチャもしくは既存の短期参照ピクチャを長期としてマークして、長期ピクチャインデックスをその長期参照ピクチャに割り当ててもよい。

30

【0093】

AVCにおいて、参照ピクチャマーキング操作、ならびにDPBからのピクチャの出力および削除のためのプロセスは、ピクチャが復号された後に実行される。

【0094】

HEVCは、参照ピクチャセット(RPS)と呼ばれる、参照ピクチャ管理のための異なる手法を導入する。AVCのMMCO/スライディングウィンドウと比較したときの、RPSの概念についての最も基本的な違いは、各々の特定のスライスに対して、現在のピクチャまたは任意の後続のピクチャによって使用される参照ピクチャの完全なセットが提供されるということである。したがって、現在または未来のピクチャによる使用のためにDPBに保持されなければならないすべてのピクチャの完全なセットがシグナリングされる。これは、DPBに対する相対的な変化のみがシグナリングされるAVC方式とは異なる。RPSの概念により、DPBに参照ピクチャの正しいステータスを維持するために、復号順序においてより早いピクチャからの情報は必要とされない。

40

【0095】

RPSの利点を活用して誤り耐性を高めるために、HEVCにおけるピクチャ復号とDPB操作の順序は、AVCと比較して変更される。AVCでは、ピクチャマーキングおよびバッファ操作(DPBからの復号されたピクチャの出力と削除の両方)は一般に、現在のピクチャが復号された後に適用される。HEVCでは、RPSがまず現在のピクチャのスライスヘッダから

50

復号され、次いで、ピクチャマーキングおよびバッファ操作が一般に現在のピクチャを復号する前に適用される。

【0096】

HEVCにおける各スライスヘッダは、スライスを含むピクチャに対するRPSのシグナリングのためのパラメータを含まなければならない。唯一の例外は、IDRスライスに対してはRPSがシグナリングされないことである。代わりに、RPSは空であると推測される。IDRピクチャに属しないスライスに対して、復号順序においてIピクチャの前にあったピクチャからのインター予測を使用する、復号順序においてIピクチャの後にあるピクチャがあり得るので、IスライスがIピクチャに属する場合であっても、RPSが提供されることがある。RPSの中のピクチャの数は、SPSにおけるsps_max_dec_pic_bufferingシNTAX要素によって指定されるようなDPBサイズ限界を超えないものとする。

10

【0097】

各ピクチャは、出力順序を表すPOC値と関連付けられる。スライスヘッダは、完全なPOC値の最下位ビットを表す固定長の符号語、pic_order_cnt_lsbを含み、これはPOC LSBとしても知られている。符号語の長さはSPSにおいてシグナリングされ、4ビットと16ビットの間であり得る。RPSの概念は、参照ピクチャを特定するためにPOCを使用する。それ自体のPOC値の他に、各スライスヘッダは、RPSの中の各ピクチャのPOC値(またはLSB)のコーディングされた表現を直接含み、またはSPSから継承する。

【0098】

各ピクチャに対するRPSは、5つのRPSサブセットとも呼ばれる、参照ピクチャの5つの異なるセットを含む。RefPicSetStCurrBeforeは、復号順序と出力順序の両方で現在のピクチャより前にあり、かつ現在のピクチャのインター予測において使用され得る、すべての短期参照ピクチャを含む。RefPicSetStCurrAfterは、復号順序において現在のピクチャの前にあり、出力順序において現在のピクチャの後であり、かつ現在のピクチャのインター予測において使用され得る、すべての短期参照ピクチャを含む。RefPicSetStFollは、復号順序において現在のピクチャの後にあるピクチャのうちの1つまたは複数のインター予測において使用され得る、かつ現在のピクチャのインター予測において使用されない、すべての短期参照ピクチャを含む。RefPicSetLtCurrは、現在のピクチャのインター予測において使用され得るすべての長期参照ピクチャを含む。RefPicSetLtFollは、復号順序において現在のピクチャの後にあるピクチャのうちの1つまたは複数のインター予測において使用され得る、かつ現在のピクチャのインター予測において使用されない、すべての長期参照ピクチャを含む。

20

30

【0099】

RPSは、現在のピクチャよりPOC値の低い短期参照ピクチャ、現在のピクチャよりPOC値の高い短期参照ピクチャ、および長期参照ピクチャという、異なるタイプの参照ピクチャにわたって反復する最大で3つのループを使用してシグナリングされる。加えて、参照ピクチャが現在のピクチャによる参照のために使用される(リストRefPicSetStCurrBefore、RefPicSetStCurrAfter、またはRefPicSetLtCurrのうちの1つに含まれる)か、またはされない(リストRefPicSetStFollまたはRefPicSetLtFollのうちの1つに含まれる)かを示すフラグ(used_by_curr_pic_X_flag)が、各参照ピクチャに対して送信される。

40

【0100】

図4は、RPS400のすべてのサブセット402の中のエン트리(たとえば、ピクチャ)を伴う現在のピクチャB14を有するRPS400を示す。図4の例では、現在のピクチャB14は、5つのサブセット402(RPSサブセットとしても知られている)の各々の中の厳密に1つのピクチャを含む。P8は、RefPicSetStCurrBeforeと呼ばれるサブセット402の中のピクチャであり、それは、ピクチャが出力順序において前にありB14によって使用されるからである。P12は、RefPicSetStCurrAfterと呼ばれるサブセット402の中のピクチャであり、それは、ピクチャが出力順序において後にありB14によって使用されるからである。P13は、RefPicSetStFollと呼ばれるサブセット402の中のピクチャであり、それは、ピクチャがB14によって使用されない(しかし、B15によって使用されるのでDPBに保持されなければ

50

ばならない)短期参照ピクチャであるからである。P4は、RefPicSetLtCurrと呼ばれるサブセット402の中のピクチャであり、それは、ピクチャがB14によって使用される長期参照ピクチャであるからである。I0は、RefPicSetStLtFolIと呼ばれるサブセット402の中のピクチャであり、それは、ピクチャが現在のピクチャによって使用されない(しかし、B15によって使用されるのでDPBに保持されなければならない)長期参照ピクチャであるからである。

【0101】

RPS400の短期部分は、スライスヘッダに直接含まれ得る。代替的に、スライスヘッダは、アクティブなSPSにおいて送信されるRPSのあらかじめ定められたリストを参照するインデックスを表すシンタックス要素のみを含み得る。RPS402の短期部分は、以下で説明されるようなインターRPSと、ここで説明されるようなイントラRPSという、2つの異なる方式のいずれかを使用してシグナリングされ得る。イントラRPSが使用されるとき、参照ピクチャの2つの異なるリストの長さを表すnum_negative_picsおよびnum_positive_picsがシグナリングされる。これらのリストは、それぞれ、現在のピクチャと比較して負のPOC差分および正のPOC差分を伴う参照ピクチャを含む。これらのリストの中の各要素は、リストの中の前の要素に対する相対的なPOC値の差から1を引いたものを表す可変長符号を用いて符号化される。

10

【0102】

シーケンスパラメータセットにおいて再発するRPSを符号化するとき、シーケンスパラメータセットにおいてすでに符号化されている別のRPSに関連して1つのRPS(たとえば、RPS400)の要素を符号化することが可能である。これはインターRPSと呼ばれる。シーケンスパラメータセットのすべてのRPSは同じネットワーク抽象化レイヤ(NAL)ユニットの中にあるので、この方法と関連付けられる誤り耐性問題はない。インターRPSシンタックスは、現在のピクチャのRPSが以前に復号されたピクチャのRPSから予測され得るという事実を利用する。これは、現在のピクチャのすべての参照ピクチャが以前のピクチャの参照ピクチャまたは以前に復号されたピクチャ自体のいずれかでなければならないからである。これらのピクチャのいずれが参照ピクチャであるべきであるか、および現在のピクチャの予測のために使用されるべきであることを示すことだけが必要である。したがって、シンタックスは、予測子として使用するためのRPSを指し示すインデックス、現在のRPSのdelta_POCを取得するために予測子のdelta_POCに加算されるべきdelta_POC、ならびに、どのピクチャが参照ピクチャであるか、およびそれらが未来のピクチャの予測のためだけに使用されるかどうかを示すためのインジケータのセットを備える。ある実施形態では、デルタPOCは、現在の参照ピクチャと別の(たとえば、以前の)参照ピクチャとの間でのPOC値の差を指す。

20

30

【0103】

長期参照ピクチャの使用を活用することを望むエンコーダは、SPSシンタックス要素long_term_ref_pics_present_flagを1に設定しなければならない。長期参照ピクチャは次いで、各長期ピクチャの完全なPOC値の最下位ビットを表す固定長の符号語pos_lsb_ltによってスライスヘッダにおいてシグナリングされ得る。各pos_lsb_ltは、特定の長期ピクチャのためにシグナリングされたpic_order_cnt_lsb符号語のコピーである。SPSの中の長期ピクチャのセットをPOC LSB値のリストとしてシグナリングすることも可能である。長期ピクチャに対するPOC LSBは、このリストに対するインデックスとしてスライスヘッダにおいてシグナリングされ得る。

40

【0104】

現在のピクチャに対する長期参照ピクチャの完全なPOC距離の計算を可能にするために、delta_poc_msb_cycle_lt_minus1シンタックス要素が追加でシグナリングされ得る。符号語delta_poc_msb_cycle_lt_minus1は、RPSの中の任意の他の参照ピクチャと同じPOC LSB値を有する各長期参照ピクチャに対してシグナリングされることが必要とされる。

【0105】

50

HEVCにおける参照ピクチャマーキングのために、通常は、ピクチャ復号の前にDPBの中にある数のピクチャが存在する。それらの一部は、予測に利用可能であり得るので、「参照のために使用される」とマークされる。その他は予測に利用不可能であり得るが出力を待機しているので、「参照のために使用されない」とマークされる。スライスヘッダが解析されると、スライスデータが復号される前に、ピクチャマーキングプロセスが行われる。DPBの中に存在し「参照のために使用される」とマークされるがRPSに含まれないピクチャは「参照のために使用されない」とマークされる。DPBの中に存在しないが参照ピクチャセットに含まれるピクチャは、used_by_curr_pic_X_flagが0に等しい場合は無視される。しかしながら、代わりにused_by_curr_pic_X_flagが1に等しい場合、この参照ピクチャは、現在のピクチャにおける予測のために使用されることが意図されていたが、失われている。そして、意図しないピクチャの喪失が推測され、デコーダは適切な行動をとらなければならない。

10

【0106】

現在のピクチャを復号した後、それは「短期参照のために使用される」とマークされる。

【0107】

次に、HEVCにおける参照ピクチャリスト構築が論じられる。HEVCでは、現在の復号されたピクチャ以外の参照ピクチャのデータ要素(たとえば、サンプル値または動きベクトル)から導出される予測を示すために、インター予測という用語が使用される。AVCのように、ピクチャは複数の参照ピクチャから予測され得る。インター予測のために使用される参照ピクチャは、1つまたは複数の参照ピクチャリストにおいて編成される。参照インデックスは、リストの中の参照ピクチャのいずれが予測信号を作成するために使用されるべきかを特定する。

20

【0108】

単一の参照ピクチャリスト、リスト0がPスライスのために使用され、2つの参照ピクチャリスト、リスト0およびリスト1がBスライスのために使用される。AVCと同様に、HEVCにおける参照ピクチャリスト構築は、参照ピクチャリスト初期化および参照ピクチャリスト修正を含む。

【0109】

AVCでは、リスト0に対する初期化プロセスは、(復号順序が使用される)Pスライスおよび(出力順序が使用される)Bスライスに対しては異なる。HEVCでは、出力順序は両方の場合において使用される。

30

【0110】

参照ピクチャリスト初期化は、3つのRPSサブセット、RefPicSetStCurrBefore、RefPicSetStCurrAfter、およびRefPicSetLtCurrに基づいて、デフォルトのリスト0およびリスト1(スライスがBスライスである場合)を作成する。出力順序がより前(後)の短期ピクチャがまず、現在のピクチャまでのPOC距離の昇順でリスト0(リスト1)に挿入され、次いで、出力順序がより後(前)の短期ピクチャが、現在のピクチャまでのPOC距離の昇順でリスト0(リスト1)に挿入され、最後に、長期ピクチャが末尾に挿入される。RPSに関して、リスト0に対して、RefPicSetStCurrBeforeの中のエンタリは初期のリストに挿入され、RefPicSetStCurrAfterの中のエンタリが後に続く。その後、利用可能であれば、RefPicSetLtCurrの中のエンタリが付加される。

40

【0111】

HEVCにおいて、リストの中のエンタリの数がアクティブな参照ピクチャの目標の数(ピクチャパラメータセットまたはスライスヘッダにおいてシグナリングされる)より少ないとき、上記のプロセスが繰り返される(参照ピクチャリストにすでに追加されている参照ピクチャが再び追加される)。エンタリの数が目標の数より多いとき、リストは切り詰められる。

【0112】

参照ピクチャリストが初期化された後で、参照ピクチャリスト修正コマンドに基づいて、1つの特定の参照ピクチャがリストの中の1つより多くの位置において現れ得る事例を含

50

めて、現在のピクチャに対する参照ピクチャが任意の順序で並べられ得るように、参照ピクチャリストが修正され得る。リスト修正の存在を示すフラグが1に設定されるとき、固定された数(参照ピクチャリストの中のエントリの目標の数に等しい)のコマンドがシグナリングされ、各コマンドが参照ピクチャリストに対して1つのエントリを挿入する。参照ピクチャは、RPSシグナリングから導出される現在のピクチャに対する参照ピクチャのリストへのインデックスによって、コマンドにおいて特定される。これは、H.264/AVCにおける参照ピクチャリスト修正とは異なり、ピクチャはピクチャ番号(frame_numシンタックス要素から導出される)または長期参照ピクチャインデックスのいずれかによって特定され、たとえば、初期リストの最初の2つのエントリを交換するために、または、初期リストの最初に1つのエントリを挿入して他方のエントリをシフトするために、より少数のコマンドしか必要とされないことがあり得る。

10

【0113】

参照ピクチャリストは、現在のピクチャより大きいTemporalIdを伴う任意の参照ピクチャを含むことが許容される。HEVCビットストリームは、いくつかの時間的なサブレイヤからなり得る。各NALユニットは、TemporalId(temporal_id_plus1-1に等しい)によって示されるような特定のサブレイヤに属する。

【0114】

参照ピクチャ管理は、参照ピクチャリストに直接基づく。JCT-VC文書JCTVC-G643は、DPBの中の参照ピクチャの管理のために、参照ピクチャリスト0、参照ピクチャリスト1、ならびにアイドル参照ピクチャリストという3つの参照ピクチャリストを直接使用するための手法を含み、それにより、1)スライディングウィンドウおよびMMCOプロセス、ならびにAVCにおける参照ピクチャリスト初期化および修正プロセス、または、2)参照ピクチャセット、ならびにHEVCにおける参照ピクチャリスト初期化および修正プロセスのいずれかのための、シグナリングおよび復号プロセスの必要をなくす。

20

【0115】

残念ながら、参照ピクチャリスト(RPL)ベースの参照ピクチャの管理には欠点がある。たとえば、一部のRPLベースの参照ピクチャ管理方式は、ビットストリームにおけるシンタックス要素のシグナリングの面で最適化されていない。これにより、そのようなRPLベースの手法は、RPSベースの手法などの他の明白な参照ピクチャ管理の手法と比較すると、シグナリングのためのビットカウントがより多くなる。RPLベースの手法のシグナリングの非効率性は、いくつかの側面によるものである。

30

【0116】

たとえば、RPL構造における一部のシンタックス要素は、より非効率なエントロピーコーディングを用いてコーディングされる。たとえば、短期参照ピクチャ(STRP)のデルタPOC値を表すシンタックス要素のコーディングは、デルタPOC値が正の値または負の値のいずれかを有し得るので、左のビットが最初の、符号付き整数0次指数ゴロム符号化されたシンタックス要素(たとえば、se(v))を使用する。指数ゴロム符号(または単にExp-Golomb符号)は、一種の汎用的な符号である。指数ゴロム符号を使用して任意の非負の整数xを符号化するために、最初のステップはx+1を二進数で書くことである。次に、書かれたビットを数え、1を引き、そして、その数の開始0ビットを以前のビット文字列の前に書く。符号の最初のいくつかの値は、0 1 1、1 10 010、2 11 011、3 100 0010 0である。しかしながら、このプロセスは、デルタPOCの符号値と絶対値が別々にコーディングされれば、より効率的にすることができる。

40

【0117】

(たとえば、参照ピクチャリスト0および参照ピクチャリスト1のための)RPLインデックスを、それらがパラメータセット(たとえば、SPS)の中のあらかじめ定められたRPL構造を参照するときにシグナリングすることは、両方のインデックスを常にシグナリングするような方式ほど効率が低いことがある。多くの場合、RPL0およびRPL1のあらかじめ定められたRPL構造は、ピクチャのRPLがパラメータセットの中のあらかじめ定められたRPL構造を参照するときにRPL0のインデックスとRPL1のインデックスが同じになるように

50

、なされ得る。

【0118】

さらに、コーディングされたビデオビットストリームへと参照ピクチャリスト構造を書き込むための、インター-RPLコーディングはない。たとえば、RPL構造は、ビットストリームにおける以前にシグナリングされたRPL構造を参照せずにコーディングされ得る。

【0119】

本明細書において説明される問題を解決し、欠点に対処する、個々におよび/または組合せで適用され得るいくつかの態様が、本明細書において説明される。それらのいくつかの態様の各々は以下で詳述される。

【0120】

一態様では、デルタPOC値がSTRPのためにコーディングされる時、符号値および絶対値はビデオビットストリームにおいて別々にコーディングされる。各RPL構造の中のデルタPOC値が同じ符号値を有する(たとえば、同じRPL構造の中のすべてのデルタPOCが正のデルタPOC値であるか、または負のデルタPOC値であるかのいずれか)かどうかを示すために、あらかじめ定められたRPL構造と同じパラメータセット(たとえば、SPS、PPS)においてフラグがシグナリングされる。このフラグは、「all_rpl_entries_same_sign_flag」と呼ばれ得る。

【0121】

all_rpl_entries_same_sign_flagが1に等しいとき、RPL構造の中のすべてのエントリの符号値(たとえば、STRPのデルタPOC値の符号値)を示すために、RPL構造においてフラグがシグナリングされる。all_rpl_entries_same_sign_flagが0に等しいとき、各RPL構造の中のSTRPと関連付けられるエントリの各々の符号値を示すために、RPL構造においてフラグがシグナリングされる。ある実施形態では、RPL構造の中のSTRPエントリのデルタPOC値の絶対値は、符号なし整数0次指数ゴロム符号(たとえば、ue(v))を使用してコーディングされる。

【0122】

代替として、all_rpl_entries_same_sign_flagという1つのフラグのための代わりに、各RPL構造リストに対して1つずつ(たとえば、RPL0のためのRPL構造リストおよびRPL1のためのRPL構造リスト)、2つのフラグがシグナリングされ得る。

【0123】

符号化のために、パラメータセットの中のRPL0リストおよびRPL1リストのためのあらかじめ定められたRPL構造を作成するとき、RPL0およびRPL1のためのリストの中のRPL構造は、リスト0のためのRPL構造およびリスト1のためのRPL構造が対になるようになされ得る。したがって、ピクチャがパラメータセットの中のあらかじめ定められたRPL構造を参照するとき、そのピクチャのRPL0のためのインデックスおよびRPL1のためのインデックスは同じである。

【0124】

上記の点に関連して、フラグは、あらかじめ定められたRPL構造を参照するRPL1のためのシンタックス要素がスライスヘッダに存在するかしないかを示し得る。このフラグは「rpl1_idx_present_flag」と呼ばれ得る。このフラグは、フラグの意図される範囲または持続性に応じて、SPSまたはPPSにおいてシグナリングされ得る。ある実施形態では、PPSにおけるフラグのシグナリングが好ましい。

【0125】

rpl1_idx_present_flagの値に基づいて、以下のことが当てはまる。rpl1_idx_present_flagが1に等しいとき、フラグを含むパラメータセットと関連付けられるスライスヘッダのRPL1はあらかじめ定められたRPL構造を参照し、そのインデックスはスライスヘッダにおいてシグナリングされず、代わりに、同じスライスの中のRPL0のための対応するシンタックス要素から推測される。すなわち、ref_pic_list_sps_flag[1]およびref_pic_list_idx[1]はシグナリングされず、それぞれ、ref_pic_list_sps_flag[0]およびref_pic_list_idx[0]の値から推測(たとえば、複製)される。それ以外の場合、rpl1_idx_present_

10

20

30

40

50

flagが0に等しいとき、ref_pic_list_sps_flag[1]およびref_pic_list_idx[1]は、フラグを含むパラメータセットと関連付けられるスライスヘッダに存在する。

【0126】

代替として、rpl1_idx_present_flagの値に基づいて、以下のことが当てはまる。rpl1_idx_present_flagが1に等しいとき、RPL1のインデックスは、フラグを含むパラメータセットと関連付けられるスライスヘッダのRPL1があらかじめ定められたRPL構造を参照するときは、スライスヘッダにおいてシグナリングされない。むしろ、RPL1のインデックスは、同じスライスの中のRPL0のための対応するシンタックス要素から推測される。すなわち、ref_pic_list_idx[1]はシグナリングされず、それぞれ、ref_pic_list_idx[0]の値から推測(たとえば、複製)される。それ以外の場合、rpl1_idx_present_flagが0に等しいとき、ref_pic_list_idx[1]は、フラグを含むパラメータセットと関連付けられるスライスヘッダに存在する。

10

【0127】

RPL1のためのリストの中のRPL構造が、RPL0のためのリストの中のRPL構造と同一のコンテンツを有するかどうかを示すために、フラグが利用され得る。このフラグは、「rpl1_copy_from_rpl0_flag」と呼ばれ得る。このフラグは、あらかじめ定められたRPL構造を含む同じパラメータセットにおいてシグナリングされ、あらかじめ定められたRPL構造のシグナリングの前に位置特定されるものとする。

【0128】

rpl1_copy_from_rpl0_flagの値に基づいて、以下のことが当てはまる。rpl1_copy_from_rpl0_flagが1に等しいとき、リスト1のためのRPL構造の数はシグナリングされず、代わりに、リスト0のためのRPL構造の数と同じであると推測される。リスト1のためのRPL構造はシグナリングされない。代わりに、リスト0のためのあらかじめ定められたRPL構造が構文解析されてデコーダにおいて復号された後で、リスト0のための各RPL構造に対して、厳密に同じRPL構造の複製が作成され、RPL構造の複製は同じインデックスを伴うリスト1のためのRPL構造として割り当てられる。それ以外の場合、rpl1_copy_from_rpl0_flagが0に等しいとき、リスト1のためのRPL構造の数およびリスト1のためのRPL構造がシグナリングされる。

20

【0129】

RPL構造の内容は、別のRPL構造から予測され得る。パラメータセットの中のあらかじめ定められたRPL構造が2つのリスト(たとえば、リスト0のためのRPL構造リストおよびリスト1のためのRPL構造リスト)へと分割されるとき、リスト0のためのリストの中のRPL構造だけが、インター-RPLのための参照として使用され得る。リスト0のためのリストの中のRPL構造は、そのRPL構造のインデックス未満のインデックスを伴う同じリストの中の別のRPL構造だけを指し、一方で、リスト1のためのリストの中のRPL構造は、リスト0のためのリストの中の任意のRPL構造を指すことができる。スライスヘッダにおいて明示的にシグナリングされるRPL構造は、リスト0のためのリストの中のあらかじめ定められたRPL構造のいずれをも指すことができる。代替として、リスト0またはリスト1のためのリストの中のRPL構造は、そのRPL構造のインデックスより小さいインデックスを伴うリスト0のためのリストの中の別のRPL構造だけを指すことができる。スライスヘッダにおいて明示的にシグナリングされるRPL構造は、リスト0のためのリストの中のあらかじめ定められたRPL構造のいずれをも指すことができる。

30

40

【0130】

1つの代替形態では、パラメータセットの中のあらかじめ定められたRPL構造が2つのリスト(たとえば、リスト0のためのRPL構造リストおよびリスト1のためのRPL構造リスト)へと分割されるとき、リスト0のためのリストの中のRPL構造は、そのRPL構造のインデックスより小さいインデックスを伴うリスト0のためのリストの中の別のRPL構造のみを指すことができる。同様に、リスト1のためのリストの中のRPL構造は、そのRPL構造のインデックスより小さいインデックスを伴うリスト1のためのリストの中の別のRPL構造のみを指すことができる。RPL0のためのスライスヘッダにおいて明示的にシグナリングされ

50

るRPL構造は、リスト0のためのリストの中のあらかじめ定められたRPL構造のいずれをも指すことができ、RPL1のためのスライスヘッダにおいて明示的にシグナリングされるRPL構造は、リスト1のためのリストの中のあらかじめ定められたRPL構造のいずれをも指すことができる。

【0131】

別の代替形態では、パラメータセットの中のあらかじめ定められたRPL構造が2つのリストへと分割されない(たとえば、リスト0のためのRPL構造リストおよびリスト1のためのRPL構造リストが1つのリストでシグナリングされる)とき、RPL構造は、そのインデックスより小さいインデックスを伴うリストの中の別のRPL構造のみを指すことができる。スライスヘッダにおいて明示的にシグナリングされるRPL構造は、あらかじめ定められたRPL構造のいずれをも指すことができる。

10

【0132】

インターRPLでは、参照RPL構造のためのインデックスは、現在のRPL構造インデックスと参照RPL構造インデックスのデルタから1を引いたものとしてコーディングされ、 $ue(v)$ コーディングを用いてコーディングされ得る。1つの代替形態では、参照RPLインデックスは、 $u(v)$ コーディングを使用して直接コーディングされる。インデックスを表現するために使用されるビットの数は、参照に適したリストの中のRPL構造の数の二進対数として設定される。たとえば、リスト0のためのリストの中のRPL構造のみが参照として使用され得るとき、参照RPLインデックスを表現するためのビットの数は、リスト0のためのリストの中のRPL構造の数の二進対数である。別の代替形態では、参照RPL構造のためのインデックスは、インターRPLのモードに応じて、 $ue(v)$ または $u(v)$ のいずれかを使用してコーディングされ得る。

20

【0133】

インターRPLをサポートするために、各RPL構造は以下で説明されるモードのうちの1つを使用してコーディングされる。以下のモードの順序は、モードの値の順序を必ずしも意味しないことに留意されたい。モードの指示は、次のようにコーディングされ得る。ある実施形態では、モードの指示は、単に $ue(v)$ を使用してコーディングされ得る。ある実施形態では、モードの指示は、定義されたモードの総数の二進対数である、モードを表すためのビットの数をを用いて、 $u(v)$ を使用してコーディングされ得る。

【0134】

第1のRPLコーディングモードは、イントラコーディングモードである。このモードは、RPL構造の内容をシグナリングするための他の機構と同じように機能する。たとえば、2018年8月17日に出版された「Reference Picture Management in Video Coding」という表題の米国仮出願第62/719,360号において説明される方法を参照されたい。代替として、0より大きい整数値が、各RPL構造に対してシグナリングされ、 $granularity_val$ と呼ばれ得る。 $granularity_val$ の値は、RPL構造においてSTRPのPOCデルタ値を表す各値をスケールリングまたは分割するために使用される。

30

【0135】

第2のRPLコーディングモードは、同じまたは均等な差を有する、参照RPL構造および現在のRPL構造の中のSTRPに対するデルタPOC値を利用する、インターコーディングモードである。このモードを使用してRPLをコーディングするために、以下の情報がビットストリームにおいてシグナリングされる。

40

【0136】

始めに、コーディングモードおよび参照インデックスがビットストリームにおいてシグナリングされる。参照インデックスは、参照RPL構造のインデックスである。参照インデックスは、上で説明されたようにコーディングされ得る。しかしながら、機構はまた、 $ue(v)$ としてコーディングされる、現在のRPL構造インデックスと参照RPL構造インデックスとのデルタから1を引いたものとして、参照インデックスをコーディングしてもよい。オフセットもビットストリームにおいてシグナリングされる。オフセットは、参照RPL構造の中のSTRPのデルタPOCと現在のRPL構造の中のSTRPのデルタPOCとの間の差である。

50

オフセットの値は、正の値だけに制約されてもよく(たとえば、参照RPL構造の中のSTRPのデルタPOC値が現在のRPL構造の中のSTRPのデルタPOC値未満である場合、現在のRPL構造をコーディングするためにこのモードを使用することはできない)、負の値だけに制約されてもよく、または正もしくは負のいずれかであってもよい。オフセットは、 $ue(v)$ としてコーディングされる場合、`term_minus1`を用いてシグナリングされ得る。エントリが現在のRPL構造の中のエントリとして使用されるかどうかを示すための、参照RPL構造の中の各エントリに対するフラグも、ビットストリームにおいてシグナリングされる。エントリが参照RPL構造の中のSTRPエントリであり、現在のRPL構造の中のエントリとして使用されるとき、エントリは、現在のRPL構造の中のSTRPエントリでもあり、その値は、参照RPL構造の中のエントリ値からオフセットを引いたものである(オフセットをコーディングするために`_minus1`項が使用される場合には1を足す)。エントリが参照RPL構造の中の長期参照ピクチャ(LTRP)エントリであり、現在のRPL構造の中のエントリとして使用されるとき、エントリは、現在のRPL構造の中のLTRPエントリでもあり、その値は、参照RPL構造の中のエントリのために単に複製される。

10

【0137】

第3のRPLコーディングモードは、現在のRPL構造の中のエントリ(たとえば、STRPおよびLTRP)が参照RPL構造の中のエントリの上位集合であるという事実を利用する、インターコーディングモードである。すなわち、参照RPL構造の中のすべてのエントリが、追加の0個以上のエントリを伴う最初のX個(参照RPL構造の中のエントリの数でXを置き換える)と同じである。このモードを使用してRPLをコーディングするために、以下の情報がビットストリームにおいてシグナリングされる。

20

【0138】

始めに、コーディングモードおよび参照インデックスが、ビットストリームにおいてシグナリングされる。参照インデックスは、参照RPL構造のインデックスである。参照インデックスは上で説明されたようにコーディングされ得る。追加のエントリの数もビットストリームにおいてシグナリングされる。追加のエントリ数は、現在のRPL構造の中のエントリ数と参照RPL構造の中のエントリ数との差である。エントリが参照RPL構造の中のSTRPエントリであるとき、エントリは、現在のRPL構造の中のSTRPエントリでもあり、その値は、参照RPL構造の中のエントリに対して単に複製される。エントリが参照RPL構造の中のLTRPエントリであるとき、エントリは、現在のRPL構造の中のLTRPエントリでもあり、その値は、参照RPL構造の中のエントリに対して単に複製される。参照RPL構造からすべてのエントリが現在のRPL構造へと複製された後で、各々の追加のエントリに対して、以下の情報がシグナリングされる。長期参照ピクチャがビットストリームにおいて使用される場合(すなわち、これは同じパラメータセットの中のフラグによって示され得る)、追加のエントリがLTRPエントリであるか、またはSTRPエントリであるかを示すために、フラグがシグナリングされる。エントリがLTRPエントリである場合、LTRPエントリのPOC LSBがシグナリングされる。それ以外の場合、STRPエントリのデルタPOCがシグナリングされる。デルタPOCの値は、以前のSTRPエントリからのデルタとして、または単に現在のピクチャのPOCからのデルタとしてシグナリングされ得る。

30

【0139】

第4のRPLコーディングモードは、現在のRPL構造の中のエントリ(たとえば、STRPおよびLTRP)が、参照RPL構造の中のエントリと厳密に同じであるか、または、符号値が反転した状態で厳密に同じであるかのいずれかであるという事実を利用する、インターコーディングモードである。このモードを使用してRPLをコーディングするために、以下の情報がビットストリームにおいてシグナリングされる。

40

【0140】

始めに、コーディングモードおよび参照インデックスがシグナリングされる。参照インデックスは参照RPL構造のインデックスである。参照インデックスは上で説明されたようにコーディングされ得る。任意選択で、符号値が反転しているかどうかを示すためのフラグも、ビットストリームにおいてシグナリングされる。

50

【 0 1 4 1 】

ビットストリームが前方インター予測(たとえば、現在のピクチャのPOC値より小さいPOC値を伴うピクチャに関するインター予測)および後方インター予測(たとえば、現在のピクチャのPOC値より大きいPOC値を伴うピクチャに関するインター予測)を用いてコーディングされるとき、RPLの中の参照ピクチャに対する以下の制約が適用される。ピクチャの各RPLに対して、RPLの中のすべての参照ピクチャは、インター予測方向が同じであり、たとえば、すべての参照ピクチャが前方インター予測のための参照ピクチャであり、または、すべての参照ピクチャが後方インター予測のための参照ピクチャである。ピクチャのRPLのペアに対して、以下の条件が真ではない限り、RPL0とRPL1の両方に含まれる参照ピクチャはないものとする。その条件とは、RPL0とRPL1の両方が同じ順序で同じ参照ピクチャを含むということである。そのような条件では、RPL1の中の最初のピクチャ(すなわち、インデックスが最低のピクチャ)を除くすべての参照ピクチャがRPL1から除去される。代替として、RPL0の中の最初のピクチャ(すなわち、インデックスが最低のピクチャ)を除くすべての参照ピクチャがRPL0から除去される。

10

【 0 1 4 2 】

デルタピクチャ順序カウント(POC)の絶対値の符号値を指定するためのフラグを使用するビデオコーディング技法が、本明細書において開示される。符号値は、デルタPOC値の絶対値とは別々にシグナリングされる。符号値および絶対値を別々にシグナリングすることによって、ビデオコーディングにおけるコーデック/デコーデック(「コーデック」としても知られている)は、現在のコーデックと比較して改善される(たとえば、使用するビットが少なくなる、必要な帯域幅が少なくなる、より効率的であるなど)。現実問題として、改善されたビデオコーディングプロセスは、ビデオが送信されるとき、受信されるとき、および/または見られるときに、より良いユーザ体験をユーザに提供する。

20

【 0 1 4 3 】

図5は、ビデオビットストリーム500の実施形態の概略図である。本明細書では、ビデオビットストリーム500は、コーディングされたビデオビットストリーム、ビットストリーム、またはそれらの変形として呼ばれることもある。図5に示されるように、ビットストリーム500は、シーケンスパラメータセット(SPS)510、ピクチャパラメータセット(PPS)512、スライスヘッダ514、および画像データ520を備える。

【 0 1 4 4 】

SPS510は、ピクチャのシーケンス(SOP)の中のすべてのピクチャに共通のデータを含む。対照的に、PPS512は、ピクチャ全体に共通のデータを含む。スライスヘッダ514は、たとえば、スライスタイプ、参照ピクチャのうちのいずれが使用されるかなどの、現在のスライスについての情報を含む。SPS510およびPPS512は、総称的にパラメータセットと呼ばれ得る。SPS510、PPS512、およびスライスヘッダ514は、ネットワーク抽象化レイヤ(NAL)ユニットのタイプである。画像データは、符号化または復号されている画像もしくはビデオと関連付けられるデータを備える。画像データ520は単に、ビットストリーム500において搬送されているペイロードまたはデータと呼ばれ得る。

30

【 0 1 4 5 】

ある実施形態では、SPS510、PPS512、スライスヘッダ514、またはビットストリーム500の別の部分は、その各々が複数の参照ピクチャエントリを含む、複数の参照ピクチャリスト構造を搬送する。ビットストリーム500は、実際の適用例では他のパラメータおよび情報を含み得ることを、当業者は理解するであろう。

40

【 0 1 4 6 】

図6は、ビデオデコーデック(たとえば、ビデオデコーデック30)によって実装されるコーディングされたビデオビットストリーム(たとえば、ビットストリーム500)を復号する方法600の実施形態である。方法600は、復号されたビットストリームがビデオエンコーダ(たとえば、ビデオエンコーダ20)から直接または間接的に受信された後で実行され得る。方法600は、デルタPOCの絶対値の符号値とデルタPOCの絶対値が互いに別々にシグナリングされるので、復号プロセスを改善する(たとえば、復号プロセスを従来の復号プロセスよりも

50

効率的に、高速になどする)。これは、HEVCおよびAVCにおいてデルタPOCが扱われる方式とは対照的である。したがって、現実問題として、コーデックの性能が改善され、これはより良いユーザ体験につながる。

【0147】

ブロック602において、ビデオビットストリーム(たとえば、ビデオビットストリーム500)からフラグが構文解析される。ある実施形態では、フラグは、複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値の符号値を指定する。符号値は、正(+)または負(-)のいずれかであり得る。ある実施形態では、フラグは、1の値または0の値を有する。ある実施形態では、正の符号値および負の符号値は、現在の参照ピクチャに関して先行する参照ピクチャまたは後続する参照ピクチャを示す。

10

【0148】

ある実施形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値の符号値が同じであることを示す。ある実施形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値の符号値が正であることを示す。ある実施形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値の符号値が負であることを示す。ある実施形態では、フラグはall_rpl_entries_same_sign_flagと指定される。

【0149】

ある実施形態では、フラグは、ビデオビットストリーム(たとえば、ビットストリーム500)のSPS(たとえば、SPS510)においてシグナリングされる。ある実施形態では、フラグおよび参照ピクチャリスト構造は、同じパラメータセットにおいてシグナリングされる。

20

【0150】

ある実施形態では、フラグが1に等しいとき、参照ピクチャリスト構造の中のすべてのエントリの符号値は、参照ピクチャリスト構造においてシグナリングされる。ある実施形態では、フラグが0に等しいとき、符号値は、参照ピクチャリスト構造の中の短期参照ピクチャエントリと関連付けられる各エントリのためにシグナリングされる。

【0151】

ブロック604において、短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値が構文解析される。ある実施形態では、デルタPOCの絶対値は、符号なし整数0次指数ゴロム符号化に基づいて復号される。ある実施形態では、フラグおよび絶対値は互いに別々に構文解析される。すなわち、フラグがまず構文解析されて、次いで絶対値が構文解析され、またはこの逆である。

30

【0152】

ブロック606において、デルタPOCの絶対値の符号値およびデルタPOCの絶対値に基づいて、参照ピクチャリストが生成される。参照ピクチャリストは、たとえば、図4に関連して図示され説明されたピクチャなどの、1つまたは複数のピクチャを特定し得る。

【0153】

ブロック608において、再構築されたブロックを生成するために、参照ピクチャリストに基づいてインター予測が実行される。ある実施形態では、再構築されたブロックは、電子デバイス(たとえば、スマートフォン、タブレット、ラップトップ、パーソナルコンピュータなど)のディスプレイまたはスクリーン上でユーザに表示するための画像を生成または作成するために使用され得る。

40

【0154】

図7は、ビデオエンコーダ(たとえば、ビデオエンコーダ20)によって実施されるビデオビットストリーム(たとえば、ビットストリーム500)を符号化する方法700の実施形態である。方法700は、(たとえば、ビデオからの)ピクチャがビデオビットストリームへと符号化され、次いでビデオデコーダ(たとえば、ビデオデコーダ30)に向かって送信されることになるときに実行され得る。方法700は、デルタPOCの絶対値の符号値とデルタPOCの絶対値が互いに別々にシグナリングされるので、符号化プロセスを改善する(たとえば、符号化プロセスを従来の符号化プロセスよりも効率的に、高速になどする)。これは、HEVC

50

およびAVCにおいてデルタPOCが扱われる方式とは対照的である。したがって、現実問題として、コーデックの性能が改善され、これはより良いユーザ体験につながる。

【0155】

ブロック702において、複数の参照ピクチャエントリを各々含む複数の参照ピクチャリスト構造が、ビデオビットストリームへと符号化される。1つまたは複数の参照ピクチャリスト構造は、たとえば、SPS(たとえば、SPS510)においてシグナリングされてもよく、それらの各々は、スライスヘッダ(たとえば、スライスヘッダ514)からインデックスによって参照されてもよい。ある実施形態では、1つまたは複数の参照ピクチャリスト構造は、スライスヘッダ(たとえば、スライスヘッダ514)においてシグナリングされてもよい。

【0156】

ブロック704において、複数の参照ピクチャエントリからの短期参照ピクチャエントリと関連付けられるデルタピクチャ順序カウンタ(POC)の絶対値の符号値を指定するフラグが、ビデオビットストリームへと符号化される。ある実施形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値の符号値が同じであることを示す。ある実施形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値の符号値が正であることを示す。ある実施形態では、フラグは、複数の参照ピクチャリスト構造のうちの1つの中のすべてのデルタPOC値が負であることを示す。ある実施形態では、フラグはall_rpl_entries_same_sign_flagと指定される。ある実施形態では、フラグは、ビデオビットストリームのSPSにおいて符号化される。ある実施形態では、フラグおよび参照ピクチャリスト構造は、同じパラメータセットにおいて符号化される。

【0157】

ある実施形態では、フラグが1に等しいとき、参照ピクチャリスト構造の中のすべてのエントリの符号値は、参照ピクチャリスト構造において符号化される。ある実施形態では、フラグが0に等しいとき、符号値は、参照ピクチャリスト構造の中の短期参照ピクチャエントリと関連付けられる各エントリのために符号化される。

【0158】

ブロック706において、短期参照ピクチャエントリと関連付けられるデルタPOCの絶対値が、ビデオビットストリームへと符号化される。ある実施形態では、デルタPOCの絶対値は、符号なし整数0次指数ゴロム符号化に基づいて符号化される。ある実施形態では、フラグおよび絶対値は、互いに別々に符号化される。すなわち、フラグがまず構文解析され、次いで絶対値が構文解析され、またはこの逆である。

【0159】

ブロック708において、ビデオビットストリーム(たとえば、コーディングまたは符号化されたビデオビットストリームとしても知られている)がビデオデコーダに向かって送信される。ビデオデコーダによって受信されると、符号化されるビデオビットストリームは、電子デバイス(たとえば、スマートフォン、タブレット、ラップトップ、パーソナルコンピュータなど)のディスプレイまたはスクリーン上でユーザに表示するための画像を生成または作成するために(たとえば、上で説明されたように)復号され得る。

【0160】

図8は、ビデオデコーダ(たとえば、ビデオデコーダ30)によって実施されるコーディングされたビデオビットストリーム(たとえば、ビットストリーム500)を復号する方法800の実施形態である。方法800は、復号されたビットストリームがビデオエンコーダ(たとえば、ビデオエンコーダ20)から直接または間接的に受信された後で実行され得る。方法800は、デルタPOCの絶対値の符号値とデルタPOCの絶対値が互いに別々にシグナリングされるので、復号プロセスを改善する(たとえば、復号プロセスを従来の復号プロセスよりも効率的に、高速になどする)。これは、HEVCおよびAVCにおいてデルタPOCが扱われる方式とは対照的である。したがって、現実問題として、コーデックの性能が改善され、これはより良いユーザ体験につながる。

【0161】

10

20

30

40

50

ブロック802において、コーディングされたビデオビットストリームにおいて表される参照ピクチャリスト構造が構文解析される。ある実施形態では、参照ピクチャリスト構造は、シンタックス要素のセットを備える。シンタックス要素のセットは、参照ピクチャを記述し、ピクチャ順序カウンタのデルタ値および符号フラグを含む。ある実施形態では、参照ピクチャのPOCを取得するために、POCのデルタ値が使用される。ある実施形態では、符号フラグは、POCのデルタ値の符号値を指定する。

【0162】

ある実施形態では、ピクチャ順序カウンタのデルタ値および符号フラグは別々に復号される。ある実施形態では、ピクチャ順序カウンタのデルタ値は絶対値である。

【0163】

ブロック804において、現在のスライスの参照ピクチャリストが、参照ピクチャリスト構造に基づいて導出される。ある実施形態では、参照ピクチャリストは参照ピクチャを備える。

【0164】

ブロック806において、現在のスライスの少なくとも1つの再構築されたブロックが、参照ピクチャリストに基づいて取得される。ある実施形態では、再構築されたブロックは、電子デバイス(たとえば、スマートフォン、タブレット、ラップトップ、パーソナルコンピュータなど)のディスプレイまたはスクリーン上でユーザに表示するための画像を生成または作成するために使用され得る。

【0165】

本明細書において開示される技法の説明は、最新のVVC WDに関連して与えられる。加えて、本明細書において開示される技法を実装するのに適した定義、シンタックス、およびセマンティクスも与えられる。

【0166】

始めに、いくつかの定義が与えられる。イントラランダムアクセスポイント(IRAP)ピクチャは、各ビデオコーディングレイヤ(VCL)NALユニットがIRAP_NUTに等しいnal_unit_typeを有するコーディングされたピクチャである。長期参照ピクチャ(LTRP)は、「長期参照のために使用される」とマークされるピクチャである。非IRAPピクチャは、各VCL NALユニットがNON_IRAP_NUTに等しいnal_unit_typeを有するコーディングされたピクチャである。参照ピクチャリストは、PまたはBスライスのインター予測のために使用される参照ピクチャのリストである。2つの参照ピクチャリスト、参照ピクチャリスト0および参照ピクチャリスト1は、非IRAPピクチャの各スライスに対して生成される。ピクチャと関連付けられる2つの参照ピクチャリストの中のすべてのエントリによって参照される固有のピクチャのセットは、関連するピクチャ、または復号順序において関連するピクチャの後にある任意のピクチャのインター予測のために使用され得る、すべての参照ピクチャを含む。Pスライスのスライスデータを復号するために、参照ピクチャリスト0だけがインター予測のために使用される。Bスライスのスライスデータを復号するために、両方の参照ピクチャリストがインター予測のために使用される。Iスライスのスライスデータを復号するために、インター予測のために参照ピクチャリストは使用されない。短期参照ピクチャ(STRP)は、「短期参照のために使用される」とマークされるピクチャである。

【0167】

次に、いくつかの略語が与えられる。本明細書では、LTRPは長期参照ピクチャを意味し、STRPは短期参照ピクチャを意味する。

【0168】

本開示の以下の部分は、本明細書において開示される技法を実施するのに適したシンタックスおよびセマンティクスを提供する。

【0169】

NALユニットヘッダシンタックス

【0170】

10

20

30

40

50

【表 1】

nal_unit_header() {	記述子
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_reserved_zero_7bits	u(7)
}	

10

【0171】

シーケンスパラメータセットRBSPシンタックス

【0172】

【表 2】

seq_parameter_set_rbsp() {	記述子
sps_seq_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
log2_max_pic_order_cnt_lsb_minus4	ue(v)
sps_max_dec_pic_buffering_minus1	ue(v)
qtbtt_dual_tree_intra_flag	ue(v)
log2_ctu_size_minus2	ue(v)
log2_min_qt_size_intra_slices_minus2	ue(v)
log2_min_qt_size_inter_slices_minus2	ue(v)
max_mtt_hierarchy_depth_inter_slices	ue(v)
max_mtt_hierarchy_depth_intra_slices	ue(v)
long_term_ref_pics_flag	u(1)
if(long_term_ref_pics_flag)	
additional_lt_poc_lsb	ue(v)
all_rpl_entries_same_sign_flag	u(1)
rpl1_copy_from_rpl0_flag	u(1)
for(i = 0; i < (!rpl1_copy_from_rpl0_flag) ? 2 : 1; i++) {	
num_ref_pic_lists_in_sps[i]	ue(v)
for(j = 0; j < num_ref_pic_lists_in_sps[i]; j++)	
ref_pic_list_struct(i, j, long_term_ref_pics_flag)	
}	
rbsp_trailing_bits()	
}	

20

30

40

【0173】

ピクチャパラメータセットRBSPシンタックス

【0174】

50

【表 3】

pic_parameter_set_rbsp() {	記述子
pps_pic_parameter_set_id	ue(v)
pps_seq_parameter_set_id	ue(v)
for(i = 0; i < 2; i++)	
num_ref_idx_default_active_minus1[i]	ue(v)
rpl1_idx_present_flag	u(1)
rbsp_trailing_bits()	
}	

10

【 0 1 7 5 】

スライスヘッダシンタックス

【 0 1 7 6 】

【表 4】

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
slice_address	u(v)
slice_type	ue(v)
if(slice_type != I)	
log2_diff_ctu_max_bt_size	ue(v)
if(nal_unit_type != IRAP_NUT) {	
slice_pic_order_cnt_lsb	u(v)
for(i = 0; i < 2; i++) {	
if(i == 0 (i == 1 && rpl1_idx_present_flag))	
ref_pic_list_sps_flag[i]	u(1)
if(ref_pic_list_sps_flag[i]) {	
if(num_ref_pic_lists_in_sps[i] > 1)	
if(i == 0 (i == 1 &&	
rpl1_idx_present_flag)	
ref_pic_list_idx[i]	u(v)
} else	
ref_pic_list_struct(i, num_ref_pic_lists_in_sps[i],	
long_term_ref_pics_flag)	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag)	
for(i = 0; i < (slice_type == B ? 2 : 1); i++)	
num_ref_idx_active_minus1[i]	ue(v)
}	
}	
byte_alignment()	
}	

20

30

40

【 0 1 7 7 】

参照ピクチャリスト構造シンタックス

50

【 0 1 7 8 】

【 表 5 】

ref_pic_list_struct(listIdx, rplsIdx, ltrpFlag) {	記述子	
rpl_mode [listIdx][rplsIdx]	u(2)	
if(rpl_mode[listIdx][rplsIdx] == 0) {		
num_strp_entries [listIdx][rplsIdx]	ue(v)	
if(ltrpFlag)		
num_ltrp_entries [listIdx][rplsIdx]	ue(v)	
if(all_rpl_entries_same_sign_flag)		10
strp_entries_sign_flag [listIdx][rplsIdx]	u(1)	
for(i = 0; i < NumEntriesInList[listIdx][rplsIdx]; i++) {		
if(num_ltrp_entries[listIdx][rplsIdx] > 0)		
lt_ref_pic_flag [listIdx][rplsIdx][i]	u(1)	
if(!lt_ref_pic_flag[listIdx][rplsIdx][i]) {		
if(!all_rpl_entries_same_sign_flag)		
strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)	
delta_poc_st [listIdx][rplsIdx][i]	ue(v)	
} else		
poc_lsb_lt [listIdx][rplsIdx][i]	u(v)	20
}		
} else if(rpl_mode[listIdx][rplsIdx] == 1) {		
mode1_ref_rpl_idx_delta_minus1 [listIdx][rplsIdx]	ue(v)	
strp_offset_val_minus1 [listIdx][rplsIdx]	ue(v)	
for(i = 0; i < NumEntriesInList[0][RefRplIdx]; i++)		
ref_entry_used_flag [listIdx][rplsIdx][i]	u(1)	
} else if(rpl_mode[listIdx][rplsIdx] == 2) {		
mode2_ref_rpl_idx [listIdx][rplsIdx]	u(v)	
num_additional_entries [listIdx][rplsIdx]	ue(v)	
for(i = 0; i < num_additional_entries[listIdx][rplsIdx]; i++) {		
if(ltrpFlag)		
add_lt_ref_pic_flag [listIdx][rplsIdx][i]	u(1)	30
if(!add_lt_ref_pic_flag[listIdx][rplsIdx][i]) {		
if(!all_rpl_entries_same_sign_flag)		
add_strp_entry_sign_flag [listIdx][rplsIdx][i]	u(1)	
add_delta_poc_st [listIdx][rplsIdx][i]	ue(v)	
} else		
add_poc_lsb_lt [listIdx][rplsIdx][i]	u(v)	
}		
} else if(rpl_mode[listIdx][rplsIdx] == 3)		
mode3_ref_rpl_idx [listIdx][rplsIdx]	u(v)	
}		40
}		

【 0 1 7 9 】

NALユニットヘッダセマンティクス

【 0 1 8 0 】

forbidden_zero_bitは0に等しいものとする。nal_unit_typeはNALユニットに含まれるRBSPデータ構造のタイプを指定する。

【 0 1 8 1 】

【表 6】

Table 4-1 - NALユニットタイプコードおよびNALユニットタイプクラス

nal_unit_type	nal_unit_typeの名前	NALユニットの内容およびRBSPシンタックス構造	NALユニットタイプクラス
0	NON_IRAP_NUT	非IRAPピクチャのコーディングされたスライスセグメント slice_layer_rbsp()	VCL
1	IRAP_NUT	IRAPピクチャのコーディングされたスライス slice_layer_rbsp()	VCL
2-15	RSV_VCL_NUT	予備のVCL NALユニット	VCL
16	SPS_NUT	シーケンスパラメータセット seq_parameter_set_rbsp()	非VCL
17	PPS_NUT	ピクチャパラメータセット pic_parameter_set_rbsp()	非VCL
18	EOS_NUT	シーケンスの終了 end_of_seq_rbsp()	非VCL
19	EOB_NUT	ビットストリームの終了 end_of_bitstream_rbsp()	非VCL
20, 21	PREFIX_SEI_NUT SUFFIX_SEI_NUT	補足強化情報 sei_rbsp()	非VCL
22-26	RSV_NVCL	予備	非VCL
27-31	UNSPEC	未指定	非VCL

【0182】

nuh_temporal_id_plus1から1を引いたものは、NALユニットに対する時間識別子を指定する。nuh_temporal_id_plus1の値は0に等しくないものとする。変数TemporalIdは、TemporalId=nuh_temporal_id_plus1-1のように指定される。nal_unit_typeがIRAP_NUTに等しいとき、コーディングされるスライスはIRAPピクチャに属し、TemporalIdは0に等しいものとする。TemporalIdの値は、アクセスユニットのすべてのVCL NALユニットに対して同じであるものとする。コーディングされたピクチャまたはアクセスユニットのTemporalIdの値は、コーディングされたピクチャまたはアクセスユニットのVCL NALユニットのTemporalIdの値である。非VCL NALユニットに対するTemporalIdの値は次のように制約される。nal_unit_typeがSPS_NUTに等しい場合、TemporalIdは0に等しいものとし、NALユニットを含むアクセスユニットのTemporalIdは0に等しいものとする。そうではなく、nal_unit_typeがEOS_NUTまたはEOB_NUTに等しい場合、TemporalIdは0に等しいものとする。それ以外の場合、TemporalIdはNALユニットを含むアクセスユニットのTemporalId以上であるものとする。NALユニットが非VCL NALユニットであるとき、TemporalIdの値は、非VCL NALユニットが適用されるすべてのアクセスユニットのTemporalId値の最小値に等しい。nal_unit_typeがPPS_NUTに等しいとき、TemporalIdは含んでいるアクセスユニットのTemporalId以上であることがあり、それは、すべてのピクチャパラメータセット(PPS)がビットストリームの最初に含まれ得るからであり、第1のコーディングされたピクチャは0に等しいTemporalIdを有する。nal_unit_typeがPREFIX_SEI_NUTまたはSUFFIX_SEI_NUTに等しいとき、TemporalIdは、含んでいるアクセスユニットのTemporalId以上であることがあり、それは、SEI NALユニットが、SEI NALユニットを含んでいるアクセスユニットのTemporalIdよりTemporalId値が大きいアクセスユニットを含むビットストリームサブセットに適用される情報

10

20

30

40

50

を含み得るからである。nuh_reserved_zero_7bitsは「0000000」に等しいものとする。nuh_reserved_zero_7bitsの他の値は、ITU-T | ISO/IECによって今後規定され得る。デコーダは、「0000000」に等しくないnuh_reserved_zero_7bitsの値を伴うNALユニットを無視する(すなわち、ビットストリームから削除して廃棄する)ものとする。

【0183】

シーケンスパラメータセットRBSPセマンティクス

【0184】

log2_max_pic_order_cnt_lsb_minus4は、ピクチャ順序カウンタのために復号プロセスにおいて使用される変数MaxPicOrderCntLsbの値を、 $\text{MaxPicOrderCntLsb} = 2(\log_2_max_pic_order_cnt_lsb_minus4 + 4)$ として指定する。log2_max_pic_order_cnt_lsb_minus4の値は、両端を含めて0から12の範囲にあるものとする。sps_max_dec_pic_buffering_minus1に1を足したものは、ピクチャストレージバッファの単位で、コーディングビデオシーケンス(CVS)のための復号ピクチャバッファの最大の必要とされるサイズを指定する。sps_max_dec_pic_buffering_minus1の値は、両端を含めて0からMaxDpbSize-1の範囲にあるものとし、MaxDpbSizeは他の箇所で指定されるようなものである。0に等しいlong_term_ref_pics_flagは、LTRPがCVSにおいていずれのコーディングされるピクチャのインター予測にも使用されないことを指定する。1に等しいlong_term_ref_pics_flagは、LTRPがCVSにおいて1つまたは複数のコーディングされるピクチャのインター予測のために使用され得ることを指定する。additional_lt_poc_lsbは、参照ピクチャリストのために復号プロセスにおいて使用される変数MaxLtPicOrderCntLsbの値を、 $\text{MaxLtPicOrderCntLsb} = 2(\log_2_max_pic_order_cnt_lsb_minus4 + 4 + \text{additional_lt_poc_lsb})$ として指定する。additional_lt_poc_lsbの値は、両端を含めて0から $32 - \log_2_max_pic_order_cnt_lsb_minus4 - 4$ の範囲にあるものとする。存在しないとき、additional_lt_poc_lsbの値は0に等しいと推測される。

【0185】

1に等しいall_rpl_entries_same_sign_flagは、各ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)の中のすべてのSTRPエントリが同じ符号値を有することを指定し、符号は正の値または負の値を示す。0に等しいall_rpl_entries_same_sign_flagは、各ref_pic_list_struct(listIdx, rplIdx, ltrpFlag)の中のSTRPエントリが同じ符号値を有することも有しないこともあることを指定する。1に等しいrpl1_copy_from_rpl0_flagは、num_ref_pic_lists_in_sps[1]およびref_pic_list_struct(1, rplIdx, ltrpFlag)が存在しないことを指定し、以下のことが当てはまる。num_ref_pic_lists_in_sps[1]の値は、num_ref_pic_lists_in_sps[0]の値に等しく設定される。シンタックス構造ref_pic_list_struct(1, rplIdx, ltrpFlag)は、ref_pic_list_struct(0, rplIdx, ltrpFlag)の値と同一であると推測される。続いて、ref_pic_list_struct(1, rplIdx, ltrpFlag)の中のシンタックス要素は、ref_pic_list_struct(0, rplIdx, ltrpFlag)の中のシンタックス要素に等しいとそれぞれ推測される。

【0186】

num_ref_pic_lists_in_sps[i]は、SPSに含まれる、listIdxがiに等しいref_pic_list_struct(listIdx, rplIdx, ltrpFlag)シンタックス構造の数を指定する。num_ref_pic_lists_in_sps[i]の値は、両端を含めて0から64の範囲にあるものとする。listIdx(0または1に等しい)の各値に対して、デコーダは、全体でnum_ref_pic_lists_in_sps[i]個のref_pic_list_struct(listIdx, rplIdx, ltrpFlag)シンタックス構造のためにメモリを割り振るべきであり、それは、現在のピクチャのスライスヘッダにおいて直接シグナリングされる1つのref_pic_list_struct(listIdx, rplIdx, ltrpFlag)シンタックス構造があり得るからである。

【0187】

ピクチャパラメータセットRBSPセマンティクス

【0188】

num_ref_idx_default_active_minus1[i]に1を足したものは、iが0に等しいとき、n

um_ref_idx_active_override_flagが0に等しいPまたはBスライスに対する変数NumRefIdxActive[0]の推測される値を指定し、iが1に等しいとき、num_ref_idx_active_override_flagが0に等しいBスライスに対するNumRefIdxActive[1]の推測される値を指定する。num_ref_idx_default_active_minus1[i]の値は、両端を含めて0から14の範囲にあるものとする。0に等しいrpl1_idx_present_flagは、ref_pic_list_sps_flag[1]およびref_pic_list_idx[1]がスライスヘッダに存在しないことを指定する。1に等しいrpl1_idx_present_flagは、ref_pic_list_sps_flag[1]およびref_pic_list_idx[1]がスライスヘッダに存在し得ることを指定する。

【 0 1 8 9 】

スライスヘッダセマンティクス

10

【 0 1 9 0 】

存在するとき、スライスヘッダシンタックス要素slice_pic_parameter_set_idおよびslice_pic_order_cnt_lsbの各々の値は、コーディングされたピクチャのすべてのスライスヘッダにおいて同じであるものとする。slice_typeはTable 7-3(表7)に従ってスライスのコーディングタイプを指定する。

【 0 1 9 1 】

【 表 7 】

Table 7-3 -slice_typeへの名前に関連付け

slice_type	slice_typeの名前
0	B (Bスライス)
1	P (Pスライス)
2	I (Iスライス)

20

【 0 1 9 2 】

nal_unit_typeがIRAP_NUTに等しい、すなわちピクチャがIRAPピクチャであるとき、slice_typeは2に等しいものとする。slice_pic_order_cnt_lsbは、現在のピクチャに対するピクチャ順序カウンタ modulo MaxPicOrderCntLsbを指定する。slice_pic_order_cnt_lsbシンタックス要素の長さは、log2_max_pic_order_cnt_lsb_minus4+4ビットである。slice_pic_order_cnt_lsbの値は、両端を含めて0からMaxPicOrderCntLsb-1の範囲にあるものとする。slice_pic_order_cnt_lsbが存在しないとき、slice_pic_order_cnt_lsbは0に等しいと推測される。1に等しいref_pic_list_sps_flag[i]は、現在のピクチャの参照ピクチャリストiが、アクティブSPSの中のlistIdxがiに等しいref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造のうちの1つに基づいて導出されることを指定する。0に等しいref_pic_list_sps_flag[i]は、現在のピクチャの参照ピクチャリストiが、現在のピクチャのスライスヘッダに直接含まれる、listIdxがiに等しいref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造に基づいて導出されることを指定する。num_ref_pic_lists_in_sps[i]が0に等しいとき、ref_pic_list_sps_flag[i]の値は0に等しいものとする。rpl1_idx_present_flagが0に等しくref_pic_list_sps_flag[0]が存在するとき、ref_pic_list_sps_flag[1]の値は、ref_pic_list_sps_flag[0]の値に等しいと推測される。ref_pic_list_idx[i]は、現在のピクチャの参照ピクチャリストiの導出のために使用されるlistIdxがiに等しいref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の、アクティブSPSに含まれるlistIdxがiに等しいref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造のリストへのインデックスを指定する。シンタックス要素ref_pic_list_idx[i]は、Ceil(Log2(num_ref_pic_lists_in_sps[i]))ビットによって表される。存在しないとき、ref_pic_list_idx[i]の値は0に等しいと推測される。ref_pic_list_idx[i]の値は、両端を含めて0からnum_ref_pic_lists_in_sps[i]-1の範囲にあるものとする。rpl1_idx_present_flagが0に等しく、ref_pic_list_sp

30

40

50

s_flag[0]が存在するとき、ref_pic_list_idx[1]の値は、ref_pic_list_idx[0]の値に等しいと推測される。1に等しいnum_ref_idx_active_override_flagは、シンタックス要素num_ref_idx_active_minus1[0]がPおよびBスライスに対して存在することと、シンタックス要素num_ref_idx_active_minus1[1]がBスライスに対して存在することとを指定する。0に等しいnum_ref_idx_active_override_flagは、シンタックス要素num_ref_idx_active_minus1[0]およびnum_ref_idx_active_minus1[1]が存在しないことを指定する。num_ref_idx_active_minus1[i]は、存在するとき、変数NumRefIdxActive[i]の値を、NumRefIdxActive[i]=num_ref_idx_active_minus1[i]+1として指定する。num_ref_idx_active_minus1[i]の値は、両端を含めて0から14の範囲にあるものとする。

【 0 1 9 3 】

NumRefIdxActive[i]-1の値は、スライスを復号するために使用され得る参照ピクチャリストiに対する最大参照インデックスを指定する。NumRefIdxActive[i]の値が0に等しいとき、参照ピクチャリストiに対する参照インデックスは、スライスを復号するために使用されないことがある。0または1に等しいiに対して、現在のスライスがBスライスであり、num_ref_idx_active_override_flagが0に等しいとき、NumRefIdxActive[i]は、num_ref_idx_default_active_minus1[i]+1に等しいと推測される。現在のスライスがPスライスであり、num_ref_idx_active_override_flagが0に等しいとき、NumRefIdxActive[0]は、num_ref_idx_default_active_minus1[0]+1に等しいと推測される。現在のスライスがPスライスであるとき、NumRefIdxActive[i]は0に等しいと推測される。現在のスライスがIスライスであるとき、NumRefIdxActive[0]とNumRefIdxActive[1]の両方が0に等しいと推測される。

【 0 1 9 4 】

代替として、0または1に等しいiに対して、上記の後で次のことが当てはまる。rplIdx1がref_pic_list_sps_flag[i]?ref_pic_list_idx[i]:num_ref_pic_lists_in_sps[i]に等しく設定されるものとし、numRpEntries[i]がnum_strp_entries[i][rplIdx1]+num_ltrp_entries[i][rplIdx1]に等しいものとする。NumRefIdxActive[i]がnumRpEntries[i]より大きいとき、NumRefIdxActive[i]の値はnumRpEntries[i]に等しく設定される。

【 0 1 9 5 】

参照ピクチャリスト構造セマンティクス

【 0 1 9 6 】

ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造は、SPSまたはスライスヘッダに存在し得る。シンタックス構造がスライスヘッダに含まれるかSPSに含まれるかに応じて、次のことが当てはまる。スライスヘッダに存在する場合、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造は、現在のピクチャ(スライスを含むピクチャ)の参照ピクチャリストlistIdxを指定する。それ以外(SPSに存在する)の場合、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造は、参照ピクチャリストlistIdxに対する候補を指定し、このセクションの残りにおいて指定されるセマンティクスにおける「現在のピクチャ」という用語は、1)SPSに含まれるref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造のリストへのインデックスに等しいref_pic_list_idx[listIdx]を含む1つまたは複数のスライスを有し、2)アクティブSPSとしてSPSを有するCVSの中にある、各ピクチャを指す。rpl_mode[listIdx][rplIdx]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中のシンタックス要素のコーディングモードを指定する。num_strp_entries[listIdx][rplIdx]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中のSTRPエントリの数を指定する。num_ltrp_entries[listIdx][rplIdx]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中のLTRPエントリの数を指定する。存在しないとき、num_ltrp_entries[listIdx][rplIdx]の値は、0に等しいと推測される。変数NumEntriesInList[listIdx][rplIdx]は、NumEntriesInList[listIdx][rplIdx]=num_strp_entries[listIdx][rplIdx]+num_ltrp_entries[listIdx][rplIdx]のように導出される。NumEntriesInList[listIdx][rplIdx]の値は、両端を含めて0からsps_max_dec_pic_buffering_minus1の範

10

20

30

40

50

囲にあるものとする。1に等しいstrp_entries_sign_flag[listIdx][rplIdx]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中のすべてのSTRPエントリが0以上の値を有することを指定する。0に等しいstrp_entries_sign_flag[listIdx][rplIdx]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中のすべてのSTRPエントリが0未満の値を有することを指定する。

【0197】

1に等しいlt_ref_pic_flag[listIdx][rplIdx][i]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中のi番目のエントリがLTRPエントリであることを指定する。0に等しいlt_ref_pic_flag[listIdx][rplIdx][i]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中のi番目のエントリがSTRPエントリであることを指定する。存在しないとき、lt_ref_pic_flag[listIdx][rplIdx][i]の値は0に等しいと推測される。両端を含めて0からNumEntriesInList[listIdx][rplIdx]-1の範囲にあるiのすべての値に対するlt_ref_pic_flag[listIdx][rplIdx][i]の合計は、num_ltrp_entries[listIdx][rplIdx]に等しくなければならないということが、ビットストリーム適合の要件である。1に等しいstrp_entry_sign_flag[listIdx][rplIdx][i]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中のi番目のエントリが0以上の値を有することを指定する。0に等しいstrp_entries_sign_flag[listIdx][rplIdx]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中のi番目のエントリが0未満の値を有することを指定する。存在しないとき、strp_entries_sign_flag[listIdx][rplIdx][i]の値は、strp_entries_sign_flag[listIdx][rplIdx]の値に等しいと推測される。

【0198】

delta_poc_st[listIdx][rplIdx][i]は、i番目のエントリがref_pic_list_struct(rplIdx,ltrpFlag)シンタックス構造の中の最初のSTRPエントリであるとき、現在のピクチャのピクチャ順序カウンタ値とi番目のエントリにより参照されるピクチャのピクチャ順序カウンタ値との差を指定し、または、i番目のエントリがSTRPエントリであるが、ref_pic_list_struct(rplIdx,ltrpFlag)シンタックス構造の中の最初のSTRPエントリではないとき、i番目のエントリにより参照されるピクチャのピクチャ順序カウンタ値とref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中の以前のSTRPエントリによって参照されるピクチャのピクチャ順序カウンタ値との差を指定する。delta_poc_st[listIdx][rplIdx][i]の値は、両端を含めて-215から215-1の範囲にあるものとする。poc_lsb_lt[listIdx][rplIdx][i]は、ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)シンタックス構造の中のi番目のエントリによって参照されるピクチャのピクチャ順序カウンタmodulo MaxLtPicOrderCntLsbの値を指定する。poc_lsb_lt[listIdx][rplIdx][i]シンタックス要素の長さは、Log2(MaxLtPicOrderCntLsb)ビットである。アレイDeltaPocSt[listIdx][rplIdx]は次のように導かれる。

```

for(i=0; i < NumEntriesInList[listIdx][rplIdx]; i++){
    if(!lt_ref_pic_flag[i][RplIdx][j]){
        DeltaPocSt[listIdx][rplIdx][i]=(strp_entry_sign_flag[listIdx][rplIdx][i])?
        delta_poc_st[listIdx][rplIdx][i]:0-
    }
}

```

【0199】

model_ref_rpl_idx_delta_minus1[listIdx][rplIdx]に1を足したものは、rplIdxの値と参照ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)のインデックスとの間の差を指定する。rpl_mode[listIdx][rplIdx]が1に等しいとき、変数RefRplIdxは次のように導かれる。

```

RefRplIdx=rplIdx-(model_ref_rpl_idx_delta_minus1[listIdx][rplIdx]+
1)

```

10

20

30

40

50

【 0 2 0 0 】

strp_offset_val_minus1[listIdx][rplIdx]に1を足したものは、現在のref_pic_list_struct(listIdx,rplIdx,ltrpFlag)のdelta_poc_st[listIdx][rplIdx][i]の値を計算するために、参照ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中の各STRPエントリから差し引かれるべき値を指定する。1に等しいref_entry_used_flag[listIdx][rplIdx][i]は、ref_pic_list_struct(0,RefRplIdx,ltrpFlag)の中のi番目のエントリがref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中のエントリとして使用されることを指定する。0に等しいref_entry_used_flag[listIdx][rplIdx]は、ref_pic_list_struct(0,RefRplIdx,ltrpFlag)の中のi番目のエントリがref_pic_list_struct(listIdx,rplIdx,ltrpFlag)の中のエントリとして使用されないことを指定する。

10

【 0 2 0 1 】

rpl_mode[listIdx][rplIdx]が1に等しいとき、シンタックス要素lt_ref_pic_flag[listIdx][rplIdx][i]の値の干渉に対して以下のことが当てはまり、lt_ref_pic_flag[listIdx][rplIdx][i]が1に等しいとき、poc_lsb_lt[listIdx][rplIdx][i]の値の干渉に対して、および変数DeltaPocSt[listIdx][rplIdx][i](lt_ref_pic_flag[listIdx][rplIdx][i]が0に等しいとき)および変数NumEntriesInList[listIdx][rplIdx]の導出に対して以下のことが当てはまる。

```

currEntryIdx=0
for(i=0;i < NumEntriesInList[0][RefRplIdx];i++){
    if(ref_entry_used_flag[listIdx][rplIdx][i]){
        lt_ref_pic_flag[listIdx][rplIdx][currEntryIdx]=lt_ref_
pic_flag[0][RefRplIdx][i]
        if(!lt_ref_pic_flag[0][RefRplIdx][i])
            DeltaPocSt[listIdx][rplIdx][currEntryIdx]=Delta
aPocSt[0][RefRplIdx][i]
        else
            poc_lsb_lt[listIdx][rplIdx][currEntryIdx]=poc_
lsb_lt[0][RefRplIdx][i]
        currEntryIdx++
    }
}
NumEntriesInList[listIdx][rplIdx]=currEntryIdx

```

20

30

【 0 2 0 2 】

mode2_ref_rpl_idx[listIdx][rplIdx]は、参照ref_pic_list_struct(listIdx,rplIdx,ltrpFlag)のインデックスを指定する。シンタックス要素mode2_ref_rpl_idx[listIdx][rplIdx]は、Ceil(Log2(num_ref_pic_lists_in_sps[0]))ビットによって表現される。rpl_mode[listIdx][rplIdx]が2に等しいとき、変数RefRplIdxは次のように導かれる。

```
RefRplIdx=mode2_ref_rpl_idx[listIdx][rplIdx]
```

【 0 2 0 3 】

num_additional_entries[listIdx][rplIdx]は、NumEntriesInList[listIdx][rplIdx]とNumEntriesInList[0][RefRplIdx]との間のデルタを指定する。add_lt_ref_pic_flag[listIdx][rplIdx][i]は、lt_ref_pic_flag[listIdx][rplIdx][NumEntriesInList[0][RefRplIdx]+i]の値の推測のために使用される。存在しないとき、add_lt_ref_pic_flag[listIdx][rplIdx][i]の値は、0に等しいと推測される。add_strp_entry_sign_flag[listIdx][rplIdx][i]は、strp_entry_sign_flag[listIdx][rplIdx][NumEntriesInList[0][RefRplIdx]+i]の値の推測のために使用される。存在しないとき、add_strp_entries_sign_flag[listIdx][rplIdx][i]の値は、strp_entries_sign_flag[listIdx][rplIdx]の値に等しくなるように設定される。

40

【 0 2 0 4 】

add_delta_poc_st[listIdx][rplIdx][i]は、delta_poc_st[listIdx][rplIdx][NumE

50

ntriesInList[0][RefRpIdx+i]の値の推測のために使用される。add_delta_poc_st[listIdx][rpIdx][i]の値は、両端を含めて-215から215-1の範囲にあるものとする。add_poc_lsb_lt[listIdx][rpIdx][i]は、poc_lsb_lt[listIdx][rpIdx][NumEntriesInList[0][RefRpIdx+i]の値の推測のために使用される。add_poc_lsb_lt[listIdx][rpIdx][i]シンタックス要素の長さは、Log2(MaxLtPicOrderCntLsb)ビットである。

【 0 2 0 5 】

rpl_mode[listIdx][rpIdx]が2に等しいとき、シンタックス要素strp_entries_sign_flag[listIdx][rpIdx]、lt_ref_pic_flag[listIdx][rpIdx][i]の値の推測のために以下のことが適用され、lt_ref_pic_flag[listIdx][rpIdx][i]が1に等しいとき、poc_lsb_lt[listIdx][rpIdx][i]の値の推測のために、ならびに、変数DeltaPocSt[listIdx][rpIdx][i](lt_ref_pic_flag[listIdx][rpIdx][i]が0に等しいとき)および変数NumEntriesInList[listIdx][rpIdx]の導出のために以下のことが適用される。

```

    strp_entries_sign_flag[listIdx][rpIdx]=strp_entries_sign_flag[0][RefRpIdx]
    for(i=0;i  NumEntriesInList[0][RefRpIdx];i++){
        lt_ref_pic_flag[listIdx][rpIdx][i]=lt_ref_pic_flag[0][RefRpIdx][i]
        if(!lt_ref_pic_flag[0][RefRpIdx][i])
            DeltaPocSt[listIdx][rpIdx][i]=DeltaPocSt[0][RefRpIdx][i]
        else
            poc_lsb_lt[listIdx][rpIdx][i]=poc_lsb_lt[0][RefRpIdx][i]
    }
    currEntryIdx=NumEntriesInList[0][RefRpIdx]
    for(i=0;i  num_additional_entries[listIdx][rpIdx];i++,currEntryIdx++)
    {
        lt_ref_pic_flag[listIdx][rpIdx][currEntryIdx]=add_lt_ref_pic_flag[listIdx][rpIdx][i]
        if(!add_lt_ref_pic_flag[listIdx][rpIdx][i])
            DeltaPocSt[listIdx][rpIdx][currEntryIdx]=(add_strp_entry_sign_flag[listIdx][rpIdx][i])?
                add_delta_poc_st[listIdx][rpIdx][i]: 0-add_delta_poc_st[listIdx][rpIdx][i]
        else
            poc_lsb_lt[listIdx][rpIdx][currEntryIdx]=add_poc_lsb_lt[listIdx][rpIdx][i]
    }
    NumEntriesInList[listIdx][rpIdx]=currEntryIdx

```

【 0 2 0 6 】

mode3_ref_rpl_idx[listIdx][rpIdx]は、参照ref_pic_list_struct(listIdx,rpIdx,itrpFlag)のインデックスを指定する。シンタックス要素mode3_ref_rpl_idx[listIdx][rpIdx]は、Ceil(Log2(num_ref_pic_lists_in_sps[0]))ビットによって表現される。rpl_mode[listIdx][rpIdx]が3に等しいとき、変数RefRpIdxは次のように導かれる。

RefRpIdx=mode3_ref_rpl_idx[listIdx][rpIdx]

【 0 2 0 7 】

rpl_mode[listIdx][rpIdx]が3に等しいとき、シンタックス要素lt_ref_pic_flag[listIdx][rpIdx][i]の値の推測のために以下のことが適用され、lt_ref_pic_flag[listIdx][rpIdx][i]が1に等しいとき、poc_lsb_lt[listIdx][rpIdx][i]の値の推測のために以下のことが適用され、変数DeltaPocSt[listIdx][rpIdx][i](lt_ref_pic_flag[listIdx][rpIdx][i]が0に等しいとき)および変数NumEntriesInList[listIdx][rpIdx]の導出のために以下のことが適用される。

sIdx][i]が0に等しいとき)および変数NumEntriesInList[listIdx][rplIdx]の導出のために以下のことが適用される。

```

    for(i=0;i < NumEntriesInList[0][RefRplIdx];i++){
        lt_ref_pic_flag[listIdx][rplIdx][i]=lt_ref_pic_flag[0][RefRplIdx][i]
        if(!lt_ref_pic_flag[0][RefRplIdx][i])
            DeltaPocSt[listIdx][rplIdx][i]=0-DeltaPocSt[0][RefRplIdx][i]
        else
            poc_lsb_lt[listIdx][rplIdx][i]=poc_lsb_lt[0][RefRplIdx][i]
    }
    NumEntriesInList[listIdx][rplIdx]=NumEntriesInList[0][RefRplIdx]

```

【0208】

一般的な復号プロセスが提供される。

【0209】

現在のピクチャCurrPicに対して、復号プロセスは次のように動作する。NALユニットの復号が以下で規定される。以下のプロセスは、スライスヘッダレイヤの中の、および上記のシンタックス要素を使用して、次の復号プロセスを規定する。ピクチャ順序カウントに関する変数および関数が導出される。これは、ピクチャの最初のスライスだけに対して呼び出される。非IRAPピクチャの各スライスに対する復号プロセスの最初において、参照ピクチャリスト構築のための復号プロセスは、参照ピクチャリスト0(RefPicList[0])および参照ピクチャリスト1(RefPicList[1])の導出のために呼び出される。参照ピクチャマーキングのための復号プロセスが呼び出され、参照ピクチャは、「参照のために使用されない」または「長期参照のために使用される」とマークされ得る。これは、ピクチャの最初のスライスだけに対して呼び出される。コーディングツリーユニット、スケーリング、変換、ループ内フィルタリングなどのための復号プロセスが呼び出される。現在のピクチャのすべてのスライスが復号された後で、現在の復号されるピクチャは「短期参照のために使用される」とマークされる。

【0210】

NALユニット復号プロセスが提供される。

【0211】

このプロセスへの入力、現在のピクチャのNALユニットおよび関連する非VCL NALユニットである。このプロセスの出力は、NALユニット内に封入される構文解析されたRBS Pシンタックス構造である。各NALユニットのための復号プロセスは、NALユニットからRBSPシンタックス構造を抽出し、次いでRBSPシンタックス構造を構文解析する。

【0212】

スライス復号プロセスが提供される。

【0213】

ピクチャ順序カウントのための復号プロセスは次のとおりである。

【0214】

このプロセスの出力は、現在のピクチャのピクチャ順序カウントであるPicOrderCntValである。マージモードにおける動きパラメータの導出および動きベクトルの予測のために、ならびにデコーダ適合性確認のために、ピクチャ順序カウントが、ピクチャを特定するために使用される。各々のコーディングされたピクチャは、PicOrderCntValと表記されるピクチャ順序カウント変数と関連付けられる。現在のピクチャがIRAPピクチャではないとき、変数prevPicOrderCntLsbおよびprevPicOrderCntMsbは次のように導出される。prevTid0Picを、0に等しいTemporalIdを有する、復号順序において以前のピクチャとする。変数prevPicOrderCntLsbは、prevTid0Picのslice_pic_order_cnt_lsbに等しく設定される。変数prevPicOrderCntMsbは、prevTid0PicのPicOrderCntMsbに等

10

20

30

40

50

しく設定される。

【0215】

現在のピクチャの変数PicOrderCntMsbは次のように導出される。現在のピクチャがIRAPピクチャである場合、PicOrderCntMsbは0に等しく設定される。それ以外の場合、PicOrderCntMsbは次のように導出される。

```
if((slice_pic_order_cnt_lsb < prevPicOrderCntLsb)&&
    ((prevPicOrderCntLsb-slice_pic_order_cnt_lsb) <=(MaxPicOrderCntLsb/2)))
```

```
    PicOrderCntMsb=prevPicOrderCntMsb+MaxPicOrderCntLsb
```

```
else if((slice_pic_order_cnt_lsb < prevPicOrderCntLsb)&&
```

```
    ((slice_pic_order_cnt_lsb-prevPicOrderCntLsb) < (MaxPicOrderCntLsb/2)))
```

```
    PicOrderCntMsb=prevPicOrderCntMsb-MaxPicOrderCntLsb
```

```
else
```

```
    PicOrderCntMsb=prevPicOrderCntMsb
```

【0216】

PicOrderCntValは次のように導出される。

```
PicOrderCntVal=PicOrderCntMsb+slice_pic_order_cnt_lsb
```

【0217】

すべてのIRAPピクチャが0に等しいPicOrderCntValを有し、それは、slice_pic_order_cnt_lsbがIRAPピクチャに対して0に等しいと推測され、prevPicOrderCntLsbとprevPicOrderCntMsbがともに0に等しく設定されるからである。PicOrderCntValの値は、両端を含めて-231から231-1の範囲にあるものとする。1つのCVSにおいて、任意の2つのコーディングされたピクチャに対するPicOrderCntVal値は、同じではないものとする。

【0218】

復号プロセスの間の任意の瞬間において、DPBの中の任意の2つの参照ピクチャに対するPicOrderCntVal&(MaxLtPicOrderCntLsb-1)の値は同じではないものとする。関数PicOrderCnt(picX)は、PicOrderCnt(picX)=ピクチャpicXのPicOrderCntValのように規定される。関数DiffPicOrderCnt(picA,picB)は、DiffPicOrderCnt(picA,picB)=PicOrderCnt(picA)-PicOrderCnt(picB)のように規定される。ビットストリームは、両端を含めて-215から215-1の範囲にない復号プロセスにおいて使用されるDiffPicOrderCnt(picA,picB)の値をもたらすデータを含まないものとする。Xを現在のピクチャとし、YおよびZを同じCVSの中の2つの他のピクチャであるものとする、YおよびZは、DiffPicOrderCnt(X,Y)およびDiffPicOrderCnt(X,Z)がともに正であるとき、またはともに負であるとき、Xから同じ出力順序方向にあると考えられる。

【0219】

参照ピクチャリスト構築のための復号プロセスが提供される。

【0220】

このプロセスは、非IRAPピクチャの各スライスに対する復号プロセスの最初に呼び出される。参照ピクチャは参照インデックスを通じてアドレス指定される。参照インデックスは参照ピクチャリストへのインデックスである。Iスライスを復号するとき、スライスデータの復号の際に参照ピクチャリストは使用されない。Pスライスを復号するとき、参照ピクチャリスト0(すなわち、RefPicList[0])だけがスライスデータの復号の際に使用される。Bスライスを復号するとき、参照ピクチャリスト0と参照ピクチャリスト1(すなわち、RefPicList[1])の両方がスライスデータの復号の際に使用される。非IRAPピクチャの各スライスに対する復号プロセスの最初において、参照ピクチャリストRefPicList[0]およびRefPicList[1]が導出される。参照ピクチャリストは、参照ピクチャのマーキングまたはスライスデータの復号において使用される。ピクチャの最初のスライスではない非IRAPピクチャのIスライスに対して、RefPicList[0]およびRefPicList[1]が、ビットストリーム適合確認の目的で導出されてもよいが、それらの導出は、現在のピクチャまたは復号順序に

10

20

30

40

50

において現在のピクチャの後にあるピクチャの復号には必要ではない。ピクチャの最初のスライスではないPスライスに対して、RefPicList[1]が、ビットストリーム適合性確認の目的で導出され得るが、その導出は、現在のピクチャまたは復号順序において現在のピクチャの後にあるピクチャの復号には必要ではない。参照ピクチャリストRefPicList[0]およびRefPicList[1]は次のように構築される。

```

    for(i=0;i<2;i++){
        if(ref_pic_list_sps_flag[i])
            RplIdx[i]=ref_pic_list_idx[i]
        else
            RplIdx[i]=num_ref_pic_lists_in_sps[i]
        for(j=0,pocBase=PicOrderCntVal;j<NumEntriesInList[i][RplIdx[i]];j++){
            if(!lt_ref_pic_flag[i][RplIdx[i]][j]){
                RefPicPocList[i][j]=pocBase-DeltaPocSt[i][RplIdx[i]][j]
                if(PicOrderCntValがRefPicPocList[i][j]に等しい
                参照ピクチャpicAがDPBの中にある)
                    RefPicList[i][j]=picA
                else
                    RefPicList[i][j]="参照ピクチャなし"
                pocBase=RefPicPocList[i][j]
            }else{
                if(PicOrderCntVal&(MaxLtPicOrderCntLsb-1)が
                poc_lsb_lt[i][RplIdx[i]][j]に等しい参照picAがDPBの中にある)
                    RefPicList[i][j]=picA
                else
                    RefPicList[i][j]="参照ピクチャなし"
            }
        }
    }
}

```

【0221】

0または1に等しい各*i*に対して、次のことが当てはまる。RefPicList[*i*]の中の最初のNumRefIdxActive[*i*]エントリは、RefPicList[*i*]の中のアクティブエントリと呼ばれ、RefPicList[*i*]の中の他のエントリは、RefPicList[*i*]の中の非アクティブエントリと呼ばれる。両端を含めて、0からNumEntriesInList[*i*][RplIdx[*i*]-1]の範囲にある*j*に対するRefPicList[*i*][*j*]の中の各エントリは、lt_ref_pic_flag[*i*][RplIdx[*i*]][*j*]が0に等しい場合STRPエントリと呼ばれ、それ以外の場合LTRPエントリと呼ばれる。特定のピクチャが、RefPicList[0]の中のエントリとRefPicList[1]の中のエントリの両方によって参照されることがあり得る。特定のピクチャが、RefPicList[0]の中の1つより多くのエントリまたはRefPicList[1]の中の1つより多くのエントリによって参照されることもあり得る。RefPicList[0]の中のアクティブエントリおよびRefPicList[1]の中のアクティブエントリは集合的に、現在のピクチャのインター予測のために使用され得るすべての参照ピクチャと、復号順序において現在のピクチャの後にある1つまたは複数のピクチャとを指す。RefPicList[0]の中の非アクティブエントリおよびRefPicList[1]の中の非アクティブエントリは、現在のピクチャのインター予測のために使用されないが復号順序において現在のピクチャの後にある1つまたは複数のピクチャのためのインター予測において使用され得る、すべての参照ピクチャを集合的に指す。対応するピクチャがDPBの中に存在しないので、「参照ピクチャなし」に等しい1つまたは複数のエントリがRefPicList[0]またはRefPicList[1]の中にあることがある。「参照ピクチャなし」に等しいRefPicList[0]またはRefPicList[0]の中の各非アクティブエントリは無視されるべきである。「参照ピクチャなし」に等し

いRefPicList[0]またはRefPicList[1]の中の各アクティブエントリに対して、意図しないピクチャの喪失が推測されるべきである。

【0222】

次の制約が適用されることが、ビットストリーム適合の要件である。0または1に等しい各*i*に対して、NumEntriesInList[*i*][RplIdx[*i*]]はNumRefIdxActive[*i*]未満ではないものとする。RefPicList[0]またはRefPicList[1]の中の各アクティブエントリによって参照されるピクチャは、DPBに存在するものとし、現在のピクチャ以下のTemporalIdを有するものとする。任意選択で、次の制約がさらに規定され得る。RefPicList[0]またはRefPicList[1]の中のいずれの非アクティブエントリのエントリインデックスも、現在のピクチャの復号のための参照インデックスとして使用されないものとする。任意選択で、次の制約がさらに規定され得る。RefPicList[0]またはRefPicList[1]の中の非アクティブエントリは、RefPicList[0]またはRefPicList[1]の中の他のエントリと同じピクチャを参照しないものとする。ピクチャのスライスのRefPicList[0]またはRefPicList[1]の中のSTRPエントリ、および同じスライスまたは同じピクチャの異なるスライスのRefPicList[0]もしくはRefPicList[1]の中のLTRPエントリは、同じピクチャを参照しないものとする。現在のピクチャ自体が、RefPicList[0]またはRefPicList[1]の中のいずれのエントリによっても参照されないものとする。現在のピクチャのPicOrderCntValとエントリによっても参照されるピクチャのPicOrderCntValとの差が224以上であるLTRPエントリは、RefPicList[0]またはRefPicList[1]の中になくする。setOfRefPicsを、RefPicList[0]の中のすべてのエントリおよびRefPicList[1]の中のすべてのエントリによって参照される固有のピクチャのセットとする。setOfRefPicsの中のピクチャの数はsps_max_dec_pic_buffering_minus1以下であるものとし、setOfRefPicsはピクチャのすべてのスライスに対して同じであるものとする。

【0223】

参照ピクチャマーキングのための復号プロセスが提供される。

【0224】

このプロセスは、スライスヘッダの復号と、スライスに対する参照ピクチャリスト構築のための復号プロセスの後で、しかしスライスデータの復号の前に、ピクチャごとに一度呼び出される。このプロセスにより、DPBの中の1つまたは複数の参照ピクチャが、「参照のために使用されない」または「長期参照のために使用される」とマークされるようになり得る。DPBの中の復号されたピクチャは、復号プロセスの動作の間の任意の所与の瞬間において、「参照のために使用されない」、「短期参照のために使用される」、または「長期参照のために使用される」ものとして、しかしこれら3つのうちの1つだけとしてマークされ得る。これらのマーキングのうちの1つをピクチャに割り当てることは、適用可能なときこれらのマーキングのうちの別のものを暗黙的に排除する。ピクチャが「参照のために使用される」とマークされるものとして参照されるとき、これは、ピクチャが「短期参照のために使用される」または「長期参照のために使用される」(しかし両方ではない)とマークされることを集合的に指す。現在のピクチャがIRAPピクチャであるとき、現在DPBの中にあるすべての参照ピクチャ(もしあれば)が、「参照のために使用されない」とマークされる。STRPは、それらのPicOrderCntVal値によって特定される。LTRPは、それらのPicOrderCntVal値の $\text{Log}_2(\text{MaxLtPicOrderCntLsb})$ 個のLSBによって特定される。以下のことが適用される。RefPicList[0]またはRefPicList[1]の中の各LTRPエントリに対して、参照されるピクチャがSTRPであるとき、ピクチャは「長期参照のために使用される」とマークされる。RefPicList[0]またはRefPicList[1]の中のいずれのエントリによっても参照されないDPBの中の各参照ピクチャは、「参照のために使用されない」とマークされる。

【0225】

図9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900(たとえば、ビデオエンコーダ20またはビデオデコーダ30)の概略図である。ビデオコーディングデバイス900は、本明細書で説明されるような開示された実施形態を実施するのに適している。ピ

10

20

30

40

50

デオコーディングデバイス900は、データを受信するための入口ポート910および受信機ユニット(Rx)920、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)930、データを送信するための送信機ユニット(Tx)940および出口ポート950、ならびにデータを記憶するためのメモリ960を備える。ビデオコーディングデバイス900は、光または電気信号の出入のために、出口ポート910、受信機ユニット920、送信機ユニット940、および出口ポート950に結合された、光-電子(OE)コンポーネントおよび電子-光(EO)コンポーネントも備え得る。

【0226】

プロセッサ930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930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)、フィールドプログラマブルゲートアレイ(FPGA)、特定用途向け集積回路(ASIC)、およびデジタルシグナルプロセッサ(DSP)として実装され得る。プロセッサ930は、入口ポート910、受信機ユニット920、送信機ユニット940、出口ポート950、およびメモリ960と通信している。プロセッサ930はコーディングモジュール970を備える。コーディングモジュール970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970は、様々なネットワーク機能を実装し、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970を含むことは、ビデオコーディングデバイス900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900の変換を生み出す。代替として、コーディングモジュール970は、メモリ960に記憶されておりプロセッサ930によって実行される命令として実装される。

【0227】

ビデオコーディングデバイス900はまた、ユーザとの間でデータをやり取りするための入力および/または出力(I/O)デバイス980を含み得る。I/Oデバイス980は、ビデオデータを表示するためのディスプレイ、オーディオデータを出力するためのスピーカーなどの、出力デバイスを含み得る。I/Oデバイス980は、キーボード、マウス、トラックボールなどの入力デバイス、および/またはそのような出力デバイスとインターフェースするための対応するインターフェースも含み得る。

【0228】

メモリ960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、そのようなプログラムが実行のために選択されるときにプログラムを記憶するための、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の、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得る。メモリ960は、揮発性および/または不揮発性であってもよく、読取り専用メモリ(ROM)、ランダムアクセスメモリ(RAM)、三項コンテンツアドレス指定可能メモリ(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。

【0229】

図10は、コーディングのための手段1000のある実施形態の概略図である。実施形態において、コーディングのための手段1000は、ビデオコーディングデバイス1002(たとえば、ビデオエンコーダ20またはビデオデコーダ30)において実装される。ビデオコーディングデバイス1002は受信手段1001を含む。受信手段1001は、符号化すべきピクチャを受信し、または復号すべきビットストリームを受信するように構成される。ビデオコーディングデバイス1002は、受信手段1001に結合された送信手段1007を含む。送信手段1007は、ビットストリームをデコーダに送信し、または復号された画像を表示手段(たとえば、I/Oデバイス1080のうちの1つ)に送信するように構成される。

【0230】

ビデオコーディングデバイス1002は記憶手段1003を含む。記憶手段1003は、受信手段1001または送信手段1007のうちの少なくとも1つに結合される。記憶手段1003は命令を記憶するように構成される。ビデオコーディングデバイス1002はまた処理手段1005を含む。処理手段1005は記憶手段1003に結合される。処理手段1005は、本明細書で開示される方法を実行するために、記憶手段1003に記憶されている命令を実行するように

構成される。

【0231】

本明細書に記載される例示的な方法のステップは、必ずしも説明される順序で実行される必要はなく、そのような方法のステップの順序は、単に例示的であるものとして理解されるべきであることも理解されたい。同様に、追加のステップがそのような方法に含まれてもよく、本開示の様々な実施形態と一貫した方法で、いくつかのステップが省略されてもよく、または組み合わせられてもよい。

【0232】

いくつかの実施形態が本開示において提供されたが、開示されるシステムおよび方法は、本開示の趣旨または範囲から逸脱することなく多くの他の具体的な形式で具現化され得ることを理解されたい。これらの例は、限定ではなく例示であると見なされるべきであり、意図は本明細書で与えられる詳細に限定されないものとする。たとえば、別のシステムでは様々な要素およびコンポーネントが組み合わせられ、もしくは統合されてもよく、またはいくつかの特徴が省略され、または実装されなくてもよい。

10

【0233】

加えて、様々な実施形態において個別または別個のものとして説明され例示されている技法、システム、サブシステム、および方法は、本開示の範囲から逸脱することなく、他のシステム、モジュール、技法、もしくは方法と組み合わせられ、または統合され得る。結合される、または直接結合される、または互いに通信するものとして、示されもしくは論じられた他の項目は、電氣的に、機械的に、または別様にのいずれでも、何らかのインターフェース、デバイス、または中間コンポーネントを通じて、間接的に結合され、または通信していてもよい。変更、置換、および変形の他の例が、当業者により認識可能であり、本明細書で開示される趣旨および範囲から逸脱することなく行われ得る。

20

【符号の説明】

【0234】

- 12 ソースデバイス
- 14 デスティネーションデバイス
- 16 コンピュータ可読媒体
- 18 ビデオソース
- 20 ビデオエンコーダ
- 22 出力インターフェース
- 28 入力インターフェース
- 30 ビデオデコーダ
- 32 表示デバイス
- 40 モード選択ユニット
- 42 動き推定ユニット
- 44 動き補償ユニット
- 46 イントラ予測ユニット
- 48 区分ユニット
- 52 変換処理ユニット
- 54 量子化ユニット
- 56 エントロピーコーディングユニット
- 58 逆量子化ユニット
- 60 逆変換ユニット
- 64 参照フレームメモリ
- 70 エントロピー復号ユニット
- 72 動き補償ユニット
- 74 イントラ予測ユニット
- 76 逆量子化ユニット
- 78 逆変換ユニット

30

40

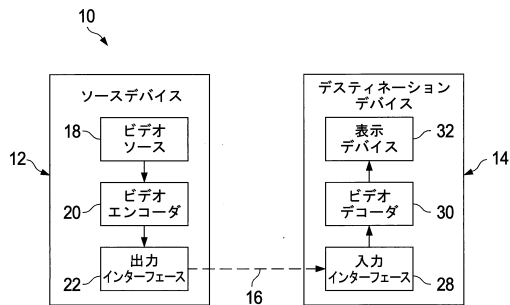
50

- 82 参照フレームメモリ
- 510 SPS
- 512 PPS
- 514 スライスヘッダ
- 520 画像データ
- 910 入口ポート
- 920 受信機ユニット
- 930 プロセッサ
- 940 送信機ユニット
- 950 出口ポート
- 960 メモリ
- 970 コーディングモジュール
- 980 I/Oデバイス
- 1001 受信手段
- 1002 ビデオコーディングデバイス
- 1003 記憶手段
- 1005 処理手段
- 1007 送信手段

10

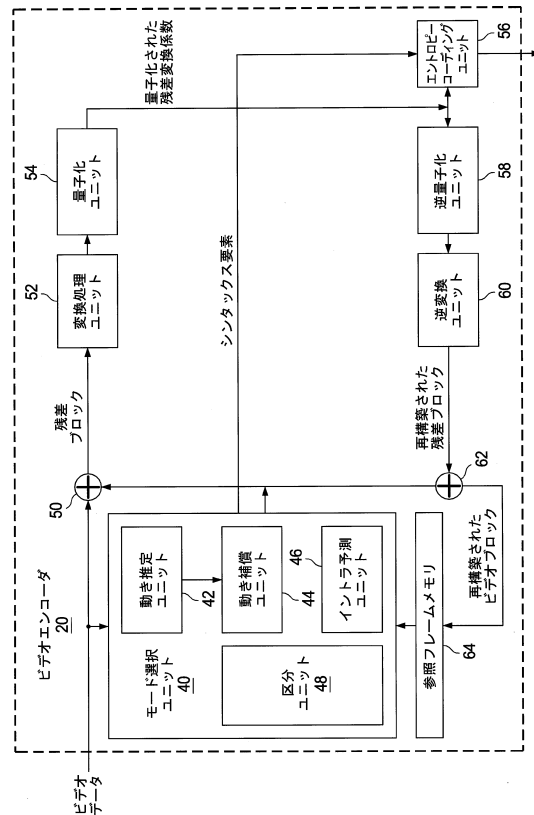
【図面】

【図 1】



【図 2】

20

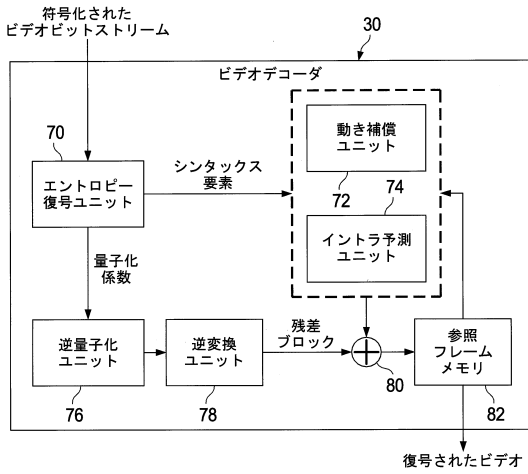


30

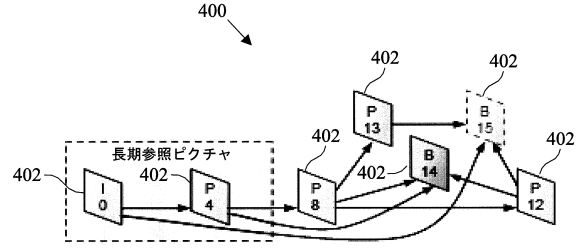
40

50

【図3】

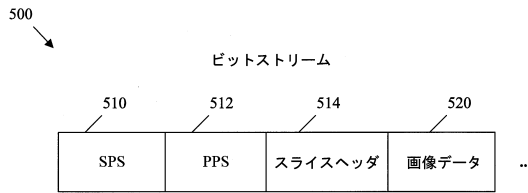


【図4】

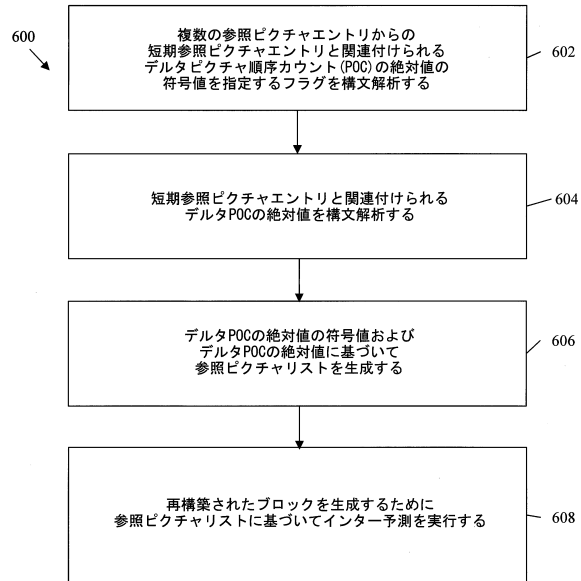


10

【図5】



【図6】



20

30

40

50

フロントページの続き

(33)優先権主張国・地域又は機関

米国(US)

前置審査

ロフト・プレイス・8584

(72)発明者 イェ・クイ・ワン

アメリカ合衆国・カリフォルニア・92130・サン・ディエゴ・サンローズ・クレスト・ウェイ
・6264

審査官 岩井 健二

(56)参考文献

特開2014-011586(JP, A)

米国特許出願公開第2013/0114742(US, A1)

国際公開第2013/157814(WO, A1)

国際公開第2013/002700(WO, A1)

Miska M. Hannuksela, AHG21: On reference picture list construction and reference picture marking, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-G643r1, 7th Meeting: Geneva, CH, 2011年11月, pp.1-17
Ye-Kui Wang, and Hendry, On reference picture management for VVC, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0112-v4, 12th Meeting: Macao, CN, 2018年10月, pp.1-12

(58)調査した分野 (Int.Cl., DB名)

H04N 19/00 - 19/98