

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7302000号
(P7302000)

(45)発行日 令和5年7月3日(2023.7.3)

(24)登録日 令和5年6月23日(2023.6.23)

(51)国際特許分類		F I	
H 0 4 N	19/85 (2014.01)	H 0 4 N	19/85
H 0 4 N	19/102 (2014.01)	H 0 4 N	19/102
H 0 4 N	19/162 (2014.01)	H 0 4 N	19/162
H 0 4 N	19/172 (2014.01)	H 0 4 N	19/172
請求項の数 17 (全58頁)			
(21)出願番号	特願2021-555265(P2021-555265)	(73)特許権者	504161984
(86)(22)出願日	令和2年3月11日(2020.3.11)		ホアウェイ・テクノロジーズ・カンパニ
(65)公表番号	特表2022-526088(P2022-526088		ー・リミテッド
	A)		中華人民共和国・5 1 8 1 2 9・グァン
(43)公表日	令和4年5月23日(2022.5.23)		ドン・シェンツェン・ロンガン・ディス
(86)国際出願番号	PCT/US2020/022185		トリクト・バンティアン・(番地なし)
(87)国際公開番号	WO2020/185962		・ホアウェイ・アドミニストレーション
(87)国際公開日	令和2年9月17日(2020.9.17)		・ビルディング
審査請求日	令和3年11月11日(2021.11.11)	(74)代理人	100110364
(31)優先権主張番号	62/816,722		弁理士 実広 信哉
(32)優先日	平成31年3月11日(2019.3.11)	(74)代理人	100133569
(33)優先権主張国・地域又は機関			弁理士 野村 進
	米国(US)	(72)発明者	フヌ・ヘンドリー
(31)優先権主張番号	62/871,020		アメリカ合衆国・カリフォルニア・9 2
(32)優先日	令和1年7月5日(2019.7.5)		1 2 9・サン・ディエゴ・フォックスク
最終頁に続く		最終頁に続く	

(54)【発明の名称】 エンコーダ、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、

第1のフラグに対する値が外部手段によって提供されるかどうかを前記ビデオデコーダによって決定するステップと、

前記第1のフラグに対する前記値が前記外部手段によって提供されるとき、コーディングされた漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、前記第1のフラグを、前記外部手段によって提供される前記値に等しく、かつ第2のフラグを前記第1のフラグに等しく、前記ビデオデコーダによって設定するステップと、

前記コーディングされたGDRピクチャを前記ビデオデコーダによって復号するステップと

を含む、方法。

【請求項 2】

前記第1のフラグに対する前記値が前記外部手段によって提供されるとき、前記コーディングされた漸進的復号リフレッシュ(GDR)ピクチャおよび出力順序で前記コーディングされたGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、前記第1のフラグを、前記外部手段によって提供される前記値に等しく、かつ前記第2のフラグを前記第1のフラグに等しく設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】

前記外部手段は、前記ビデオデコーダのグラフィックユーザインターフェース(GUI)であり、前記第1のフラグの前記値は、前記外部手段を使用して前記ビデオデコーダのユーザによって提供される、請求項1または2に記載の方法。

【請求項 4】

前記第1のフラグはHandleGdrAsCvsStartFlagと指定される、請求項1から3のいずれか一項に記載の方法。

【請求項 5】

前記コーディングされたGDRピクチャおよび出力順序で前記コーディングされたGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、前記第1のフラグおよび前記第2のフラグの前記値は1に設定される、請求項1から4のいずれか一項に記載の方法。

10

【請求項 6】

前記第1のフラグに対する前記値が前記外部手段によって提供されないとき、前記第1のフラグの前記値は0に設定される、請求項1に記載の方法。

【請求項 7】

復号デバイスであって、
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た受信機と、
前記受信機に結合されたメモリであって、命令を記憶する、メモリと、
前記メモリに結合されたプロセッサとを備え、前記プロセッサは、前記復号デバイスに、
第1のフラグに対する値が外部手段によって提供されるかどうかを決定することと、
前記第1のフラグに対する前記値が前記外部手段によって提供されるとき、コーディングされた漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、前記第1のフラグを、前記外部手段によって提供される前記値に等しく、かつ第2のフラグを前記第1のフラグに等しく設定することと、
前記コーディングされたGDRピクチャを復号することと
をさせるために、前記命令を実行するように構成される、
復号デバイス。

20

【請求項 8】

前記第1のフラグに対する前記値が前記外部手段によって提供されるとき、コーディングされた漸進的復号リフレッシュ(GDR)ピクチャおよび出力順序で前記コーディングされたGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、前記第1のフラグは前記外部手段によって提供される前記値に等しく、かつ第2のフラグは前記第1のフラグに等しく設定される、請求項7に記載の復号デバイス。

30

【請求項 9】

前記外部手段は、ビデオデコーダのグラフィックユーザインターフェース(GUI)であり、前記第1のフラグの前記値は、前記外部手段を使用して前記ビデオデコーダのユーザによって提供される、請求項7または8に記載の復号デバイス。

【請求項 10】

前記第1のフラグはHandleGdrAsCvsStartFlagと指定される、請求項7から9のいずれか一項に記載の復号デバイス。

40

【請求項 11】

前記コーディングされたGDRピクチャおよび出力順序で前記コーディングされたGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、前記第1のフラグおよび前記第2のフラグの前記値は1に設定される、請求項7から10のいずれか一項に記載の復号デバイス。

【請求項 12】

前記第1のフラグに対する前記値が前記外部手段によって提供されないとき、前記第1のフラグの前記値は0に設定される、請求項7に記載の復号デバイス。

50

【請求項 13】

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た受信機と、
前記受信機に結合された送信機であって、復号画像をディスプレイへ送信するように構成された、送信機と、
前記受信機または前記送信機のうちの少なくとも1つに結合されたメモリであって、命令を記憶するように構成された、メモリと、
前記メモリに結合されたプロセッサであって、請求項1から6のいずれか一項に記載の方法を実行するために、前記メモリの中に記憶された前記命令を実行するように構成された、プロセッサと
を備える、コーディング装置。

10

【請求項 14】

画像を表示するように構成されたディスプレイをさらに備える、請求項13に記載のコーディング装置。

【請求項 15】

エンコーダと、
前記エンコーダと通信しているデコーダであって、請求項7から14のいずれか一項に記載の復号デバイスまたはコーディング装置を含む、デコーダと
を備える、システム。

【請求項 16】

符号化すべきピクチャを受信するか、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た、受信手段と、
前記受信手段に結合された送信手段であって、前記ビットストリームを復号手段へ送信するか、または復号画像を表示手段へ送信するように構成された、送信手段と、
前記受信手段または前記送信手段のうちの少なくとも1つに結合された記憶手段であって、命令を記憶するように構成された、記憶手段と、
前記記憶手段に結合された処理手段であって、請求項1から6のいずれか一項に記載の方法を実行するために、前記記憶手段の中に記憶された前記命令を実行するように構成された、処理手段と
を備える、コーディングするための手段。

20

【請求項 17】

第1のフラグに対する値が外部手段によって提供されるかどうかを決定することと、前記第1のフラグに対する前記値が前記外部手段によって提供されるとき、コーディングされた漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、前記第1のフラグを、前記外部手段によって提供される前記値に等しく、かつ第2のフラグを前記第1のフラグに等しく設定することとを行うように構成された、決定ユニットと、
前記コーディングされたGDRピクチャを復号するように構成された復号ユニットと
を備える、デコーダ。

30

【発明の詳細な説明】

【技術分野】

【0002】

一般に、本開示は、ビデオコーディングにおける漸進的復号リフレッシュをサポートする技法を説明する。より詳細には、本開示は、イントラランダムアクセスポイント(Intra random access point)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする。

40

【背景技術】

【0003】

比較的短いビデオを描くのに必要とされるビデオデータの量でさえ相当であり得、そのことは、帯域幅容量が限定された通信ネットワークを横断してデータがストリーミングまたは別の方法で通信されることになるときに困難をもたらす場合がある。したがって、ビデオデータは、一般に、現代の電気通信ネットワークを横断して通信される前に圧縮され

50

る。メモリリソースが限定されることがあるので、ビデオが記憶デバイス上に記憶されるときも、ビデオのサイズが問題であり得る。ビデオ圧縮デバイスは、しばしば、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、送信または記憶の前にビデオデータをコーディングし、それによって、デジタルビデオ画像を描写するために必要とされるデータの量を減らす。圧縮されたデータは、次いで、ビデオデータを復号するビデオ復元デバイスによって宛先において受信される。ネットワークリソースが限定され、より高いビデオ品質の需要が絶えず増大すると、画像品質における犠牲をほとんど伴わずに圧縮率を改善する、改善された圧縮および復元技法が望ましい。

【発明の概要】

【課題を解決するための手段】

【0004】

第1の態様は、ビデオデコード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、第1のフラグに対する値が外部入力によって提供されるかどうかをビデオデコードによって決定するステップと、第1のフラグに対する値が外部入力によって提供されるとき、コーディングされたビデオビットストリームの中の漸進的復号リフレッシュ(GDR: gradual decoding refresh)ピクチャが出力されることを防止するために、第1のフラグを、外部入力によって提供される値に等しく、かつ第2のフラグを第1のフラグに等しく、ビデオデコードによって設定するステップと、GDRピクチャをビデオデコードによって復号するステップと、ビデオデコードによってGDRピクチャを復号ピクチャバッファ(DPB: decoded picture buffer)の中に記憶するステップとを含む。

【0005】

本方法は、イントラランダムアクセスポイント(IRAP)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする技法を提供する。第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。外部入力は、たとえば、ビデオデコードのグラフィックユーザインターフェース(GUI)を介してユーザ(たとえば、ネットワークアドミニストレータ)から受け取られる入力であってもよい。このようにして第1および第2のフラグを設定することは、潜在的にダーティなデータがディスプレイに出力されることを防止する。すなわち、第1および第2のフラグの値は、GDRピクチャからの潜在的にダーティなデータが出力されるかどうか、またはビデオデコードが完全な同期を待ってデータを表示し始めるかどうかを制御する。ダーティなデータが出力されることを制限するための能力を有することによって、ビデオコーディングにおけるコード/デコード(「コーデック」とも呼ばれる)が、現在のコーデックに比べて改善される。実際問題として、改善されたビデオコーディングプロセスは、ビデオが送られ、受け取られ、かつ/または見られるとき、より良好なユーザエクスペリエンスをユーザに与える。

【0006】

第1の態様それ自体に係る方法の第1の実装形式では、第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく、かつ第2のフラグは第1のフラグに等しく設定される。

【0007】

第1の態様それ自体または第1の態様の先行する任意の実装形式に係る方法の第2の実装形式では、外部入力は、ビデオデコードのグラフィックユーザインターフェース(GUI)であり、第1のフラグの値は、外部入力を使用してビデオデコードのユーザによって提供される。

【0008】

第1の態様それ自体または第1の態様の先行する任意の実装形式に係る方法の第3の実装

10

20

30

40

50

形式では、第1のフラグはHandleGdrAsCvsStartFlagと指定される。

【 0 0 0 9 】

第1の態様それ自体または第1の態様の先行する任意の実装形式に係る方法の第4の実装形式では、GDRピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、第1のフラグおよび第2のフラグの値は1に設定される。

【 0 0 1 0 】

第1の態様それ自体または第1の態様の先行する任意の実装形式に係る方法の第5の実装形式では、第1のフラグに対する値が外部入力によって提供されないとき、第1のフラグの値は0に設定される。

10

【 0 0 1 1 】

第2の態様は、復号デバイスに関する。復号デバイスは、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た受信機と、受信機に結合されたメモリであって、命令を記憶する、メモリと、メモリに結合されたプロセッサとを含み、プロセッサは、復号デバイスに、第1のフラグに対する値が外部入力によって提供されるかどうかを決定することと、第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、第1のフラグを、外部入力によって提供される値に等しく、かつ第2のフラグを第1のフラグに等しく設定することと、GDRピクチャを復号することと、GDRピクチャを復号ピクチャバッファ(DPB)の中に記憶することとをさせるために、命令を実行するように構成される。

20

【 0 0 1 2 】

復号デバイスは、イントラランダムアクセスポイント(IRAP)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする技法を提供する。第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。外部入力は、たとえば、ビデオデコーダのグラフィックユーザインターフェース(GUI)を介してユーザ(たとえば、ネットワークアドミニストレータ)から受け取られる入力であってもよい。このようにして第1および第2のフラグを設定することは、潜在的にダーティなデータがディスプレイに出力されることを防止する。すなわち、第1および第2のフラグの値は、GDRピクチャからの潜在的にダーティなデータが出力されるかどうか、またはビデオデコーダが完全な同期を待ってデータを表示し始めるかどうかを制御する。ダーティなデータが出力されることを制限するための能力を有することによって、ビデオコーディングにおけるコーデ/デコーデ(「コーデック」とも呼ばれる)が、現在のコーデックに比べて改善される。実際問題として、改善されたビデオコーディングプロセスは、ビデオが送られ、受け取られ、かつ/または見られるとき、より良好なユーザエクスペリエンスをユーザに与える。

30

【 0 0 1 3 】

第2の態様それ自体に係る復号デバイスの第1の実装形式では、第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく、かつ第2のフラグは第1のフラグに等しく設定される。

40

【 0 0 1 4 】

第2の態様それ自体に係る復号デバイスの第2の実装形式では、外部入力は、ビデオデコーダのグラフィックユーザインターフェース(GUI)であり、第1のフラグの値は、外部入力を使用してビデオデコーダのユーザによって提供される。

【 0 0 1 5 】

第2の態様それ自体に係る復号デバイスの第3の実装形式では、第1のフラグはHandleGdrAsCvsStartFlagと指定される。

50

【 0 0 1 6 】

第2の態様それ自体に係る復号デバイスの第4の実装形式では、GDRピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、第1のフラグおよび第2のフラグの値は1に設定される。

【 0 0 1 7 】

第2の態様それ自体に係る復号デバイスの第5の実装形式では、第1のフラグに対する値が外部入力によって提供されないとき、第1のフラグの値は0に設定される。

【 0 0 1 8 】

第3の態様は、コーディング装置に関する。コーディング装置は、符号化すべきピクチャを受信するか、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た、受信機と、受信機に結合された送信機であって、ビットストリームをデコーダへ送信するか、または復号画像をディスプレイへ送信するように構成された、送信機と、受信機または送信機のうちの少なくとも1つに結合されたメモリであって、命令を記憶するように構成された、メモリと、メモリに結合されたプロセッサであって、本明細書で開示する方法のうちのいずれかを実行するために、メモリの中に記憶された命令を実行するように構成された、プロセッサを含む。

【 0 0 1 9 】

コーディング装置は、イントラランダムアクセスポイント(IRAP)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする技法を提供する。第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。外部入力は、たとえば、ビデオデコーダのグラフィックユーザインターフェース(GUI)を介してユーザ(たとえば、ネットワークアドミニストレータ)から受け取られる入力であってもよい。このようにして第1および第2のフラグを設定することは、潜在的にダーティなデータがディスプレイに出力されることを防止する。すなわち、第1および第2のフラグの値は、GDRピクチャからの潜在的にダーティなデータが出力されるかどうか、またはビデオデコーダが完全な同期を待ってデータを表示し始めるかどうかを制御する。ダーティなデータが出力されることを制限するための能力を有することによって、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」とも呼ばれる)が、現在のコーデックに比べて改善される。実際問題として、改善されたビデオコーディングプロセスは、ビデオが送られ、受け取られ、かつ/または見られるとき、より良好なユーザエクスペリエンスをユーザに与える。

【 0 0 2 0 】

第3の態様それ自体に係るコーディング装置の第1の実装形式では、画像を表示するように構成されたディスプレイをさらに備える。

【 0 0 2 1 】

第4の態様は、システムに関する。システムは、エンコーダと、エンコーダと通信しているデコーダとを含み、エンコーダまたはデコーダは、本明細書で開示する復号デバイス、符号化デバイス、またはコーディング装置を含む。

【 0 0 2 2 】

システムは、イントラランダムアクセスポイント(IRAP)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする技法を提供する。第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。外部入力は、たとえば、ビデオデコーダのグラフィックユーザインターフェース(GUI)を介してユーザ(たとえば、ネットワークアドミニストレータ)から受け取られる入力であってもよい。このようにして第1および第2のフラグを設定することは、潜在的にダーティなデータが

10

20

30

40

50

ディスプレイに出力されることを防止する。すなわち、第1および第2のフラグの値は、GDRピクチャからの潜在的にダーティなデータが出力されるかどうか、またはビデオデコーダが完全な同期を待ってデータを表示し始めるかどうかを制御する。ダーティなデータが出力されることを制限するための能力を有することによって、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」とも呼ばれる)が、現在のコーデックに比べて改善される。実際問題として、改善されたビデオコーディングプロセスは、ビデオが送られ、受け取られ、かつ/または見られるとき、より良好なユーザエクスペリエンスをユーザに与える。

【0023】

第5の態様は、コーディングするための手段に関する。コーディングするための手段は、符号化すべきピクチャを受信するか、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た、受信手段と、受信手段に結合された送信手段であって、ビットストリームを復号手段へ送信するか、または復号画像を表示手段へ送信するように構成された、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つに結合された記憶手段であって、命令を記憶するように構成された、記憶手段と、記憶手段に結合された処理手段であって、本明細書で開示する方法のうちのいずれかを実行するために、記憶手段の中に記憶された命令を実行するように構成された、処理手段とを含む。

【0024】

コーディングするための手段は、イントラランダムアクセスポイント(IRAP)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする技法を提供する。第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。外部入力は、たとえば、ビデオデコーダのグラフィックユーザインターフェース(GUI)を介してユーザ(たとえば、ネットワークアドミニストレータ)から受け取られる入力であってもよい。このようにして第1および第2のフラグを設定することは、潜在的にダーティなデータがディスプレイに出力されることを防止する。すなわち、第1および第2のフラグの値は、GDRピクチャからの潜在的にダーティなデータが出力されるかどうか、またはビデオデコーダが完全な同期を待ってデータを表示し始めるかどうかを制御する。ダーティなデータが出力されることを制限するための能力を有することによって、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」とも呼ばれる)が、現在のコーデックに比べて改善される。実際問題として、改善されたビデオコーディングプロセスは、ビデオが送られ、受け取られ、かつ/または見られるとき、より良好なユーザエクスペリエンスをユーザに与える。

【0025】

本開示のより完全な理解のために、添付図面および発明を実施するための形態とともに理解される、以下の簡潔な説明を次に参照し、同様の参照番号は同様の部分を表す。

【図面の簡単な説明】

【0026】

【図1】GDR技法を利用してもよい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】GDR技法を実施してもよい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】GDR技法を実施してもよいビデオデコーダの一例を示すブロック図である。

【図4】復号順序および提示順序における、リーディングピクチャ(leading picture)に対するIRAPピクチャと、トレーリングピクチャとの間の関係の描写である。

【図5】漸進的復号リフレッシュ技法を示す図である。

【図6】望ましくない動き探索を示す概略図である。

【図7】本開示の一実施形態に係る、漸進的復号リフレッシュ技法を実施するように構成されたビデオビットストリームを示す図である。

【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一実施形態を示す

10

20

30

40

50

図である。

【図 9】ビデオコーディングデバイスの概略図である。

【図 10】コーディングするための手段の一実施形態の概略図である。

【発明を実施するための形態】

【0027】

図1は、本明細書で説明するようなビデオコーディング技法を利用してもよい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に示すように、コーディングシステム10は、後で宛先デバイス14によって復号されるべき符号化ビデオデータを提供するソースデバイス12を含む。詳細には、ソースデバイス12は、コンピュータ可読媒体16を介して宛先デバイス14にビデオデータを提供してもよい。ソースデバイス12および宛先

10

【0028】

宛先デバイス14は、コンピュータ可読媒体16を介して、復号されるべき符号化ビデオデータを受信してもよい。コンピュータ可読媒体16は、符号化ビデオデータをソースデバイス12から宛先デバイス14に移動させ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てもよい。一例では、コンピュータ可読媒体16は、ソースデバイス12が符号化ビデオデータを宛先デバイス14へリアルタイムで直接送信することを可能にするための通信媒体を備えてもよい。符号化ビデオデータは、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てもよく、宛先デバイス14へ送信されてもよい。通信媒体は、無線周波数(RF)スペクトルまたは1つ以上の物理伝送線路などの、任意のワイヤレスまたは有線の通信媒体を備えてもよい。通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはインターネットなどのグローバルネットワークなどの、パケットベースネットワークの一部を形成してもよい。通信媒体は、ルータ、スイッチ、基地局、またはソースデバイス12から宛先デバイス14への通信を容易にするために有用であり得る任意の他の機器を含んでもよい。

20

30

【0029】

いくつかの例では、符号化データは、出力インターフェース22から記憶デバイスに出力されてもよい。同様に、符号化データは、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てもよい。記憶デバイスは、ハードドライブ、Blu-ray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み取り専用メモリ(CD-ROM)、フラッシュメモリ、揮発性メモリもしくは不揮発性メモリ、または符号化ビデオデータを記憶するための任意の他の好適なデジタル記憶媒体などの、分散されるかまたは局所的にアクセスされる様々なデータ記憶媒体のうちのいずれかを含んでもよい。さらなる一例では、記憶デバイスは、ソースデバイス12によって生成された符号化ビデオを記憶してもよいファイルサーバまたは別の中間記憶デバイスに相当してもよい。宛先デバイス14は、ストリーミングまたはダウンロードを介して記憶デバイスからの記憶済みのビデオデータにアクセスしてもよい。ファイルサーバは、符号化ビデオデータを記憶すること、およびその符号化ビデオデータを宛先デバイス14へ送信することが可能な、任意のタイプのサーバであってもよい。例示的なファイルサーバは、(たとえば、ウェブサイト用の)ウェブサーバ、ファイル転送プロトコル(FTP)サーバ、ネットワーク接続ストレージ(NAS)デバイス、またはローカルディスクドライブを含む。宛先デバイス14は、インターネット接続を含む任意の標準データ接続を通じて符号化ビデオデータにアクセスしてもよい。このことは、ファイルサーバ上に記憶された符号化ビデオデータにアクセスするのに適した、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、デジタル加入者回線(DSL)、ケーブルモデムなど)、またはその両方の組み合わせを含んでもよい。記憶デバイスからの符号化ビデ

40

50

データの送信は、ストリーミング送信、ダウンロード送信、またはそれらの組み合わせであってもよい。

【0030】

本開示の技法は、必ずしもワイヤレスの適用例または設定に限定されとは限らない。本技法は、オーバーエアテレビ放送、ケーブルテレビ送信、衛星テレビ送信、動的適応ストリーミングオーバーHTTP(DASH)などのインターネットストリーミングビデオ送信、データ記憶媒体上に符号化されるデジタルビデオ、データ記憶媒体上に記憶されたデジタルビデオの復号、または他の適用例などの、様々なマルチメディア適用例のうちのいずれかをサポートするビデオコーディングに適用されてもよい。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、ビデオブロードキャスト、および/またはビデオ電話などの適用例をサポートするために、一方向または二方向のビデオ送信をサポートするように構成されてもよい。

10

【0031】

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。宛先デバイス14は、入力インターフェース28、ビデオデコーダ30、およびディスプレイデバイス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/または宛先デバイス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てもよい。他の例では、ソースデバイスおよび宛先デバイスは、他の構成要素または構成を含んでもよい。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、宛先デバイス14は、統合型ディスプレイデバイスを含むのではなく、外部のディスプレイデバイスとインターフェースしてもよい。

20

【0032】

図1の図示した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てもよい。本開示の技法は一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、技法はまた、通常、「コーデック」と呼ばれる、ビデオエンコーダ/デコーダによって実行されてもよい。その上、本開示の技法はまた、ビデオプリプロセッサによって実行されてもよい。ビデオエンコーダおよび/またはデコーダは、グラフィックス処理ユニット(GPU)または類似のデバイスであってもよい。

30

【0033】

ソースデバイス12および宛先デバイス14は、宛先デバイス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ソースデバイス12が生成する、その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および宛先デバイス14は、ソースデバイス12および宛先デバイス14の各々がビデオ符号化および復号構成要素を含むように、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、14の間での一方向または二方向のビデオ送信をサポートしてもよい。

【0034】

ソースデバイス12のビデオソース18は、ビデオカメラ、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などの、ビデオキャプチャデバイスを含んでもよい。さらなる代替として、ビデオソース18は、ソースビデオとしてのコンピュータグラフィックススペースのデータ、またはライブビデオと、アーカイブされたビデオと、コンピュータ生成されたビデオとの組み合わせを生成してもよい。

40

【0035】

場合によっては、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および宛先デバイス14は、いわゆるカメラフォンまたはビデオフォンを形成してもよい。しかしながら、上述のように、本開示で説明する技法は、一般にビデオコーディングに適用可

50

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。各場合において、キャプチャ、プリキャプチャ、またはコンピュータ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てもよい。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上に出力されてもよい。

【0036】

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)を含んでもよい。いくつかの例では、ネットワークサーバ(図示せず)が、たとえば、ネットワーク送信を介して、ソースデバイス12から符号化ビデオデータを受信してもよく、宛先デバイス14に符号化ビデオデータを提供してもよい。同様に、ディスクスタンプング設備などの媒体製造設備のコンピューティングデバイスが、ソースデバイス12から符号化ビデオデータを受信してもよく、符号化ビデオデータを含むディスクを製造してもよい。したがって、様々な例では、コンピュータ可読媒体16は、様々な形態の1つ以上のコンピュータ可読媒体を含むものと理解されてもよい。

10

【0037】

宛先デバイス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、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、グループオブピクチャ(GOP:group of pictures)の特性および/または処理を記述するシンタックス要素を含む、ビデオエンコーダ20によって規定されるシンタックス情報を含んでもよく、シンタックス情報はまた、ビデオデコーダ30によって使用される。ディスプレイデバイス32は、復号されたビデオデータをユーザに表示し、陰極線管(CRT)、液晶ディスプレイ(LCD)、プラズマディスプレイ、有機発光ダイオード(OLED)ディスプレイ、または別のタイプのディスプレイデバイスなどの、様々なディスプレイデバイスのうちのいずれかを備えてもよい。

20

【0038】

ビデオエンコーダ20およびビデオデコーダ30は、現在開発中の高効率ビデオコーディング(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テストモデル(HM)に適合してもよい。代替的に、ビデオエンコーダ20およびビデオデコーダ30は、代替として、ムービングピクチャエキスパートグループ(MPEG)-4パート10と呼ばれる、国際電気通信連合電気通信規格セクタ(ITU-T)H.264規格、アドバンストビデオコーディング(AVC)、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てもよい。しかしながら、本開示の技法は、いかなる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は、MPEG-2およびITU-T H.263を含む。図1に示さないが、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または別個のデータストリームの中のオーディオとビデオの両方の符号化を処理するための、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適合してもよい。

30

40

【0039】

ビデオエンコーダ20およびビデオデコーダ30は各々、1つ以上のマイクロプロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、個別論理、ソフトウェア、ハードウェア、ファームウェア、またはそれらの任意の組み合わせなどの、様々な好適なエンコーダ回路構成のうちのいずれかとして実装されてもよい。本技法が部分的にソフトウェアで実装されるとき、デバイスは、好適な非一時的コンピュータ可読媒体の中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つ以上のプロセッサを使用してハードウェアで命

50

令を実行してもよい。ビデオエンコーダ20およびビデオデコーダ30の各々は、1つ以上のエンコーダまたはデコーダの中に含まれてもよく、それらのいずれも、組み合わせられたエンコーダ/デコーダ(コーデック)の一部としてそれぞれのデバイスの中で統合されてもよい。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/またはセルラー電話などのワイヤレス通信デバイスを備えてもよい。

【0040】

図2は、ビデオコーディング技法を実施してもよいビデオエンコーダ20の一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およびインターコーディングを実行してもよい。イントラコーディングは、所与のビデオフレーム内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内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)とは、いくつかの空間ベースのコーディングモードのうちのいずれかを指してもよい。単方向予測(単予測とも呼ばれる)(Pモード)または双予測(bi-prediction)(双予測(bi prediction)とも呼ばれる)(Bモード)などのインターモードとは、いくつかの時間ベースのコーディングモードのうちのいずれかを指してもよい。

【0041】

図2に示すように、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。次に、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)とも呼ばれる)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成のために、ビデオエンコーダ20はまた、逆量子化ユニット58、逆変換ユニット60、および加算器62を含む。ブロック境界をフィルタ処理して再構成ビデオからブロックネスアーティファクトを除去するために、デブロッキングフィルタ(図2に示さず)も含まれてもよい。所望される場合、デブロッキングフィルタは、通常、加算器62の出力をフィルタ処理することになる。デブロッキングフィルタに加えて、(ループ内またはループ後の)追加のフィルタも使用されてもよい。そのようなフィルタは、簡潔のために図示されないが、所望される場合、(ループ内フィルタとして)加算器50の出力をフィルタ処理してもよい。

【0042】

符号化プロセス中、ビデオエンコーダ20は、コーディングされるべき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てもよい。動き推定ユニット42および動き補償ユニット44は、1つ以上の参照フレームの中の1つ以上のブロックに対する受信ビデオブロックのインター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、代替的に、コーディングされるべきブロックと同じフレームまたはスライスの中の1つ以上の隣接ブロックに対する受信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、空間予測を行ってよい。ビデオエンコーダ20は、たとえば、ビデオデータのブロックごとに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てもよい。

【0043】

その上、区分ユニット48は、前のコーディングパスにおける、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に区分してもよい。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:largest coding unit)に区分してもよく、レートひずみ分析(たとえば、レートひずみ最適化)に基づいてLCUの各々をサブコーディングユニット(サブCU)に区分してもよい。モード選択ユニット40は、サブCUへのLCUの区分を示す4分木データ構造をさらに生成してもよい。4分

10

20

30

40

50

木のリーフノードCUは、1つ以上の予測ユニット(PU:prediction unit)および1つ以上の変換ユニット(TU:transform unit)を含んでもよい。

【0044】

本開示は、HEVCのコンテキストにおけるCU、PU、もしくはTU、または他の規格のコンテキスト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それらのサブブロック)のうち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に関連する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、最大値が64×64ピクセル以上のツリーブロックのサイズまでにわたってもよい。各CUは、1つ以上のPUおよび1つ以上のTUを含んでもよい。CUに関連するシンタックスデータは、たとえば、1つ以上のPUへのCUの区分を記述してもよい。CUが、スキップモードまたはダイレクトモードで符号化されるのか、イントラ予測モードで符号化されるのか、それともインター予測(inter-prediction)(インター予測(inter prediction)とも呼ばれる)モードで符号化されるのかの間で、区分モードは異なってもよい。PUは、形状が非正方形となるように区分されてもよい。CUに関連するシンタックスデータはまた、たとえば、4分木による1つ以上のTUへのCUの区分を記述してもよい。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。

10

【0045】

モード選択ユニット40は、たとえば、誤差結果に基づいて、コーディングモード、すなわち、イントラモードまたはインターモードのうちの1つを選択してもよく、残差ブロックデータを生成するために加算器50に、また参照フレームとして使用するための符号化ブロックを再構成するために加算器62に、得られたイントラコーディングまたはインターコーディングされたブロックを提供する。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、区分情報などのシンタックス要素、および他のそのようなシンタックス情報を、エントロピーコーディングユニット56に提供する。

20

【0046】

動き推定ユニット42および動き補償ユニット44は、高度に統合されてもよいが、概念的な目的のために別個に図示される。動き推定ユニット42によって実行される動き推定とは、ビデオブロックに対する動きを推定する、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、現在フレーム(または、他のコーディングされたユニット)内でコーディング中の現在ブロックに対して、参照フレーム(または、他のコーディングされたユニット)内の予測ブロックと比べて、現在ビデオフレーム内または現在ピクチャ内でのビデオブロックのPUの変位を示してもよい。予測ブロックとは、ピクセル差分の観点から、コーディングされるべきブロックに密に整合するものと見出されるブロックであり、ピクセル差分は、絶対差分和(SAD:sum of absolute difference)、2乗差分和(SSD:sum of square difference)、または他の差分メトリックによって決定されてもよい。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の中に記憶された参照ピクチャのサブ整数ピクセル位置に対する値を計算してもよい。たとえば、ビデオエンコーダ20は、参照ピクチャの1/4ピクセル位置、1/8ピクセル位置、または他の分数ピクセル位置の値を補間してもよい。したがって、動き推定ユニット42は、フル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てもよく、分数ピクセル精度を有する動きベクトルを出力してもよい。

30

40

【0047】

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のための動きベクトルを計算する。参照ピクチャは、参照フレームメモリ64の中に記憶された1つ以上の参照ピクチャをその各々が識別する、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い。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へ送る。

【0048】

50

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によって決定された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ってもよい。再び、動き推定ユニット42および動き補償ユニット44は、いくつかの例では機能的に統合されてもよい。現在ビデオブロックの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44は、参照ピクチャリストのうちの1つの中で、動きベクトルが指し示す先の予測ブロックの位置を特定してもよい。加算器50は、コーディング中の現在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減算することによって残差ビデオブロックを形成して、以下で説明するようにピクセル差分値を形成する。一般に、動き推定ユニット42は、ルーマ成分に対して動き推定を実行し、動き補償ユニット44は、ルーマ成分に基づいて計算された動きベクトルを、クロマ成分とルーマ成分の両方のために使用する。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によって使用するための、ビデオブロックおよびビデオスライスに関連するシンタックス要素を生成してもよい。

10

【0049】

イントラ予測ユニット46は、上記で説明したように、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、現在ブロックをイントラ予測してもよい。詳細には、イントラ予測ユニット46は、現在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てもよい。いくつかの例では、イントラ予測ユニット46は、たとえば、別個の符号化パス中に、様々なイントラ予測モードを使用して現在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、使用すべき適切なイントラ予測モードを、テストされたモードから選択してもよい。

20

【0050】

たとえば、イントラ予測ユニット46は、テストされた様々なイントラ予測モードに対して、レートひずみ分析を使用してレートひずみ値を計算してもよく、テストされたモードの中から、レートひずみ特性が最良のイントラ予測モードを選択してもよい。レートひずみ分析は、概して、符号化ブロックと、符号化ブロックを生成するために符号化された、符号化されていない元のブロックとの間のひずみ(すなわち、誤差)の量、ならびに符号化ブロックを生成するために使用されるビットレート(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、様々な符号化ブロックに対して、ひずみおよびレートから比率を計算して、ブロックに対してどのイントラ予測モードが最良のレートひずみ値を呈示するのかを決定してもよい。

30

【0051】

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:depth modeling mode)を使用して深度マ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てもよい。モード選択ユニット40は、利用可能なDMMモードが、たとえば、レートひずみ最適化(RDO:rate-distortion optimization)を使用して、イントラ予測モードおよび他のDMMモードよりも良好なコーディング結果を生成するかどうかを決定してもよい。深度マップに対応するテクスチャ画像に対するデータが、参照フレームメモリ64の中に記憶されてもよい。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てもよい。

40

【0052】

ブロック用の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、ブロック用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コーディングユニット56に提供してもよい。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てもよい。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済みのイントラ予測モードインデックステーブル(コードワードマッピングテーブルとも呼ばれる)、様々なブロックに対する符号化コンテキストの定義、ならびにコンテキストの各々に対して使用すべき最確のイントラ予測モード、イントラ予測モ

50

ードインデックステーブル、および修正済みのイントラ予測モードインデックステーブルの表示を含んでもよい構成データを、送信されるビットストリームの中にも含めてもよい。

【0053】

ビデオエンコーダ20は、コーディング中の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減算することによって、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つ以上の構成要素を表す。

【0054】

変換処理ユニット52は、離散コサイン変換(DCT)または概念的に類似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成する。変換処理ユニット52は、DCTと概念的に類似の他の変換を実行してもよい。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。

【0055】

変換処理ユニット52は、残差ブロックに変換を適用して、残差変換係数のブロックを生成する。変換は、残差情報をピクセル値領域から周波数領域などの変換領域に変換してもよい。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へ送ってもよい。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、係数の一部または全部に関連するビット深度を低減してもよい。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てもよい。いくつかの例では、量子化ユニット54は、次いで、量子化変換係数を含む行列の走査を実行してもよい。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。

【0056】

量子化に続いて、エントロピーコーディングユニット56は、量子化変換係数をエントロピーコーディングする。たとえば、エントロピーコーディングユニット56は、コンテキスト適応型可変長コーディング(CAVLC)、コンテキスト適応型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型バイナリ算術コーディング(SBAC)、確率区間区分エントロピー(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てもよい。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には、コンテキストは隣接ブロックに基づいてもよい。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)へ送信されてもよく、または後で送信するかもしくは取り出すためにアーカイブされてもよい。

【0057】

逆量子化ユニット58および逆変換ユニット60は、それぞれ、逆量子化および逆変換を適用して、たとえば、後で参照ブロックとして使用するために、ピクセル領域における残差ブロックを再構成する。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加算することによって、参照ブロックを計算してもよい。動き補償ユニット44はまた、再構成された残差ブロックに1つ以上の補間フィルタを適用して、動き推定における使用のためのサブ整数ピクセル値を計算してもよい。加算器62は、動き補償ユニット44によって生成された、動き補償された予測ブロックに、再構成された残差ブロックを加算して、参照フレームメモリ64の中に記憶するための再構成ビデオブロックを生成する。再構成ビデオブロックは、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の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によって使用されてもよい。

【0058】

図3は、ビデオコーディング技法を実施してもよいビデオデコーダ30の一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、エントロピー復号ユニット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)に関して説明した符号化パスとは概して相反の、復号パスを実行してもよい。動き補償ユニット72は、エントロピー復号ユニット70から受信され

10

20

30

40

50

た動きベクトルに基づいて予測データを生成してもよいが、イントラ予測ユニット74は、エントロピー復号ユニット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てもよい。

【0059】

復号プロセス中、ビデオデコーダ30は、符号化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のエントロピー復号ユニット70は、ビットストリームをエントロピー復号して、量子化係数、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。エントロピー復号ユニット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てもよい。

10

【0060】

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、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、および現在フレームまたは現在ピクチャの、前に復号されたブロックからのデータに基づいて、現在ビデオスライスのビデオブロックのための予測データを生成してもよい。ビデオフレームがインターコーディングされた(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とき、動き補償ユニット72は、エントロピー復号ユニット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を生成する。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つから生成されてもよい。ビデオデコーダ30は、参照フレームメモリ82の中に記憶された参照ピクチャに基づいてデフォルトの構成技法を使用して、参照フレームリスト、すなわち、リスト0およびリスト1を構成してもよい。

20

【0061】

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構文解析することによって、現在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し、予測情報を使用して、復号中の現在ビデオブロックのための予測ブロックを生成する。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うち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た予測モード(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)、スライス用の参照ピクチャリストのうちの1つ以上に対する構成情報、スライスのインター符号化ビデオブロックごとの動きベクトル、スライスのインターコーディングされたビデオブロックごとのインター予測ステータス、および現在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。

30

【0062】

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てもよい。動き補償ユニット72は、ビデオブロックの符号化中にビデオエンコーダ20によって使用された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値を計算してもよい。この場合、動き補償ユニット72は、ビデオエンコーダ20によって使用された補間フィルタを、受信されたシンタックス要素から決定し得、その補間フィルタを使用して予測ブロックを生成してもよい。

40

【0063】

深度マップに対応するテクスチャ画像に対するデータが、参照フレームメモリ82の中に記憶されてもよい。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てもよい。

【0064】

一実施形態では、ビデオデコーダ30はユーザインターフェース(UI)84を含む。ユーザインターフェース84は、ビデオデコーダ30のユーザ(たとえば、ネットワークアドミニストレータ)から入力を受け取るように構成される。ユーザインターフェース84を通じて、

50

ユーザは、ビデオデコーダ30上の設定を管理または変更することができる。たとえば、ユーザは、ユーザの選好に従ってビデオデコーダ30の構成および/または動作を制御するために、パラメータ(たとえば、フラグ)に対する値を入力または別の方法で提供することができる。ユーザインターフェース84は、たとえば、ユーザがグラフィカルアイコン、ドロップダウンメニュー、チェックボックスなどを通じてビデオデコーダ30と対話することを可能にする、グラフィカルユーザインターフェース(GUI)であってもよい。場合によっては、ユーザインターフェース84は、キーボード、マウス、または他の周辺デバイスを介してユーザから情報を受け取ってもよい。一実施形態では、ユーザは、ビデオデコーダ30から遠く離れて配置されたスマートフォン、タブレットデバイス、パーソナルコンピュータなどを介して、ユーザインターフェース84にアクセスすることができる。本明細書で使用するユーザインターフェース84は、外部入力または外部手段と呼ばれてもよい。

10

【0065】

上記のことを念頭に置いて、ビデオ圧縮技法は、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)予測を実行して、ビデオシーケンスに固有の冗長性を低減または除去する。ブロックベースのビデオコーディングの場合、ビデオスライス(すなわち、ビデオピクチャ、またはビデオピクチャの一部)は、ツリーブロック、コーディングツリーブロック(CTB: coding tree block)、コーディングツリーユニット(CTU: coding tree unit)、コーディングユニット(CU: coding unit)、および/またはコーディングノードとも呼ばれてもよい、ビデオブロックに区分されてもよい。ピクチャのイントラコーディングされた(I)スライスの中のビデオブロックは、同じピクチャの中の隣接ブロックの中の参照サンプルに対する空間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隣接ブロックの中の参照サンプルに対する空間予測、または他の参照ピクチャの中の参照サンプルに対する時間予測を使用してよい。ピクチャはフレームと呼ばれてもよく、参照ピクチャは参照フレームと呼ばれてもよい。

20

【0066】

空間予測または時間予測は、コーディングされるべきブロックのための予測ブロックをもたらす。残差データは、コーディングされるべき元のブロックと予測ブロックとの間の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、およびコーディングされたブロックと予測ブロックとの間の差分を示す残差データに従って符号化される。イントラコーディングされたブロックは、イントラコーディングモードおよび残差データに従って符号化される。さらなる圧縮のために、残差データは、ピクセル領域から変換領域に変換されてもよく、結果として残差変換係数になり、残差変換係数は、次いで、量子化されてもよい。当初は二次元アレイをなして配置された量子化変換係数は、変換係数の一次元ベクトルを生成するために走査されてもよく、なお一層の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。

30

【0067】

画像およびビデオ圧縮は急成長を経ており、様々なコーディング規格に至る。そのようなビデオコーディング規格は、ITU-T H.261、ISO/IEC MPEG-1パート2、ITU-T H.262またはISO/IEC MPEG-2パート2、ITU-T H.263、ISO/IEC MPEG-4パート2、ITU-T H.264またはISO/IEC MPEG-4パート10とも呼ばれるアドバンスドビデオコーディング(AVC)、およびITU-T H.265またはMPEG-Hパート2とも呼ばれる高効率ビデオコーディング(HEVC)を含む。AVCは、スケーラブルビデオコーディング(SVC)、マルチビュービデオコーディング(MVC)およびマルチビュービデオコーディングプラス深度(MVC+D)、ならびに3D AVC(3D-AVC)などの拡張を含む。HEVCは、スケーラブルHEVC(SHVC)、マルチビューHEVC(MV-HEVC)、および3D HEVC(3D-HEVC)などの拡張を含む。

40

【0068】

ITU-TとISO/IECとの共同ビデオエキスパートチーム(JVET)によって開発中の多用途ビデオコーディング(VVC)と称する新たなビデオコーディング規格もある。VVC規格はいく

50

つかのワーキングドラフトを有するが、特にVVCの1つのワーキングドラフト(WD)、すなわち、B.Bross、J.Chen、およびS.Liu、「Versatile Video Coding (Draft 4)」、JVE T-M1001-v5、第13回JVET会合、2019年1月(VVCドラフト4)が、本明細書で参照される。

【0069】

本明細書で開示する技法の説明は、開発中のビデオコーディング規格、すなわち、ITU-TとISO/IECとの共同ビデオエキスパートチーム(JVET)による多用途ビデオコーディング(VVC)に基づく。しかしながら、本技法は、他のビデオコーデック仕様にも適用される。

【0070】

図4は、復号順序408および提示順序410における、リーディングピクチャ404に対するIRAPピクチャ402と、トレーリングピクチャ406との間の関係の描写400である。一実施形態では、IRAPピクチャ402は、クリーンランダムアクセス(CRA:clean random access)ピクチャ、またはランダムアクセス復号可能(RADL:random access decodable)ピクチャを伴う瞬時デコーダリフレッシュ(IDR:instantaneous decoder refresh)ピクチャと呼ばれる。HEVCでは、IDRピクチャ、CRAピクチャ、およびブローケンリンクアクセス(BLA:Broken Link Access)ピクチャは、すべてIRAPピクチャ402と見なされる。VVCの場合、2018年10月における第12回JVET会合の間に、IRAPピクチャとしてIDRピクチャとCRAピクチャの両方を有することが合意された。

10

【0071】

図4に示すように、リーディングピクチャ404(たとえば、ピクチャ2および3)は、復号順序408でIRAPピクチャ402に後続するが、提示順序410でIRAPピクチャ402に先行する。トレーリングピクチャ406は、復号順序408と提示順序410の両方でIRAPピクチャ402に後続する。2つのリーディングピクチャ404および1つのトレーリングピクチャ406が図4に示されるが、実際の適用例では、より多数またはより少数のリーディングピクチャ404および/またはトレーリングピクチャ406が復号順序408および提示順序410で存在してもよいことを、当業者は諒解されよう。

20

【0072】

図4の中のリーディングピクチャ404は、2つのタイプ、すなわち、ランダムアクセススキップドリーディング(RASL:random access skipped leading)およびRADLに分割されている。IRAPピクチャ402(たとえば、ピクチャ1)で復号が開始するとき、RADLピクチャ(たとえば、ピクチャ3)は正しく復号され得るが、RASLピクチャ(たとえば、ピクチャ2)は正しく復号され得ない。したがって、RASLピクチャは廃棄される。RADLピクチャとRASLピクチャとの間の相違に照らして、IRAPピクチャに関連するリーディングピクチャのタイプは、効率的かつ適切なコーディングのためにRADLまたはRASLのいずれかとして識別されるべきである。HEVCでは、RASLピクチャおよびRADLピクチャが存在するとき、同じIRAPピクチャに関連するRASLピクチャおよびRADLピクチャに対して、RASLピクチャが提示順序410でRADLピクチャに先行しなければならないことが制約される。

30

【0073】

IRAPピクチャ402は、以下の2つの重要な機能/利点をもたらす。第一に、IRAPピクチャ402の存在は、そのピクチャから復号プロセスが開始できることを示す。この機能により、IRAPピクチャ402がその位置に存在する限り、必ずしもビットストリームの冒頭とは限らずビットストリームの中のその位置において復号プロセスが開始する、ランダムアクセス特徴が可能になる。第二に、IRAPピクチャ402の存在は、IRAPピクチャ402において開始しRASLピクチャを除外するコーディングされたピクチャが、前のピクチャへのいかなる参照も伴わずにコーディングされるように、復号プロセスをリフレッシュする。それ故に、ビットストリームの中に存在するIRAPピクチャ402を有することは、IRAPピクチャ402および復号順序408においてIRAPピクチャ402に後続するピクチャにIRAPピクチャ402が伝搬する前に、コーディングされたピクチャの復号中に起こる可能性がある、いかなるエラーも止めることになる。

40

【0074】

50

IRAPピクチャ402は、重要な機能を提供するが、圧縮効率への不利益が付いてくる。IRAPピクチャ402の存在は、ビットレートの急上昇を引き起こす。圧縮効率へのこの不利益は、2つの理由に起因する。第一に、IRAPピクチャ402がイントラ予測されるピクチャであるとき、インター予測されるピクチャである他のピクチャ(たとえば、リーディングピクチャ404、トレーリングピクチャ406)と比較すると、ピクチャ自体が、描写するために比較的多くのビットを必要とすることになる。第二に、IRAPピクチャ402の存在が時間予測を破壊するので(なぜなら、デコーダが復号プロセスをリフレッシュすることになり、このことに対する復号プロセスのアクションのうちの1つが、復号ピクチャバッファ(DPB)の中の、前の参照ピクチャを除去することであるからである)、IRAPピクチャ402は、それらのインター予測コーディング用のより少ない参照ピクチャしか有しないので、復号順序408でIRAPピクチャ402に後続するピクチャのコーディングをさほど効率的にさせない(すなわち、描写するためにより多くのビットを必要とする)。

10

【0075】

IRAPピクチャ402と見なされるピクチャタイプのうち、HEVCにおけるIDRピクチャは、他のピクチャタイプと比較したとき、異なるシグナリングおよび導出を有する。差異のうちのいくつかは次の通りである。

【0076】

IDRピクチャのピクチャ順序カウンタ(POC:picture order count)値のシグナリングおよび導出のために、POCの最上位ビット(MSB)部分は前のキーピクチャ(key picture)から導出されず、単に0に等しくなるように設定される。

20

【0077】

参照ピクチャ管理のために必要とされるシグナリング情報の場合、IDRピクチャのスライスヘッダは、参照ピクチャ管理を支援するためにシグナリングされることを必要とされる情報を含まない。他のピクチャタイプ(すなわち、CRA、トレーリング、時間サブレイヤアクセス(TSA:temporal sub-layer access)など)の場合、以下で説明する参照ピクチャセット(RPS:reference picture set)などの情報、または他の形式の類似の情報(たとえば、参照ピクチャリスト)は、参照ピクチャマーキングプロセス(すなわち、復号ピクチャバッファ(DPB)の中の参照ピクチャのステータス、すなわち、参照のために使用済みおよび参照のために未使用のいずれかを、決定するためのプロセス)のために必要とされる。しかしながら、IDRピクチャの場合、復号プロセスが、参照のために未使用としてDPBの中のすべての参照ピクチャに単にマークしなければならないことを、IDRの存在が示すので、そのような情報はシグナリングされる必要がない。

30

【0078】

HEVCおよびVVCでは、IRAPピクチャ402およびリーディングピクチャ404は各々、単一のネットワークアブストラクションレイヤ(NAL)ユニット内に含まれてもよい。NALユニットのセットは、アクセスユニットと呼ばれてもよい。IRAPピクチャ402およびリーディングピクチャ404は、システムレベルアプリケーションによって容易に識別され得るように、異なるNALユニットタイプが与えられる。たとえば、ビデオスプライサは、詳細には、トレーリングピクチャ406からRASLピクチャおよびRADLピクチャを決定することを含めて、非IRAPピクチャからIRAPピクチャ402を識別するために、またリーディングピクチャ404を識別するために、コーディングされたビットストリームの中のシンタックス要素のあまりにも多くの詳細を理解する必要なく、コーディングされたピクチャタイプを理解する必要がある。トレーリングピクチャ406は、IRAPピクチャ402に関連するピクチャであり、提示順序410でIRAPピクチャ402に後続する。ピクチャは、復号順序408で特定のIRAPピクチャ402に後続してもよく、復号順序408で任意の他のIRAPピクチャ402に先行してもよい。このことのために、IRAPピクチャ402およびリーディングピクチャ404にそれら自体のNALユニットタイプを与えることは、そのようなアプリケーションの助けとなる。

40

【0079】

HEVCの場合、IRAPピクチャのためのNALユニットタイプは以下を含む。

50

リーディングピクチャを伴うBLA(BLA_W_LP): 復号順序で1つ以上のリーディングピクチャが後続してもよいブローkunリンクアクセス(BLA)ピクチャのNALユニット。

RADLを伴うBLA(BLA_W_RADL): 復号順序で1つ以上のRADLピクチャが後続してもよいがRASLピクチャが後続し得ないBLAピクチャのNALユニット。

リーディングピクチャを伴わないBLA(BLA_N_LP): 復号順序でリーディングピクチャが後続しないBLAピクチャのNALユニット。

RADLを伴うIDR(IDR_W_RADL): 復号順序で1つ以上のRADLピクチャが後続してもよいがRASLピクチャが後続し得ないIDRピクチャのNALユニット。

リーディングピクチャを伴わないIDR(IDR_N_LP): 復号順序でリーディングピクチャが後続しないIDRピクチャのNALユニット。

CRA: リーディングピクチャが後続してもよいクリーンランダムアクセス(CRA)ピクチャのNALユニット(すなわち、RASLピクチャもしくはRADLピクチャのいずれか、またはその両方)。

RADL: RADLピクチャのNALユニット。

RASL: RASLピクチャのNALユニット。

【0080】

VVCの場合、IRAPピクチャ402およびリーディングピクチャ404のためのNALユニットタイプは次の通りである。

RADLを伴うIDR(IDR_W_RADL): 復号順序で1つ以上のRADLピクチャが後続してもよいがRASLピクチャが後続し得ないIDRピクチャのNALユニット。

リーディングピクチャを伴わないIDR(IDR_N_LP): 復号順序でリーディングピクチャが後続しないIDRピクチャのNALユニット。

CRA: リーディングピクチャが後続してもよいクリーンランダムアクセス(CRA)ピクチャのNALユニット(すなわち、RASLピクチャもしくはRADLピクチャのいずれか、またはその両方)。

RADL: RADLピクチャのNALユニット。

RASL: RASLピクチャのNALユニット。

【0081】

順次イントラリフレッシュ/漸進的復号リフレッシュが以下で説明される。

【0082】

低遅延適用例に対して、非IRAPピクチャ(すなわち、Pピクチャ/Bピクチャ)と比較して、比較的大きく、それ故に、より大きいレイテンシ/遅延を引き起こす、そのビットレート要件に起因して、IRAPピクチャ(たとえば、IRAPピクチャ402)としてピクチャをコーディングすることを回避することが望ましい。しかしながら、IRAPの使用を完全に回避することは、すべての低遅延適用例において可能であるとは限らない場合がある。たとえば、マルチパーティ遠隔会議などの会話型の適用例の場合、新たなユーザが遠隔会議に参加できる定期的なポイントを提供することが必要である。

【0083】

新たなユーザがマルチパーティ遠隔会議適用例に参加することを可能にする、ビットストリームへのアクセスを提供するために、1つの可能な方策は、ビットレートにおけるピークを有することを回避するために、IRAPピクチャを使用するのではなく順次イントラリフレッシュ技法(PIR:progressive intra refresh)を使用することである。PIRは、漸進的復号リフレッシュ(GDR)とも呼ばれてもよい。PIRおよびGDRという用語は、本開示において互換的に使用されてもよい。

【0084】

図5は、漸進的復号リフレッシュ(GDR)技法500を示す。図示のように、GDR技法500は、ビットストリームのコーディングされたビデオシーケンス508内で、GDRピクチャ502、1つ以上のトレーリングピクチャ504、およびリカバリポイントピクチャ506を使用して示される。一実施形態では、GDRピクチャ502、トレーリングピクチャ504、およびリカバリポイントピクチャ506は、CVS508の中のGDR期間を規定してもよい。CVS508は

10

20

30

40

50

、GDRピクチャ502で開始する一連のピクチャ(または、それらの部分)であり、次のGDRピクチャまでの(ただし、それを含まない)、またはビットストリームの終端までの、すべてのピクチャ(または、それらの部分)を含む。GDR期間は、GDRピクチャ502で開始する一連のピクチャであり、リカバリポイントピクチャ506までの(それを含む)すべてのピクチャを含む。

【0085】

図5に示すように、GDR技法500または原理は、GDRピクチャ502で開始しリカバリポイントピクチャ506で終了する一連のピクチャにわたって機能する。GDRピクチャ502は、すべてがイントラ予測を使用してコーディングされているブロック(すなわち、イントラ予測ブロック)を含むリフレッシュ済みの/クリーンな領域510、およびすべてがインター予測を使用してコーディングされているブロック(すなわち、インター予測ブロック)を含むリフレッシュされていない/ダーティな領域512を含む。

10

【0086】

GDRピクチャ502に直接隣接するトレーリングピクチャ504は、イントラ予測を使用してコーディングされる第1の部分510Aおよびインター予測を使用してコーディングされる第2の部分510Bを有する、リフレッシュ済みの/クリーンな領域510を含む。第2の部分510Bは、たとえば、CVS508のGDR期間内の先行するピクチャの、リフレッシュ済みの/クリーンな領域510を参照することによってコーディングされる。図示のように、トレーリングピクチャ504のリフレッシュ済みの/クリーンな領域510は、一貫した方向で(たとえば、左から右に)コーディングプロセスが移動または進行するにつれて拡大し、それに対応して、そのことはリフレッシュされていない/ダーティな領域512を縮小する。最終的に、リフレッシュ済みの/クリーンな領域510しか含まないリカバリポイントピクチャ506が、コーディングプロセスから取得される。特に、さらに以下で説明するように、インター予測ブロックとしてコーディングされる、リフレッシュ済みの/クリーンな領域510の第2の部分510Bは、参照ピクチャの中のリフレッシュ済みの領域/クリーンな領域510を参照するだけでよい。

20

【0087】

HEVCでは、図5のGDR技法500が、リカバリポイント補足エンハンスメント情報(SEI: Supplemental Enhancement Information)メッセージおよび領域リフレッシュ情報SEIメッセージを使用して非規範的にサポートされた。これらの2つのSEIメッセージは、GDRがどのように実行されるのかを規定しない。むしろ、その2つのSEIメッセージは、(すなわち、リカバリポイントSEIメッセージによって提供される)GDR期間の中の最初のピクチャおよび最後のピクチャならびに(すなわち、領域リフレッシュ情報SEIメッセージによって提供される)リフレッシュされている領域を示すためのメカニズムを単に提供する。

30

【0088】

実際には、GDR技法500は、2つの技法を一緒に使用することによって実行される。これらの2つの技法とは、制約イントラ予測(CIP: constraint intra prediction)、および動きベクトルに対するエンコーダ制約である。CIPは、リフレッシュされていない領域(たとえば、リフレッシュされていない/ダーティな領域512)からのサンプルを使用しない領域が、参照のために使用されることを可能にするので、CIPは、特にイントラ予測ブロック(たとえば、リフレッシュ済みの/クリーンな領域510の第1の部分510A)としてのみコーディングされる領域をコーディングするための、GDR目的のために使用され得る。しかしながら、イントラブロックへの制約が、リフレッシュ済みの領域の中のイントラブロックに対してだけでなくピクチャの中のすべてのイントラブロックにも適用されなければならないので、CIPの使用は深刻なコーディング性能劣化を引き起こす。動きベクトルに対するエンコーダ制約は、リフレッシュ済みの領域の外側に位置する参照ピクチャの中の任意のサンプルをエンコーダが使用することを制限する。そのような制約は、最適でない動き探索を引き起こす。

40

【0089】

図6は、GDRをサポートするためにエンコーダ制約を使用するときの、望ましくない動

50

き探索600を示す概略図である。図示のように、動き探索600は、現在ピクチャ602および参照ピクチャ604を示す。現在ピクチャ602および参照ピクチャ604は各々、イントラ予測を用いてコーディングされたリフレッシュ済みの領域606、インター予測を用いてコーディングされたリフレッシュ済みの領域608、およびリフレッシュされていない領域608を含む。リフレッシュ済みの領域604、リフレッシュ済みの領域606、およびリフレッシュされていない領域608は、図5の中のリフレッシュ済みの/クリーンな領域510の第1の部分510A、リフレッシュ済みの/クリーンな領域510の第2の部分510B、およびリフレッシュされていない/ダーティな領域512と類似である。

【0090】

動き探索プロセス中、エンコーダは、リフレッシュ済みの領域606の外側に位置している参照ブロック612のサンプルのうちのいくつかをもたらし、いかなる動きベクトル610を選択することも制約または防止される。このことは、現在ピクチャ602の中の現在ブロック614を予測するときに参照ブロック612が最良のレートひずみコスト基準を与えるとときにさえ行われる。したがって、図6は、GDRをサポートするためのエンコーダ制約を使用するときの、動き探索600における非最適性に対する理由を示す。

【0091】

JVET寄稿JVET-K0212およびJVET-L0160は、CIPおよびエンコーダ制約手法の使用に基づくGDRの実装形態を記載している。実装形態は、次のように要約され得る。すなわち、列ごとにコーディングユニットに対してイントラ予測モードが強制され、イントラCUの再構成を確実なものにするために、制約付きイントラ予測が有効化され、動きベクトルは、フィルタが原因で広がる誤差(たとえば、6ピクセル)を回避するための追加のマージンを考慮に入れるとともに、イントラ列を再ループさせるときに過去の参照ピクチャを除去しながら、リフレッシュ済みのエリア内を指し示すように制約される。

【0092】

JVET寄稿JVET-M0529は、ピクチャがGDR期間の中で最初のピクチャおよび最後のピクチャであることを規範的に示すための方法を提案した。提案された着想は次のように機能する。

【0093】

NALユニットタイプリカバリポイント表示を有する新たなNALユニットを、非ビデオコーディングレイヤ(VCL)NALユニットとして規定する。NALユニットのペイロードは、GDR期間の中の最後のピクチャのPOC値を導出するために使用され得る情報を指定するためのシンタックス要素を含む。タイプリカバリポイント表示を有する非VCL NALユニットを含むアクセスユニットは、リカバリポイント開始(RBP:recovery point begin)アクセスユニット(AU:access unit)と呼ばれ、RBPアクセスユニットの中のピクチャは、RBPピクチャと呼ばれる。復号プロセスは、RBP AUから開始することができる。復号がRBP AUから開始するとき、最後のピクチャを除いてGDR期間の中のすべてのピクチャは出力されない。

【0094】

既存のGDR設計を伴う問題のうちのいくつかが説明される。

【0095】

GDRをサポートするための既存の設計/手法は、少なくとも以下の問題を有する。

【0096】

JVET-M0529におけるGDRを規範的に規定するための方法は、以下の問題を有する。提案された方法は、GDRがどのように実行されるのかを説明していない。代わりに、提案された方法は、GDR期間の中の最初のピクチャおよび最後のピクチャを示すためのいくつかのシグナリングを提供するにすぎない。GDR期間の中の最初のピクチャおよび最後のピクチャを示すために、新たな非VCL NALユニットが必要とされる。リカバリポイント表示(RPI:recovery point indication)NALユニットの中に含まれる情報が、GDR期間の中の最初のピクチャのタイルグループヘッダの中に単に含められ得るので、このことは冗長性である。また、提案された方法は、GDR期間の中のピクチャの中のどの領域がリフレッシュ

10

20

30

40

50

リフレッシュ済みの領域およびリフレッシュされていない領域であるのかを表すことができない。

【0097】

JVET-K0212およびJVET-L0160に記載されるGDR手法は、以下の問題を有する。第一に、CIPの使用。リフレッシュされていない領域からの任意のサンプルが空間的な参照のために使用されることを防止するために、リフレッシュ済みの領域を、いくつかの制約を伴うイントラ予測を用いてコーディングすることが必要である。CIPが使用されるとき、コーディングはピクチャベースであり、そのことは、ピクチャの中のすべてのイントラブロックもCIPイントラブロックとしてコーディングされなければならないことを意味する。それ故に、このことは性能劣化を引き起こす。さらに、動きベクトルに関連する参照ブロックのサンプルが、完全に参照ピクチャの中のリフレッシュ済みの領域内にあるとは限らないとき、動き探索を限定するためのエンコーダ制約の使用は、最良の動きベクトルをエンコーダが選ぶことを妨げる。また、イントラ予測のみを用いてコーディングされるリフレッシュ済みの領域はCTUサイズでない。代わりに、リフレッシュ済みの領域は、最小CUサイズまで下に、CTUサイズよりも小さくすることができる。このことは、ブロックレベルでの表示を必要としてもよいので、実装を不必要に複雑にさせる。

10

【0098】

ビデオコーディングにおける漸進的復号リフレッシュ(GDR)をサポートするための技法が、本明細書で開示される。開示する技法は、イントラランダムアクセスポイント(IRAP)ピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とする。第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、第1のフラグは外部入力によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。外部入力は、たとえば、ビデオデコーダ30のグラフィックユーザインターフェース(GUI)を介してユーザ(たとえば、ネットワークアドミニストレータ)から受け取られる入力であってもよい。このようにして第1および第2のフラグを設定することは、潜在的にダーティなデータがディスプレイに出力されることを防止する。すなわち、第1および第2のフラグの値は、GDRピクチャからの潜在的にダーティなデータが出力されるかどうか、またはビデオデコーダが完全な同期を待ってデータを表示し始めるかどうかを制御する。ダーティなデータが出力されることを制限するための能力を有することによって、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」とも呼ばれる)が、現在のコーデックに比べて改善される。実際問題として、改善されたビデオコーディングプロセスは、ビデオが送られ、受け取られ、かつ/または見られるとき、より良好なユーザエクスペリエンスをユーザに与える。

20

30

【0099】

上記で説明した問題のうちの1つ以上を解決するために、本開示は以下の態様を開示する。態様の各々は個別に適用され得、それらのうちのいくつかは組み合わせで適用され得る。

【0100】

- 1) タイプGDR_NUTを有するVCL NALユニットが規定される。

40

【0101】

- a. NALユニットタイプGDR_NUTを有するピクチャは、GDRピクチャ、すなわち、GDR期間の中の最初のピクチャと呼ばれる。

【0102】

- b. GDRピクチャは、temporalIDが0に等しい。

【0103】

- c. GDRピクチャを含むアクセスユニットは、GDRアクセスユニットと呼ばれる。上述のように、アクセスユニットはNALユニットのセットである。各NALユニットは、単一のピクチャを含んでもよい。

【0104】

50

2) コーディングされたビデオシーケンス(CVS)は、GDRアクセスユニットで開始してもよい。

【0105】

3) 次のことのうちの1つが真であるとき、GDRアクセスユニットはCVSの中の最初のアクセスユニットである。

【0106】

a. GDRアクセスユニットがビットストリームの中の最初のアクセスユニットである。

【0107】

b. GDRアクセスユニットがエンドオブシーケンス(EOS)アクセスユニットの直後にくる。

【0108】

c. GDRアクセスユニットがエンドオブビットストリーム(EOB:end-of-bitstream)アクセスユニットの直後にくる。

【0109】

d. デコーダフラグ、いわゆる、NoIncorrectPicOutputFlagがGDRピクチャに関連付けられ、デコーダの外側のエンティティによってフラグの値が1(すなわち、真)に等しく設定される。

【0110】

4) GDRピクチャがCVSの中の最初のアクセスユニットであるとき、以下のことが適用される。

【0111】

a. DPBの中のすべての参照ピクチャが「参照のために未使用」としてマークされる。

【0112】

b. ピクチャのPOC MSBが0に等しくなるように設定される。

【0113】

c. GDRピクチャ、およびGDR期間の中の最後のピクチャを除外してGDR期間の中の最後のピクチャまで出力順序でGDRピクチャに後続するすべてのピクチャは、出力されない(すなわち、「出力のために不必要」としてマークされる)。

【0114】

5) GDRが有効化されているかどうかを指定するためのフラグが、シーケンスレベルパラメータセットの中で(たとえば、SPSの中で)シグナリングされる。

【0115】

a. そのフラグはgdr_enabled_flagと指定されてもよい。

【0116】

b. フラグが1に等しいとき、GDRピクチャがCVSの中に存在してもよい。そうではなく、フラグが0に等しいとき、GDRピクチャがCVSの中に存在しないようにGDRは有効化されていない。

【0117】

6) GDR期間の中の最後のピクチャのPOC値を導出するために使用され得る情報が、GDRピクチャのタイルグループヘッダの中でシグナリングされる。

【0118】

a. GDR期間の中の最後のピクチャとGDRピクチャとの間の差分(delta)POCとして、情報がシグナリングされる。その情報は、recovery_point_cntと指定されたシンタックス要素を使用してシグナリングされ得る。

【0119】

b. タイルグループヘッダの中のシンタックス要素recovery_point_cntの存在は、gdr_enabledフラグの値、およびピクチャのNALユニットタイプが条件とされてもよく、すなわち、gdr_enabled_flagが1に等しく、かつタイルグループを含むNALユニットのnal_unit_typeがGDR_NUTであるときのみ、フラグが存在する。

【0120】

10

20

30

40

50

7) タイルグループが、リフレッシュ済みの領域の一部であるか否かを指定するためのフラグが、タイルグループヘッダの中でシグナリングされる。

【0 1 2 1】

a. そのフラグは、refreshed_region_flagと指定されてもよい。

【0 1 2 2】

b. そのフラグの存在は、gdr_enabled_flagの値、およびタイルグループを含むピクチャがGDR期間内にあるかどうか、条件とされてもよい。したがって、次のことのすべてが真であるときのみ、フラグが存在する。

【0 1 2 3】

i. gdr_enabled_flagの値が1に等しい。

10

【0 1 2 4】

ii. 現在ピクチャのPOCが、最後のGDRピクチャのPOC値以上であり(現在ピクチャがGDRピクチャであるとき、最後のGDRピクチャが現在ピクチャである)、GDR期間の中の最後のピクチャのPOCよりも小さい。

【0 1 2 5】

c. フラグがタイルグループヘッダの中に存在しないとき、フラグの値は1に等しいものと推測される。

【0 1 2 6】

8) refreshed_region_flagが1に等しいすべてのタイルグループは、連結している領域をカバーする。同様に、refreshed_region_flagが0に等しいすべてのタイルグループも、連結している領域をカバーする。

20

【0 1 2 7】

9) refreshed_region_flagを有するタイルグループは、タイプI(すなわち、イントラタイルグループ)またはBもしくはP(すなわち、インタータイルグループ)のものであり得る。

【0 1 2 8】

10) GDRピクチャから開始しGDR期間の中の最後のピクチャまでの各ピクチャは、refreshed_region_flagが1に等しい少なくとも1つのタイルグループを含む。

【0 1 2 9】

11) GDRピクチャは、refreshed_region_flagが1に等しく、かつtile_group_typeがI(すなわち、イントラタイルグループ)に等しい、少なくとも1つのタイルグループを含む。

30

【0 1 3 0】

12) gdr_enabled_flagが1に等しいとき、長方形タイルグループの情報、すなわち、タイルグループの数およびそれらのアドレスが、ピクチャパラメータセット(PPS:picture parameter set)またはタイルグループヘッダのいずれかの中でシグナリングされることを許容される。これを行うために、長方形タイルグループ情報がPPSの中に存在するか否かを指定するために、PPSの中でフラグがシグナリングされる。このフラグは、rect_tile_group_info_in_pps_flagと呼ばれてもよい。このフラグは、gdr_enabled_flagが1に等しいとき、1に等しくなるように制約されてもよい。

【0 1 3 1】

40

a. 一代替形態では、長方形タイルグループ情報がPPSの中に存在するか否かをシグナリングするのではなく、タイルグループ情報(すなわち、長方形タイルグループ、ラスタ走査タイルグループなどの、任意のタイプのタイルグループ)がPPSの中に存在かどうかを指定するために、より全般的なフラグがPPSの中でシグナリングされ得る。

【0 1 3 2】

13) タイルグループ情報がPPSの中に存在しないとき、明示的なタイルグループ識別子(ID)情報のシグナリングが存在しないことが、さらに制約されてもよい。明示的なタイルグループID情報は、signaled_tile_group_id_flag、signaled_tile_group_id_length_minus1、およびtile_group_id[i]を含む。

【0 1 3 3】

50

14) ピクチャの中のリフレッシュ済みの領域とリフレッシュされていない領域との間の境界を横断するループフィルタ処理演算が許容されるかどうかを指定するために、フラグがシグナリングされる。

【0 1 3 4】

a. このフラグはPPSの中でシグナリングされてもよく、loop_filter_across_refreshed_region_enabled_flagと呼ばれてもよい。

【0 1 3 5】

b. loop_filter_across_refreshed_region_enabled_flagの存在は、loop_filter_across_tile_enabled_flagの値が条件とされてもよい。loop_filter_across_tile_enabled_flagが0に等しいとき、loop_filter_across_refreshed_region_enabled_flagは存在しなくてもよく、その値は0に等しいものと推測される。

10

【0 1 3 6】

c. 一代替形態では、タイルグループヘッダの中でそのフラグがシグナリングされてもよく、その存在は、refreshed_region_flagの値が条件とされてもよく、すなわち、refreshed_region_flagの値が1に等しいときのみ、そのフラグが存在する。

【0 1 3 7】

15) タイルグループがリフレッシュ済みの領域であることが示され、かつリフレッシュ済みの領域を横断するループフィルタが許容されないことが示されるとき、以下のことが適用される。

【0 1 3 8】

20

a. エッジを共有する隣接タイルグループが、リフレッシュされていないタイルグループであるとき、タイルグループの境界におけるエッジのデブロッキングが実行されない。

【0 1 3 9】

b. タイルグループの境界におけるブロックに対するサンプル適応オフセット(SAO:sample adaptive offset)プロセスは、リフレッシュ済み領域境界の外側からのいかなるサンプルも使用しない。

【0 1 4 0】

c. タイルグループの境界におけるブロックに対する適応ループフィルタ処理(ALF:adaptive loop filtering)プロセスは、リフレッシュ済み領域境界の外側からのいかなるサンプルも使用しない。

30

【0 1 4 1】

16) gdr_enabled_flagが1に等しいとき、各ピクチャは、ピクチャの中のリフレッシュ済みの領域の境界を決定するための変数に関連付けられる。これらの変数は、次のように呼ばれてもよい。

【0 1 4 2】

a. ピクチャの中のリフレッシュ済みの領域の左の境界位置に対する、PicRefreshedLeftBoundaryPos。

【0 1 4 3】

b. ピクチャの中のリフレッシュ済みの領域の右の境界位置に対する、PicRefreshedRightBoundaryPos。

40

【0 1 4 4】

c. ピクチャの中のリフレッシュ済みの領域の上の境界位置に対する、PicRefreshedTopBoundaryPos。

【0 1 4 5】

d. ピクチャの中のリフレッシュ済みの領域の下の境界位置に対する、PicRefreshedBottomBoundaryPos。

【0 1 4 6】

17) ピクチャの中のリフレッシュ済みの領域の境界が導出されてもよい。ピクチャのリフレッシュ済みの領域の境界は、タイルグループヘッダが構文解析された後にデコーダによって更新され、タイルグループのrefreshed_region_flagの値は1に等しい。

50

【 0 1 4 7 】

18) 解決策17)の一代替形態では、ピクチャの中のリフレッシュ済みの領域の境界は、ピクチャの各タイルグループの中で明示的にシグナリングされる。

【 0 1 4 8 】

a. タイルグループが属するピクチャが、リフレッシュされていない領域を含むかどうかを示すために、フラグがシグナリングされてもよい。リフレッシュされていない領域をピクチャが含まないことが指定されるとき、リフレッシュ済み境界情報はシグナリングされず、ピクチャ境界に等しいものと単に推測され得ない。

【 0 1 4 9 】

19) 現在ピクチャに対して、次のように、ループ内フィルタプロセスにおいてリフレッシュ済みの領域の境界が使用される。

10

【 0 1 5 0 】

a. デブロッキングプロセスの場合、エッジがデブロッキングされる必要があるか否かを定めるために、リフレッシュ済みの領域のエッジを決定する。

【 0 1 5 1 】

b. SAOプロセスの場合、リフレッシュ領域を横断するループフィルタが許容されないとき、リフレッシュされていない領域からのサンプルを使用することを回避するためにクリッピングプロセスが適用され得るように、リフレッシュ済みの領域の境界を決定する。

【 0 1 5 2 】

c. ALFプロセスの場合、リフレッシュ済みの領域を横断するループフィルタが許容されないとき、リフレッシュされていない領域からのサンプルを使用することを回避するためにクリッピングプロセスが適用され得るように、リフレッシュ領域の境界を決定する。

20

【 0 1 5 3 】

20) 動き補償プロセスに対して、リフレッシュ済みの領域の境界、特に参照ピクチャの中のリフレッシュ済みの領域の境界についての情報が、次のように使用される。すなわち、現在ピクチャの中の現在ブロックが、refreshed_region_flagが1に等しいタイルグループの中にあり、かつ参照ブロックが、リフレッシュされていない領域を含む参照ピクチャの中にあるとき、以下のことが適用される。

【 0 1 5 4 】

a. 現在ブロックからその参照ピクチャへの動きベクトルは、その参照ピクチャの中のリフレッシュ済みの領域の境界によってクリッピングされる。

30

【 0 1 5 5 】

b. その参照ピクチャの中のサンプルのための分数補間フィルタに対して、そうした動きベクトルはその参照ピクチャの中のリフレッシュ済みの領域の境界によってクリッピングされる。

【 0 1 5 6 】

本開示の実施形態の詳細な説明が提供される。説明はベーステキストに関連し、ベーステキストはJVET寄稿JVET-M1001-v5である。すなわち、差分だけが記載されるが、以下で述べられないベーステキストの中のテキストは、現状のままで適用される。ベーステキストに比べて修正されるテキストは、イタリック体が使用される。

40

【 0 1 5 7 】

定義が与えられる。

【 0 1 5 8 】

3.1 クリーンランダムアクセス(CRA)ピクチャ: CRA_NUTに等しいnal_unit_typeを各VCL NALユニットが有するIRAPピクチャ。

【 0 1 5 9 】

注 - CRAピクチャは、その復号プロセスにおけるインター予測のために、それ自体以外のいかなるピクチャも参照せず、復号順序でビットストリームの中の最初のピクチャであってもよく、または後でビットストリームの中に出現してもよい。CRAピクチャは、関連するRADLピクチャまたはRASLピクチャを有してもよい。1に等しいNoIncorrectPicOut

50

putFlagをCRAピクチャが有するとき、RASLピクチャは、ビットストリームの中に存在しないピクチャへの参照を含んでもよいときに復号可能でない場合があるので、関連するRASLピクチャはデコーダによって出力されない。

【0160】

3.2 コーディングされたビデオシーケンス(CVS): NoIncorrectPicOutputFlagが1に等しいIRAPアクセスユニットまたはNoIncorrectPicOutputFlagが1に等しいGDRアクセスユニットである後続の任意のアクセスユニットまでの(ただし、それを含まない)すべての後続のアクセスユニットを含む、NoIncorrectPicOutputFlagが1に等しいIRAPアクセスユニットまたはNoIncorrectPicOutputFlagが1に等しいGDRアクセスユニットと、それに後続する0、またはNoIncorrectPicOutputFlagが1に等しいIRAPアクセスユニットもしくはNoIncorrectPicOutputFlagが1に等しいGDRアクセスユニットでない、より多くのアクセスユニットとを復号順序で備える、アクセスユニットのシーケンス。

10

【0161】

注1 - IRAPアクセスユニットは、IDRアクセスユニットまたはCRAアクセスユニットであってもよい。各IDRアクセスユニットに対してNoIncorrectPicOutputFlagの値は1に等しく、復号順序でビットストリームの中の最初のアクセスユニットである各CRAアクセスユニットは、復号順序でエンドオブシーケンスNALユニットに後続するか、または1に等しいHandleCraAsCvsStartFlagを有する、最初のアクセスユニットである。

【0162】

注2 - 復号順序でビットストリームの中の最初のアクセスユニットである各GDRアクセスユニットに対して、NoIncorrectPicOutputFlagの値が1に等しいことは、復号順序でエンドオブシーケンスNALユニットに後続するか、または1に等しいHandleGdrAsCvsStartFlagを有する、最初のアクセスユニットである。

20

【0163】

3.3 漸進的復号リフレッシュ(GDR)アクセスユニット: コーディングされたピクチャがGDRピクチャであるアクセスユニット。

【0164】

3.4 漸進的復号リフレッシュ(GDR)ピクチャ: GDR_NUTに等しいnal_unit_typeを各VCL NALユニットが有するピクチャ。

【0165】

3.5 ランダムアクセススキップドリーディング(RASL)ピクチャ: RASL_NUTに等しいnal_unit_typeを各VCL NALユニットが有するコーディングされたピクチャ。

30

【0166】

注 - すべてのRASLピクチャは、関連するCRAピクチャのリーディングピクチャである。関連するCRAピクチャが、1に等しいNoIncorrectPicOutputFlagを有するとき、ビットストリームの中に存在しないピクチャへの参照をRASLピクチャが含んでもよいので、RASLピクチャは出力されず、正しく復号可能でない場合がある。RASLピクチャは、非RASLピクチャの復号プロセスのための参照ピクチャとして使用されない。存在するとき、すべてのRASLピクチャは、関連する同じCRAピクチャのすべてのトレーリングピクチャに復号順序で先行する。

40

【0167】

シーケンス・パラメータ・セット・ロー・バイト・シーケンス・ペイロード(RBSP:raw byte sequence payload)のシンタックスおよびセマンティック。

【表 1】

seq_parameter_set_rbsp() {	記述子
...	
<i>gdr_enabled_flag</i>	<i>u(1)</i>
...	

【 0 1 6 8 】

1に等しい*gdr_enabled_flag*は、コーディングされたビデオシーケンスの中にGDRピクチャが存在してもよいことを指定する。0に等しい*gdr_enabled_flag*は、コーディングされたビデオシーケンスの中にGDRピクチャが存在しないことを指定する。

【 0 1 6 9 】

ピクチャパラメータセットRBSPのシンタックスおよびセマンティック。

10

20

30

40

50

【表 2】

pic_parameter_set_rbsp() {	記述子
...	
single_tile_in_pic_flag	u(1)
if(!single_tile_in_pic_flag) {	
...	ue(v)
single_tile_per_tile_group_flag	u(1)
if(!single_tile_per_tile_group_flag)	
rect_tile_group_flag	u(1)
if(rect_tile_group_flag)	
rect_tile_group_info_in_pps_flag	u(1)
if(rect_tile_group_flag && !single_tile_per_tile_group_flag && rect_tile_group_info_in_pps_flag) {	
num_tile_groups_in_pic_minus1	ue(v)
for(i = 0; i <= num_tile_groups_in_pic_minus1; i++) {	
if(i > 0)	
top_left_tile_idx[i]	u(v)
bottom_right_tile_idx[i]	u(v)
}	
}	
loop_filter_across_tiles_enabled_flag	u(1)
if(loop_filter_across_tiles_enabled_flag)	
loop_filter_across_tile_groups_enabled_flag	u(1)
if(loop_filter_across_tile_groups_enabled_flag)	
loop_filter_across_refreshed_region_enabled_flag	u(1)
}	
...	

【 0 1 7 0 】

1に等しいrect_tile_group_info_in_pps_flagは、長方形タイルグループ情報がPPSの中でシグナリングされることを指定する。0に等しいrect_tile_group_info_in_pps_flagは、長方形タイルグループ情報がPPSの中でシグナリングされないことを指定する。

【 0 1 7 1 】

アクティブなSPSの中のgdr_enabled_flagの値が0に等しいとき、rect_tile_group_info_in_pps_flagの値が0に等しくなければならないことが、ビットストリーム適合の要件である。

【 0 1 7 2 】

1に等しいloop_filter_across_refreshed_region_enabled_flagは、PPSを参照するピクチャの中で、refreshed_region_flagが1に等しいタイルグループの境界を横断してループ内フィルタ処理演算が実行されてもよいことを指定する。0に等しいloop_filter_across_refreshed_region_enabled_flagは、PPSを参照するピクチャの中で、refreshed_region_flagが1に等しいタイルグループの境界を横断してループ内フィルタ処理演算が実行されないことを指定する。ループ内フィルタ処理演算は、デブロッキングフィルタ

、サンプル適応オフセットフィルタ、および適応ループフィルタ演算を含む。存在しないとき、loop_filter_across_refreshed_region_enabled_flagの値は0に等しいものと推測される。

【 0 1 7 3 】

1に等しいsignalled_tile_group_id_flagは、タイルグループごとのタイルグループIDがシグナリングされることを指定する。0に等しいsignalled_tile_group_index_flagは、タイルグループIDがシグナリングされないことを指定する。存在しないとき、signalled_tile_group_index_flagの値は0に等しいものと推測される。

【 0 1 7 4 】

signalled_tile_group_id_length_minus1+1は、存在するときシンタックス要素tile_group_id[i]を、かつタイルグループヘッダの中のシンタックス要素tile_group_addressを表すために使用される、ビットの数を指定する。signalled_tile_group_index_length_minus1の値は、両端値を含む0～15という範囲の中になければならない。存在しないとき、signalled_tile_group_index_length_minus1の値は次のように推測される。

【 0 1 7 5 】

rect_tile_group_info_in_pps_flagが1に等しい場合、 $\text{Ceil}(\text{Log2}(\text{num_tile_groups_in_pic_minus1} + 1)) - 1$ 。

【 0 1 7 6 】

そうでない場合、 $\text{Ceil}(\text{Log2}(\text{NumTilesInPic})) - 1$ 。

【 0 1 7 7 】

全般的なタイルグループヘッダのシンタックスおよびセマンティック。

【表 3】

tile_group_header() {	記述子
...	
if(rect_tile_group_flag NumTilesInPic > 1)	
tile_group_address	$u(v)$
if(!rect_tile_group_info_in_pps_flag && !single_tile_per_tile_group_flag) {	
if(rect_tile_group_flag)	
bottom_right_tile_id	$u(v)$
else	
num_tiles_in_tile_group_minus1	$ue(v)$
}	
tile_group_type	$ue(v)$
tile_group_pic_order_cnt_lsb	$u(v)$
if(gdr_enabled_flag) {	
if(nal_unit_type == GDR_NUT)	
recovery_poc_cnt	$se(v)$
if(PicOrderCntVal >= LastGDRPocVal && PicOrderCntVal < RecoveryPointPocVal)	
refreshed_region_flag	$u(1)$
}	
...	)

10

20

30

40

50

【 0 1 7 8 】

tile_group_addressは、タイルグループの中の最初のタイルのタイルアドレスを指定する。存在しないとき、tile_group_addressの値は、0に等しいものと推測される。

【 0 1 7 9 】

rect_tile_group_flagが0に等しい場合、以下のことが適用される。

tile_group_addressは、式6-7によって指定されるタイルIDである。

tile_group_addressの長さは、 $\text{Ceil}(\text{Log}_2(\text{NumTilesInPic}))$ ビットである。

tile_group_addressの値は、両端値を含む0 ~ NumTilesInPic - 1という範囲の中になければならない。

【 0 1 8 0 】

そうではなく、rect_tile_group_flagが1に等しく、かつrect_tile_group_info_in_ppsが0に等しい場合、以下のことが適用される。

tile_group_addressは、第iのタイルグループの左上隅角に位置するタイルのタイルインデックスである。

tile_group_addressの長さは、signalled_tile_group_index_length_minus1 + 1ビットである。

signalled_tile_group_id_flagが0に等しい場合、tile_group_addressの値は、両端値を含む0 ~ NumTilesInPic - 1という範囲の中になければならない。そうでない場合、tile_group_addressの値は、両端値を含む0 ~ $2(\text{signalled_tile_group_index_length_minus1} + 1) - 1$ という範囲の中になければならない。

【 0 1 8 1 】

それ以外の(rect_tile_group_flagが1に等しく、かつrect_tile_group_info_in_ppsが1に等しい)場合、以下のことが適用される。

tile_group_addressは、タイルグループのタイルグループIDである。

tile_group_addressの長さは、signalled_tile_group_index_length_minus1 + 1ビットである。

signalled_tile_group_id_flagが0に等しい場合、tile_group_addressの値は、両端値を含む0 ~ num_tile_groups_in_pic_minus1という範囲の中になければならない。そうでない場合、tile_group_addressの値は、両端値を含む0 ~ $2(\text{signalled_tile_group_index_length_minus1} + 1) - 1$ という範囲の中になければならない。

【 0 1 8 2 】

bottom_right_tile_idは、タイルグループの右下隅角に位置するタイルのタイルインデックスを指定する。single_tile_per_tile_group_flagが1に等しいとき、bottom_right_tile_idは、tile_group_addressに等しいものと推測される。bottom_right_tile_idシンタックス要素の長さは、 $\text{Ceil}(\text{Log}_2(\text{NumTilesInPic}))$ ビットである。

【 0 1 8 3 】

現在タイルグループの中のタイルの数を指定する変数NumTilesInCurrTileGroup、タイルグループの左上のタイルのタイルインデックスを指定するTopLeftTileIdx、タイルグループの右下のタイルのタイルインデックスを指定するBottomRightTileIdx、および現在タイルグループの中の第iのタイルのタイルインデックスを指定するTgTileIdx[i]が、次のように導出される。

```
if( rect_tile_group_flag ) {
    if ( tile_group_info_in_pps ) {
        tileGroupIdx = 0
        while( tile_group_address != rect_tile_group_id[ tileGroupIdx ] )
            tileGroupIdx++
        tileIdx = top_left_tile_idx[ tileGroupIdx ]
        BottomRightTileIdx = bottom_right_tile_idx[ tileGroupIdx ]
    } else {
        tileIdx = tile_group_address
```

10

20

30

40

50

```

        BottomRightTileIdx = bottom_right_tile_id
    }
    TopLeftTileIdx = tileIdx
    deltaTileIdx = BottomRightTileIdx - TopLeftTileIdx
    NumTileRowsInTileGroupMinus1 = deltaTileIdx / ( num_tile_columns_minus1 + 1 ) (7-35)
    NumTileColumnsInTileGroupMinus1 = deltaTileIdx % ( num_tile_columns_minus1 + 1 )
    NumTilesInCurrTileGroup = ( NumTileRowsInTileGroupMinus1 + 1 ) * (
    NumTileColumnsInTileGroupMinus1 + 1 )
    for( j = 0, tldx = 0; j < NumTileRowsInTileGroupMinus1 + 1; j++, tileIdx
    += num_tile_columns_minus1 + 1 )
        for( i = 0, currTileIdx = tileIdx; i < NumTileColumnsInTileGroupMinus1
    + 1; i++, currTileIdx++, tldx++ )
            TgTileIdx[ tldx ] = currTileIdx
    } else {
        NumTilesInCurrTileGroup = num_tiles_in_tile_group_minus1 + 1
        TgTileIdx[ 0 ] = tile_group_address
        for( i = 1; i < NumTilesInCurrTileGroup; i++ )
            TgTileIdx[ i ] = TgTileIdx[ i - 1 ] + 1
    }
}

```

10

20

【 0 1 8 4 】

recovery_poc_cntは、出力順序での復号ピクチャのリカバリポイントを指定する。CVSの中で復号順序で現在ピクチャ(すなわち、GDRピクチャ)に後続し、かつ現在ピクチャのPicOrderCntVal+recovery_poc_cntの値に等しいPicOrderCntValを有する、ピクチャpicAがある場合、ピクチャpicAは、リカバリポイントピクチャと呼ばれる。そうでない場合、現在ピクチャのPicOrderCntVal+recovery_poc_cntの値よりも大きいPicOrderCntValを有する、出力順序で最初のピクチャが、リカバリポイントピクチャと呼ばれる。リカバリポイントピクチャは、復号順序で現在ピクチャに先行してはならない。出力順序でのすべての復号ピクチャは、リカバリポイントピクチャの出力順序位置において開始する内容の中で正確またはほぼ正確であるものと示される。recovery_poc_cntの値は、両端値を含む $\text{MaxPicOrderCntLsb} / 2 \sim \text{MaxPicOrderCntLsb} / 2 - 1$ という範囲の中になければならない。

30

【 0 1 8 5 】

RecoveryPointPocValの値は、次のように導出される。

【 0 1 8 6 】

RecoveryPointPocVal = PicOrderCntVal + recovery_poc_cnt

【 0 1 8 7 】

1に等しいrefreshed_region_flagは、タイルグループの復号が、関連するGDRのNoIncorrectPicOutputFlagの値にかかわらず正確な再構成サンプル値を生成することを指定する。0に等しいrefreshed_region_flagは、タイルグループの復号が、NoIncorrectPicOutputFlagが1に等しい関連するGDRから開始するとき、不正確な再構成サンプル値を生成してもよいことを指定する。存在しないとき、refreshed_region_flagの値は、1に等しいものと推測される。

40

【 0 1 8 8 】

注x - 現在ピクチャ自体が、NoIncorrectPicOutputFlagが1に等しいGDRピクチャであり得る。

【 0 1 8 9 】

タイルグループがリフレッシュされる境界は、次のように導出される。

tileColIdx = TopLeftTileIdx % (num_tile_columns_minus1 + 1)

50

```

tileRowIdx = TopLeftTileIdx / ( num_tile_columns_minus1 + 1 )
TGR refreshedLeftBoundary = ColBd[ tileColIdx ]      CtbLog2SizeY
TGR refreshedTopBoundary = RowBd[ tileRowIdx ]      CtbLog2SizeY
tileColIdx = BottomRightTileIdx % ( num_tile_columns_minus1 + 1 )
tileRowIdx = BottomRightTileIdx / ( num_tile_columns_minus1 + 1 )
TGR refreshedRightBoundary = ( ( ColBd[ tileColIdx ] + ColWidth[ tileColIdx
] )      CtbLog2SizeY ) - 1
TGR refreshedRightBoundary = TGR refreshedRightBoundary    pic_width_in_
luma_samples ? pic_width_in_luma_samples : TGR refreshedRightBoundary
TGR refreshedBotBoundary = ( ( RowBd[ tileRowIdx ] + RowHeight[ tileRowI
dx ] )      CtbLog2SizeY ) - 1
TGR refreshedBotBoundary = TGR refreshedBotBoundary    pic_height_in_lu
ma_samples ? pic_height_in_luma_samples : TGR refreshedBotBoundary
【 0 1 9 0 】
NALユニットヘッダのセマンティック。

```

10

20

30

40

50

【表 4】

Table 7-1 - 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 の名称	NAL ユニットの内容及び RBSP シンタックス構造	NAL ユニ ットタイプク ラス
0	TRAIL_NUT	非 STSA トレーリングピクチャのコーディングされた タイルグループ tile_group_layer_rbsp()	VCL
1	STSA_NUT	STSA ピクチャのコーディングされたタイルグループ tile_group_layer_rbsp()	VCL
2	RASL_NUT	RASL ピクチャのコーディングされたタイルグループ tile_group_layer_rbsp()	VCL
3	RADL_NUT	RADL ピクチャのコーディングされたタイルグループ tile_group_layer_rbsp()	VCL
4..7	RSV_VCL_4.. RSV_VCL_7	予約済みの非 IRAP VCL NAL ユニットタイプ	VCL
8	IDR_W_RADL	IDR ピクチャのコーディングされたタイルグループ tile_group_layer_rbsp()	VCL
9	IDR_N_LP		
10	CRA_NUT	CRA ピクチャのコーディングされたタイルグループ tile_group_layer_rbsp()	VCL
11	RSV_IRAP_VCL	予約済みの IRAP VCL NAL ユニットタイプ	VCL
12	11 RSV_IRAP_VCL 12		
13	GDR_NUT	GDR ピクチャのコーディングされたタイルグループ tile_group_layer_rbsp()	VCL
14..15	RSV_VCL14.. RSV_VCL15	予約済みの非 IRAP VCL NAL ユニットタイプ	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	APS_NUT	適応パラメータセット adaptation_parameter_set_rbsp()	非 VCL
19	AUD_NUT	アクセスユニットデリミタ access_unit_delimiter_rbsp()	非 VCL
20	EOS_NUT	エンドオブシーケンス end_of_seq_rbsp()	非 VCL
21	EOB_NUT	エンドオブビットストリーム end_of_bitstream_rbsp()	非 VCL
22, 23	PREFIX_SEI_NU T SUFFIX_SEI_NU T	補足エンハンスメント情報 sei_rbsp()	非 VCL
24..27	RSV_NVCL24.. RSV_NVCL27	予約済みの非 VCL NAL ユニットタイプ	非 VCL
28..31	UNSPEC28.. UNSPEC31	未指定の非 VCL NAL ユニットタイプ	非 VCL

【 0 1 9 1 】

...

【 0 1 9 2 】

nal_unit_type が GDR_NUT に等しく、コーディングされたタイルグループが GDR ピクチャに属するとき、TemporalId は 0 に等しくなければならない。

【 0 1 9 3 】

アクセスユニットの順序および CVS への関連付けが説明される。

【 0 1 9 4 】

この仕様(すなわち、JVET 寄稿 JVET-M1001-v5)に適合するビットストリームは、1 つ

10

20

30

40

50

以上のCVSを含む。

【0195】

CVSは、1つ以上のアクセスユニットを含む。NALユニットおよびコーディングされたピクチャの順序、ならびにアクセスユニットへのそれらの関連付けが、第7.4.2.4.4節に記載される。

【0196】

CVSの最初のアクセスユニットは、以下のうちの1つである。

【0197】

- NoBrokenPictureOutputFlagが1に等しいIRAPアクセスユニット。

【0198】

- NoIncorrectPicOutputFlagが1に等しいGDRアクセスユニット。

【0199】

存在するとき、エンドオブシーケンスNALユニットまたはエンドオブビットストリームNALユニットを含むアクセスユニットの後の次のアクセスユニットが、以下のうちの1つでなければならないことが、ビットストリーム適合の要件である。

【0200】

- IDRアクセスユニットまたはCRAアクセスユニットであってもよいIRAPアクセスユニット。

【0201】

- GDRアクセスユニット。

【0202】

8.1.1 コーディングされたピクチャのための復号プロセスが説明される。

【0203】

...

【0204】

現在ピクチャがIRAPピクチャであるとき、以下のことが適用される。

【0205】

- 現在ピクチャが、IDRピクチャ、復号順序でビットストリームの中の最初のピクチャ、または復号順序でエンドオブシーケンスNALユニットに後続する最初のピクチャであるとき、変数NoIncorrectPicOutputFlagは、1に等しく設定される。

【0206】

- そうではなく、この仕様に指定されないいくつかの外部手段(たとえば、ユーザ入力)が、変数HandleCraAsCvsStartFlagを現在ピクチャに対する値に設定するために利用可能であるとき、変数HandleCraAsCvsStartFlagは、外部手段によって提供される値に等しく設定され、変数NoIncorrectPicOutputFlagは、HandleCraAsCvsStartFlagに等しく設定される。

【0207】

- そうでない場合、変数HandleCraAsCvsStartFlagは、0に等しく設定され、変数NoIncorrectPicOutputFlagは、0に等しく設定される。

【0208】

現在ピクチャがGDRピクチャであるとき、以下のことが適用される。

【0209】

- 現在ピクチャが、GDRピクチャ、復号順序でビットストリームの中の最初のピクチャ、または復号順序でエンドオブシーケンスNALユニットに後続する最初のピクチャであるとき、変数NoIncorrectPicOutputFlagは、1に等しく設定される。

【0210】

- そうではなく、この仕様に指定されないいくつかの外部手段が、変数HandleGdrAsCvsStartFlagを現在ピクチャに対する値に設定するために利用可能であるとき、変数HandleGdrAsCvsStartFlagは、外部手段によって提供される値に等しく設定され、変数NoIncorrectPicOutputFlagは、HandleGdrAsCvsStartFlagに等しく設定される。

10

20

30

40

50

【 0 2 1 1 】

- そうでない場合、変数HandleGdrAsCvsStartFlagは、0に等しく設定され、変数NoIncorrectPicOutputFlagは、0に等しく設定される。

【 0 2 1 2 】

...

【 0 2 1 3 】

現在ピクチャCurrPicに対して復号プロセスは次のように動作する。

【 0 2 1 4 】

1. NALユニットの復号が第8.2節で指定される。

【 0 2 1 5 】

2. 第8.3節におけるプロセスは、タイルグループヘッダレイヤの中のシンタックス要素を使用する以下の復号プロセス、および上記のことを指定する。

【 0 2 1 6 】

- ピクチャ順序カウントに関係する変数および関数が、第8.3.1節で指定されるように導出される。これは、ピクチャの最初のタイルグループに対してのみ呼び出される必要がある。

【 0 2 1 7 】

- 非IDRピクチャのタイルグループごとの復号プロセスの開始において、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、第8.3.2節で指定される参照ピクチャリスト構成のための復号プロセスが呼び出される。

【 0 2 1 8 】

- 第8.3.3節におけ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、「参照のために未使用」または「長期の参照のために使用済み」としてマークされてもよい。これは、ピクチャの最初のタイルグループに対してのみ呼び出される。

【 0 2 1 9 】

- PicOutputFlagは次のように設定される。

【 0 2 2 0 】

- 次の条件のうちの1つが真であるとき、PictureOutputFlagは、0に等しく設定される。

【 0 2 2 1 】

- 現在ピクチャがRASLピクチャであり、関連するIRAPピクチャのNoIncorrectPicOutputFlagが1に等しい。

【 0 2 2 2 】

- gdr_enabled_flagが1に等しく、現在ピクチャが、NoIncorrectPicOutputFlagが1に等しいGDRピクチャである。

【 0 2 2 3 】

- gdr_enabled_flagが1に等しく、現在ピクチャが、refreshed_region_flagが0に等しい1つ以上のタイルグループを含み、関連するGDRピクチャのNoBrokenPictureOutputFlagが1に等しい。

【 0 2 2 4 】

- そうでない場合、PicOutputFlagは、1に等しく設定される。

【 0 2 2 5 】

3. 復号プロセスは、ツリーユニットをコーディングすること、スケーリング、変換、ループ内フィルタ処理などのために呼び出される。

【 0 2 2 6 】

4. 現在ピクチャのすべてのタイルグループが復号された後、現在の復号ピクチャが「短期の参照のために使用済み」としてマークされる。

【 0 2 2 7 】

ピクチャ順序カウントのための復号プロセスが説明される。

【 0 2 2 8 】

10

20

30

40

50

このプロセスの出力は、PicOrderCntVal、すなわち、現在ピクチャのピクチャ順序カウントである。

【0229】

コーディングされた各ピクチャは、PicOrderCntValとして示されるピクチャ順序カウント変数に関連する。

【0230】

現在ピクチャが、NoIncorrectPicOutputFlagが1に等しいIRAPピクチャ、またはNoIncorrectPicOutputFlagが1に等しいGDRピクチャでないとき、変数prevPicOrderCntLsbおよびprevPicOrderCntMsbは、次のように導出される。

【0231】

- 0に等しいTemporalIdを有するとともにRASLまたはRADLピクチャでない、復号順序で前のピクチャを、prevTid0Picとする。

【0232】

- 変数prevPicOrderCntLsbは、prevTid0Picのtile_group_pic_order_cnt_lsbに等しく設定される。

【0233】

- 変数prevPicOrderCntMsbは、prevTid0PicのPicOrderCntMsbに等しく設定される。

【0234】

現在ピクチャの変数PicOrderCntMsbは、次のように導出される。

【0235】

- 現在ピクチャが、NoIncorrectPicOutputFlagが1に等しいIRAPピクチャ、またはNoIncorrectPicOutputFlagが1に等しいGDRピクチャであるとき、PicOrderCntMsbは、0に等しく設定される。

【0236】

- そうでない場合、PicOrderCntMsbは、次のように導出される。

if((tile_group_pic_order_cnt_lsb == prevPicOrderCntLsb) && ((prevPicOrderCntLsb - tile_group_pic_order_cnt_lsb) >= (MaxPicOrderCntLsb / 2)))

PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb (8-1)

else if((tile_group_pic_order_cnt_lsb == prevPicOrderCntLsb) && ((tile_group_pic_order_cnt_lsb - prevPicOrderCntLsb) < (MaxPicOrderCntLsb / 2)))

PicOrderCntMsb = prevPicOrderCntMsb - MaxPicOrderCntLsb

else

PicOrderCntMsb = prevPicOrderCntMsb

PicOrderCntValは、次のように導出される。

PicOrderCntVal = PicOrderCntMsb + tile_group_pic_order_cnt_lsb (8-2)

【0237】

注1 - NoIncorrectPicOutputFlagが1に等しいIRAPピクチャに対してPicOrderCntMsbが0に等しく設定されるので、NoIncorrectPicOutputFlagが1に等しいすべてのIRAPピクチャは、tile_group_pic_order_cnt_lsbに等しいPicOrderCntValを有する。

【0238】

注1 - NoIncorrectPicOutputFlagが1に等しいGDRピクチャに対してPicOrderCntMsbが0に等しく設定されるので、NoIncorrectPicOutputFlagが1に等しいすべてのGDRピクチャは、tile_group_pic_order_cnt_lsbに等しいPicOrderCntValを有する。

【0239】

PicOrderCntValの値は、両端値を含む $-2^{31} \sim 2^{31} - 1$ という範囲の中になければならない。

【0240】

現在ピクチャがGDRピクチャであるとき、LastGDRPocValの値は、PicOrderCntValに等しくなるように設定される。

10

20

30

40

50

【 0 2 4 1 】

ピクチャがリフレッシュされた境界位置のための復号プロセスが説明される。

【 0 2 4 2 】

このプロセスは、gdr_enabled_flagが1に等しいときにしか呼び出されない。

【 0 2 4 3 】

このプロセスは、タイルグループヘッダ構文解析が完了した後に呼び出される。

【 0 2 4 4 】

このプロセスの出力は、PicRefreshedLeftBoundaryPos、PicRefreshedRightBoundaryPos、PicRefreshedTopBoundaryPos、およびPicRefreshedBotBoundaryPos、すなわち、現在ピクチャのリフレッシュ済みの領域の境界位置である。

10

【 0 2 4 5 】

コーディングされた各ピクチャは、PicOrderCntValとして示されるリフレッシュ済み領域境界位置変数のセットに関連する。

【 0 2 4 6 】

PicRefreshedLeftBoundaryPos、PicRefreshedRightBoundaryPos、PicRefreshedTopBoundaryPos、およびPicRefreshedBotBoundaryPosは、次のように導出される。

【 0 2 4 7 】

タイルグループが、refreshed_region_flagが1に等しい現在ピクチャの最初の受信タイルグループである場合、以下のことが適用される。

20

【 0 2 4 8 】

PicRefreshedLeftBoundaryPos = TGRRefreshedLeftBoundary

【 0 2 4 9 】

PicRefreshedRightBoundaryPos = TGRRefreshedRightBoundary

【 0 2 5 0 】

PicRefreshedTopBoundaryPos = TGRRefreshedTopBoundary

【 0 2 5 1 】

PicRefreshedBotBoundaryPos = TileGroupBotBoundary

【 0 2 5 2 】

そうではなく、refreshed_region_flagが1に等しい場合、以下のことが適用される。

30

【 0 2 5 3 】

PicRefreshedLeftBoundaryPos = TGRRefreshedLeftBoundary PicRefreshedLeftBoundaryPos ?

【 0 2 5 4 】

TGRRefreshedLeftBoundary : PicRefreshedLeftBoundaryPos

【 0 2 5 5 】

PicRefreshedRightBoundaryPos = TGRRefreshedRightBoundary PicRefreshedRightBoundaryPos ?

【 0 2 5 6 】

TGRRefreshedRightBoundary : PicRefreshedRightBoundaryPos

40

【 0 2 5 7 】

PicRefreshedTopBoundaryPos = TGRRefreshedTopBoundary PicRefreshedTopBoundaryPos ?

【 0 2 5 8 】

TGRRefreshedTopBoundary : RefreshedRegionTopBoundaryPos

【 0 2 5 9 】

PicRefreshedBotBoundaryPos = TileGroupBotBoundary PicRefreshedBotBoundaryPos ?

【 0 2 6 0 】

TileGroupBotBoundary : PicRefreshedBotBoundaryPos

50

【 0 2 6 1 】

参照ピクチャリスト構成のための復号プロセスが説明される。

【 0 2 6 2 】

...

【 0 2 6 3 】

NoIncorrectPicOutputFlagが1に等しいIRAPピクチャまたはNoIncorrectPicOutputFlagが1に等しいGDRピクチャでない、各現在ピクチャに対して、maxPicOrderCnt - minPicOrderCntの値がMaxPicOrderCntLsb / 2よりも小さくなければならないことが、ビットストリーム適合の要件である。

【 0 2 6 4 】

10

...

【 0 2 6 5 】

参照ピクチャマーキングのための復号プロセス。

【 0 2 6 6 】

...

【 0 2 6 7 】

現在ピクチャが、NoIncorrectPicOutputFlagが1に等しいIRAPピクチャまたはNoIncorrectPicOutputFlagが1に等しいGDRピクチャである場合、(もしあれば)現在、DPBの中の、すべての参照ピクチャは、「参照のために未使用」としてマークされる。

【 0 2 6 8 】

20

...

【 0 2 6 9 】

時間ルーマ動きベクトル予測のための導出プロセスが説明される。

【 0 2 7 0 】

...

【 0 2 7 1 】

変数currCbは、ルーマロケーション(xCb, yCb)における現在ルーマコーディングブロックを指定する。

【 0 2 7 2 】

変数mvLXColおよびavailableFlagLXColは、次のように導出される。

30

【 0 2 7 3 】

- tile_group_temporal_mvp_enabled_flagが0に等しい場合、mvLXColの両方の成分は、0に等しく設定され、availableFlagLXColは、0に等しく設定される。

【 0 2 7 4 】

- そうでない(tile_group_temporal_mvp_enabled_flagが1に等しい)場合、以下の順序付きステップが適用される。

【 0 2 7 5 】

1. 右下のコロケート動きベクトルが、次のように導出される。

【 0 2 7 6 】

$xColBr = xCb + cbWidth$ (8-414)

40

【 0 2 7 7 】

$yColBr = yCb + cbHeight$ (8-415)

【 0 2 7 8 】

leftBoundaryPos = gdr_enabled_flag ?

【 0 2 7 9 】

RefPicList[X][refIdxLX]によって参照されるピクチャのPicRefreshedLeftBoundaryPos :

【 0 2 8 0 】

0 (8-415)

【 0 2 8 1 】

50

topBoundaryPos = gdr_enabled_flag ?
 【 0 2 8 2 】
 RefPicList[X][refIdxLX]によって参照されるピクチャのPicRefreshedTopBoundaryPos :
 【 0 2 8 3 】
 0 (8-415)
 【 0 2 8 4 】
 rightBoundaryPos = gdr_enabled_flag ?
 【 0 2 8 5 】
 RefPicList[X][refIdxLX]によって参照されるピクチャのPicRefreshedRightBoundaryPos : 10
 【 0 2 8 6 】
 pic_width_in_luma_samples (8-415)
 【 0 2 8 7 】
 botBoundaryPos = gdr_enabled_flag ?
 【 0 2 8 8 】
 RefPicList[X][refIdxLX]によって参照されるピクチャのPicRefreshedBotBoundaryPos :
 【 0 2 8 9 】
 pic_height_in_luma_samples (8-415) 20
 【 0 2 9 0 】
 - yCb CtbLog2SizeYがyColBr CtbLog2SizeYに等しく、yColBrが、両端値を含むtopBoundaryPosからbotBoundaryPosまでの範囲の中にあり、かつxColBrが、両端値を含むleftBoundaryPosからrightBoundaryPosまでの範囲の中にある場合、以下のことが適用される。
 【 0 2 9 1 】
 - 変数colCbは、ColPicによって指定されるコロケートピクチャの内側の、((xColBr 3) 3, (yColBr 3) 3)によって与えられる修正済みのロケーションをカバーするルーマコーディングブロックを指定する。
 【 0 2 9 2 】 30
 - ルーマロケーション(xColCb, yColCb)は、ColPicによって指定されるコロケートピクチャの左上のルーマサンプルに対してcolCbによって指定されるコロケートルーマコーディングブロックの左上のサンプルに等しく設定される。
 【 0 2 9 3 】
 - 第8.5.2.12節で指定されるようなコロケート動きベクトルのための導出プロセスは、入力としてcurrCb、colCb、(xColCb, yColCb), refIdxLX、および0に等しく設定されたsbFlagとともに呼び出され、出力がmvLXColおよびavailableFlagLXColに割り当てられる。
 【 0 2 9 4 】
 - そうでない場合、mvLXColの両方の成分は、0に等しく設定され、availableFlagLXColは、0に等しく設定される。 40
 【 0 2 9 5 】
 2. ...
 【 0 2 9 6 】
 ルーマサンプル双線形補間プロセスが説明される。
 【 0 2 9 7 】
 このプロセスの入力は、以下の通りである。
 【 0 2 9 8 】
 - フルサンプル単位でのルーマロケーション(xIntL, yIntL)、
 【 0 2 9 9 】 50

- 分数サンプル単位でのルーマロケーション(x_{FracL} , y_{FracL})、
【 0 3 0 0 】
 - ルーマ参照サンプルアレイ $refPicLX_L$ 、
【 0 3 0 1 】
 - 参照ピクチャのリフレッシュ済み領域境界 $PicRefreshedLeftBoundaryPos$ 、 $PicRefreshedTopBoundaryPos$ 、 $PicRefreshedRightBoundaryPos$ 、および $PicRefreshedBotBoundaryPos$ 。
【 0 3 0 2 】
 ...
【 0 3 0 3 】 10
 フルサンプル単位でのルーマロケーション(x_{Int_i} , y_{Int_i})は、 $i = 0..1$ に対して次のように導出される。
【 0 3 0 4 】
 - $gdr_enabled_flag$ が1に等しい場合、以下のことが適用される。
【 0 3 0 5 】
 $x_{Int_i} = Clip3(PicRefreshedLeftBoundaryPos, PicRefreshedRightBoundaryPos, x_{Int_L} + i)$ (8-458)
【 0 3 0 6 】
 $y_{Int_i} = Clip3(PicRefreshedTopBoundaryPos, PicRefreshedBotBoundaryPos, y_{Int_L} + i)$ (8-458) 20
【 0 3 0 7 】
 - そうでない($gdr_enabled_flag$ が0に等しい)場合、以下のことが適用される。
【 0 3 0 8 】
 $x_{Int_i} = sps_ref_wraparound_enabled_flag ?$
【 0 3 0 9 】
 $ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, (x_{Int_L} + i))$: (8-459)
【 0 3 1 0 】
 $Clip3(0, picW - 1, x_{Int_L} + i)$
【 0 3 1 1 】 30
 $y_{Int_i} = Clip3(0, picH - 1, y_{Int_L} + i)$ (8-460)
【 0 3 1 2 】
 ...
【 0 3 1 3 】
 ルーマサンプル8タップ補間フィルタ処理プロセスが説明される。
【 0 3 1 4 】
 このプロセスの入力は、以下の通りである。
【 0 3 1 5 】
 - フルサンプル単位でのルーマロケーション(x_{Int_L} , y_{Int_L})、
【 0 3 1 6 】 40
 - 分数サンプル単位でのルーマロケーション(x_{FracL} , y_{FracL})、
【 0 3 1 7 】
 - ルーマ参照サンプルアレイ $refPicLX_L$ 、
【 0 3 1 8 】
 - 参照サンプルパディングの方向および量を指定する、 $dir = 0,1$ を伴うリスト $padVal[dir]$ 。
【 0 3 1 9 】
 - 参照ピクチャのリフレッシュ済み領域境界 $PicRefreshedLeftBoundaryPos$ 、 $PicRefreshedTopBoundaryPos$ 、 $PicRefreshedRightBoundaryPos$ 、および $PicRefreshedBotBoundaryPos$ 。 50

【 0 3 2 0 】

...

【 0 3 2 1 】

フルサンプル単位でのルーマロケーション($xInt_i$, $yInt_i$)は、 $i = 0..7$ に対して次のように導出される。

【 0 3 2 2 】

- $gdr_enabled_flag$ が1に等しい場合、以下のことが適用される。

【 0 3 2 3 】

$xInt_i = Clip3(PicRefreshedLeftBoundaryPos, PicRefreshedRightBoundaryPos, xInt_L + i - 3)$ (8-830)

10

【 0 3 2 4 】

$yInt_i = Clip3(PicRefreshedTopBoundaryPos, PicRefreshedBotBoundaryPos, yInt_L + i - 3)$ (8-830)

【 0 3 2 5 】

- そうでない($gdr_enabled_flag$ が0に等しい)場合、以下のことが適用される。

【 0 3 2 6 】

$xInt_i = sps_ref_wraparound_enabled_flag ?$

【 0 3 2 7 】

$ClipH((sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY, picW, xInt_L + i - 3)$: (8-831)

20

【 0 3 2 8 】

$Clip3(0, picW - 1, xInt_L + i - 3)$

【 0 3 2 9 】

$yInt_i = Clip3(0, picH - 1, yInt_L + i - 3)$ (8-832)

【 0 3 3 0 】

クロマサンプル補間プロセスが説明される。

【 0 3 3 1 】

このプロセスの入力は、以下の通りである。

【 0 3 3 2 】

- フルサンプル単位でのクロマロケーション($xInt_C$, $yInt_C$)、

30

【 0 3 3 3 】

- $1/32$ 分数サンプル単位でのクロマロケーション($xFrac_C$, $yFrac_C$)、

【 0 3 3 4 】

- クロマ参照サンプルアレイ $refPicLXC$ 。

【 0 3 3 5 】

- 参照ピクチャのリフレッシュ済み領域境界 $PicRefreshedLeftBoundaryPos$ 、 $PicRefreshedTopBoundaryPos$ 、 $PicRefreshedRightBoundaryPos$ 、および $PicRefreshedBotBoundaryPos$ 。

【 0 3 3 6 】

...

40

【 0 3 3 7 】

変数 $xOffset$ は、 $(sps_ref_wraparound_offset_minus1 + 1) * MinCbSizeY) / SubWidthC$ に等しく設定される。

【 0 3 3 8 】

フルサンプル単位でのクロマロケーション($xInt_i$, $yInt_i$)は、 $i = 0..3$ に対して次のように導出される。

【 0 3 3 9 】

- $gdr_enabled_flag$ が1に等しい場合、以下のことが適用される。

【 0 3 4 0 】

$xInt_i = Clip3(PicRefreshedLeftBoundaryPos / SubWidthC,$

50

【 0 3 4 1 】

PicRefreshedRightBoundaryPos / SubWidthC, xInt_L + i) (8-844)

【 0 3 4 2 】

yInt_i = Clip3(PicRefreshedTopBoundaryPos / SubHeightC,

【 0 3 4 3 】

PicRefreshedBotBoundaryPos / SubHeightC, yInt_L + i) (8-844)

【 0 3 4 4 】

- そうでない(gdr_enabled_flagが0に等しい)場合、以下のことが適用される。

【 0 3 4 5 】

xInt_i = sps_ref_wraparound_enabled_flag ? ClipH(xOffset, picW_C, xInt_C + i - 1) : (8-845) 10

【 0 3 4 6 】

Clip3(0, picW_C - 1, xInt_C + i - 1)

【 0 3 4 7 】

yInt_i = Clip3(0, picH_C - 1, yInt_C + i - 1) (8-846)

【 0 3 4 8 】

デブロッキングフィルタプロセスが説明される。

【 0 3 4 9 】

全般的なプロセス。

【 0 3 5 0 】

20

...

【 0 3 5 1 】

デブロッキングフィルタプロセスは、以下のタイプのエッジを除いて、ピクチャのすべてのコーディングサブブロックエッジおよび変換ブロックエッジに適用される。

【 0 3 5 2 】

- ピクチャの境界にあるエッジ。

【 0 3 5 3 】

- 以下のことのすべてが満たされるとき、タイルグループtgAの上の境界に一致するエッジ。

【 0 3 5 4 】

30

- gdr_enabled_flagが1に等しい。

【 0 3 5 5 】

- loop_filter_across_refreshed_region_enabled_flagが0に等しい。

【 0 3 5 6 】

- そのエッジがタイルグループtgBの下境界に一致し、tgBのrefreshed_region_flagの値がtgAのrefreshed_region_flagの値とは異なる。

【 0 3 5 7 】

- 以下のことのすべてが満たされるとき、タイルグループtgAの左の境界に一致するエッジ。

【 0 3 5 8 】

40

- gdr_enabled_flagが1に等しい。

【 0 3 5 9 】

- loop_filter_across_refreshed_region_enabled_flagが0に等しい。

【 0 3 6 0 】

- そのエッジがタイルグループtgBの右の境界に一致し、tgBのrefreshed_region_flagの値がtgAのrefreshed_region_flagの値とは異なる。

【 0 3 6 1 】

- loop_filter_across_tiles_enabled_flagが0に等しいとき、タイル境界に一致するエッジ。

【 0 3 6 2 】

50

- tile_group_loop_filter_across_tile_groups_enabled_flagが0に等しいかまたは tile_group_deblocking_filter_disabled_flagが1に等しいタイルグループの、上または左の境界に一致するエッジ。

【0363】

- tile_group_deblocking_filter_disabled_flagが1に等しいタイルグループ内のエッジ。

【0364】

- 考慮される成分の8×8サンプルグリッド境界に対応しないエッジ。

【0365】

- エッジの両側がインター予測を使用するクロマ成分内のエッジ。

10

【0366】

- 関連する変換ユニットのエッジでないクロマ変換ブロックのエッジ。

【0367】

- IntraSubPartitionsSplit値がISP_NO_SPLITに等しくないコーディングユニットのルーマ変換ブロックを横断するエッジ。

【0368】

- 一方向のためのデブロッキングフィルタプロセスが説明される。

【0369】

...

【0370】

コーディングブロック幅log2CbW、コーディングブロック高さlog2CbH、およびコーディングブロックの左上のサンプルのロケーション(xCb, yCb)を有するコーディングユニットごとに、edgeTypeがEDGE_VERに等しくxCb % 8が0に等しいとき、またはedgeTypeがEDGE_HORに等しくyCb % 8が0に等しいとき、以下の順序付きステップによってエッジがフィルタ処理される。

20

【0371】

1. コーディングブロック幅nCbWが、1 log2CbWに等しく設定され、コーディングブロック高さnCbHが、1 log2CbHに等しく設定される。

【0372】

2. 変数filterEdgeFlagが、次のように導出される。

30

【0373】

- edgeTypeがEDGE_VERに等しく、かつ次の条件のうちの1つ以上が真である場合、filterEdgeFlagは、0に等しく設定される。

【0374】

- 現在コーディングブロックの左の境界がピクチャの左の境界である。

【0375】

- 現在コーディングブロックの左の境界がタイルの左の境界であり、loop_filter_across_tiles_enabled_flagが0に等しい。

【0376】

- 現在コーディングブロックの左の境界がタイルグループの左の境界であり、tile_group_loop_filter_across_tile_groups_enabled_flagが0に等しい。

40

【0377】

- 現在コーディングブロックの左の境界が現在タイルグループの左の境界であり、次のすべての条件が満たされる。

【0378】

- gdr_enabled_flagが1に等しい。

【0379】

- loop_filter_across_refreshed_region_enabled_flagが0に等しい。

【0380】

- 現在タイルグループの左の境界と境界を共有したタイルグループが存在し、そのrefr

50

eshed_region_flagの値が現在タイルグループのrefreshed_region_flagの値とは異なる。

【 0 3 8 1 】

- そうではなく、edgeTypeがEDGE_HORに等しく、かつ次の条件のうちの1つ以上が真である場合、変数filterEdgeFlagは、0に等しく設定される。

【 0 3 8 2 】

- 現在ルーマコーディングブロックの上の境界がピクチャの上の境界である。

【 0 3 8 3 】

- 現在コーディングブロックの上の境界がタイルの上の境界であり、loop_filter_across_tiles_enabled_flagが0に等しい。

【 0 3 8 4 】

- 現在コーディングブロックの上の境界がタイルグループの上の境界であり、tile_group_loop_filter_across_tile_groups_enabled_flagが0に等しい。

【 0 3 8 5 】

- 現在コーディングブロックの上の境界が現在タイルグループの上の境界であり、次のすべての条件が満たされる。

【 0 3 8 6 】

- gdr_enabled_flagが1に等しい。

【 0 3 8 7 】

- loop_filter_across_refreshed_region_enabled_flagが0に等しい。

【 0 3 8 8 】

- 現在タイルグループの上の境界と境界を共有したタイルグループが存在し、そのrefreshed_region_flagの値が現在タイルグループのrefreshed_region_flagの値とは異なる。

【 0 3 8 9 】

- そうでない場合、filterEdgeFlagは、1に等しく設定される。

【 0 3 9 0 】

タイルが統合されると、シンタックスを適合させる。

【 0 3 9 1 】

3. 二次元(nCbW) × (nCbH)アレイedgeFlagsのすべての要素は、0に等しくなるように初期化される。

【 0 3 9 2 】

SAOのためのCTB修正プロセスが説明される。

【 0 3 9 3 】

...

【 0 3 9 4 】

i = 0..nCbSw - 1かつj = 0..nCbSh - 1を伴うすべてのサンプルロケーション(xSi, ySj)および(xYi, yYj)に対して、recPicture[xSi][ySj]をカバーするコーディングブロックを含むコーディングユニットのpcm_loop_filter_disabled_flag、pcm_flag[xYi][yYj]、およびcu_transquant_bypass_flagの値に応じて、以下のことが適用される。

【 0 3 9 5 】

-

【 0 3 9 6 】

将来決定変換/量子化バイパスにおいて保留中の強調されたセクションを修正する。

【 0 3 9 7 】

- そうではなく、SaoTypeIdx[cldx][rx][ry]が2に等しい場合、以下の順序付きステップが適用される。

【 0 3 9 8 】

1. k = 0..1に対するhPos[k]およびvPos[k]の値が、SaoEoClass[cldx][rx][ry]に基づいて表8-18の中で指定される。

10

20

30

40

50

【 0 3 9 9 】

2. 変数edgeIdxが、次のように導出される。

【 0 4 0 0 】

- 修正済みのサンプルロケーション($x_{S_{ik}'}$, $y_{S_{jk}'}$)および($x_{Y_{ik}'}$, $y_{Y_{ik}'}$)が、次のように導出される。

【 0 4 0 1 】

($x_{S_{ik}'}$, $y_{S_{jk}'}$) = (x_{Si} + $hPos[k]$, y_{Sj} + $vPos[k]$) (8-1128)

【 0 4 0 2 】

($x_{Y_{ik}'}$, $y_{Y_{jk}'}$) = ($cldx == 0$) ? ($x_{S_{ik}'}$, $y_{S_{jk}'}$) : ($x_{S_{ik}'} * SubWidthC$, $y_{S_{jk}'} * SubHeightC$) (8-1129)

10

【 0 4 0 3 】

- $k = 0..1$ を伴うすべてのサンプルロケーション($x_{S_{ik}'}$, $y_{S_{jk}'}$)および($x_{Y_{ik}'}$, $y_{Y_{jk}'}$)に対して次の条件のうちの1つ以上が真である場合、edgeIdxは、0に等しく設定される。

【 0 4 0 4 】

- ロケーション($x_{S_{ik}'}$, $y_{S_{jk}'}$)におけるサンプルが、ピクチャ境界の外側にある。

【 0 4 0 5 】

- $gdr_enabled_flag$ が1に等しく、 $loop_filter_across_refreshed_region_enabled_flag$ が0に等しく、現在タイルグループの $refreshed_region_flag$ が1に等しく、かつロケーション($x_{S_{ik}'}$, $y_{S_{jk}'}$)におけるサンプルを含むタイルグループの $refreshed_region_flag$ が0に等しい。

20

【 0 4 0 6 】

- ロケーション($x_{S_{ik}'}$, $y_{S_{jk}'}$)におけるサンプルが、異なるタイルグループに属し、次の2つの条件のうちの1つが真である。

【 0 4 0 7 】

- $MinTbAddrZs[x_{Y_{ik}'} \quad MinTbLog2SizeY][y_{Y_{jk}'} \quad MinTbLog2SizeY]$ が $MinTbAddrZs[x_{Y_i} \quad MinTbLog2SizeY][y_{Y_j} \quad MinTbLog2SizeY]$ よりも小さく、サンプル $recPicture[x_{Si}][y_{Sj}]$ が属するタイルグループの中の $tile_group_loop_filter_across_tile_groups_enabled_flag$ が0に等しい。

【 0 4 0 8 】

- $MinTbAddrZs[x_{Y_i} \quad MinTbLog2SizeY][y_{Y_j} \quad MinTbLog2SizeY]$ が $MinTbAddrZs[x_{Y_{ik}'} \quad MinTbLog2SizeY][y_{Y_{jk}'} \quad MinTbLog2SizeY]$ よりも小さく、サンプル $recPicture[x_{S_{ik}'}][y_{S_{jk}'}]$ が属するタイルグループの中の $tile_group_loop_filter_across_tile_groups_enabled_flag$ が0に等しい。

30

【 0 4 0 9 】

- $loop_filter_across_tiles_enabled_flag$ が0に等しく、ロケーション($x_{S_{ik}'}$, $y_{S_{jk}'}$)におけるサンプルが、異なるタイルに属する。

【 0 4 1 0 】

タイルグループを有しないタイルが組み込まれるとき、強調されたセクションを修正する。

【 0 4 1 1 】

- そうでない場合、edgeIdxは、次のように導出される。

【 0 4 1 2 】

- 以下のことが適用される。

【 0 4 1 3 】

$edgeIdx = 2 + Sign(recPicture[x_{Si}][y_{Sj}] - recPicture[x_{Si} + hPos[0]][y_{Sj} + vPos[0]]) +$

【 0 4 1 4 】

$Sign(recPicture[x_{Si}][y_{Sj}] - recPicture[x_{Si} + hPos[1]][y_{Sj} + vPos[1]])$ (8-1130)

【 0 4 1 5 】

50

- edgeIdxが0、1、または2に等しいとき、edgeIdxは次のように修正される。

【0 4 1 6】

edgeIdx = (edgeIdx == 2) ? 0 : (edgeIdx + 1) (8-1131)

【0 4 1 7】

3. 修正済みのピクチャサンプルアレイsaoPicture[xSi][ySj]が、次のように導出される。

【0 4 1 8】

saoPicture[xSi][ySj] = Clip3(0, (1 - bitDepth) - 1, recPicture[xSi][ySj] +

【0 4 1 9】

SaoOffsetVal[cldx][rx][ry][edgeIdx]) (8-1132)

10

【0 4 2 0】

ALFのためのルーマサンプルに対するコーディングツリーブロックフィルタ処理プロセスが説明される。

【0 4 2 1】

...

【0 4 2 2】

フィルタ処理済みの再構成ルーマサンプルalfPictureL[x][y]の導出のために、現在のルーマコーディングツリーブロックの内側の各再構成ルーマサンプルrecPictureL[x][y]が、x, y = 0..CtbSizeY - 1を伴って次のようにフィルタ処理される。

【0 4 2 3】

20

- ...

【0 4 2 4】

- ルーマサンプルの所与のアレイrecPictureの内側の対応するルーマサンプル(x, y)の各々に対するロケーション(hx, vy)は、次のように導出される。

【0 4 2 5】

- gdr_enabled_flagが1に等しく、loop_filter_across_refreshed_region_enabled_flagが0に等しく、ロケーション(x, y)におけるルーマサンプルを含むタイルグループtgAのrefreshed_region_flagが1に等しい場合、以下のことが適用される。

【0 4 2 6】

- ロケーション(hx, vy)が別のタイルグループtgBの中に位置し、かつtgBのrefreshed_region_flagが0に等しい場合、変数leftBoundary、rightBoundary、topBoundary、およびbotBoundaryは、それぞれ、TGRefreshedLeftBoundary、TGRefreshedRightBoundary、TGRefreshedTopBoundary、およびTGRefreshedBotBoundaryに等しく設定される。

【0 4 2 7】

- そうでない場合、変数leftBoundary、rightBoundary、topBoundary、およびbotBoundaryは、それぞれ、PicRefreshedLeftBoundaryPos、PicRefreshedRightBoundaryPos、PicRefreshedTopBoundaryPos、およびPicRefreshedBotBoundaryPosに等しく設定される。

【0 4 2 8】

40

hx = Clip3(leftBoundary, rightBoundary, xCtb + x) (8-1140)

【0 4 2 9】

vy = Clip3(topBoundary, botBoundary, yCtb + y) (8-1141)

【0 4 3 0】

- そうでない場合、以下のことが適用される。

【0 4 3 1】

hx = Clip3(0, pic_width_in_luma_samples - 1, xCtb + x) (8-1140)

【0 4 3 2】

vy = Clip3(0, pic_height_in_luma_samples - 1, yCtb + y) (8-1141)

【0 4 3 3】

50

- ...

【 0 4 3 4 】

ルーマサンプルに対するALF転置およびフィルタインデックスのための導出プロセスが説明される。

【 0 4 3 5 】

...

【 0 4 3 6 】

ルーマサンプルの所与のアレイrecPictureの内側の対応するルーマサンプル(x , y)の各々に対するロケーション(h_x , v_y)が、次のように導出される。

【 0 4 3 7 】

- `gdr_enabled_flag`が1に等しく、`loop_filter_across_refreshed_region_enabled_flag`が0に等しく、ロケーション(x , y)におけるルーマサンプルを含むタイルグループtgAの`refreshed_region_flag`が1に等しい場合、以下のことが適用される。

【 0 4 3 8 】

- ロケーション(h_x , v_y)が別のタイルグループtgBの中に位置し、かつtgBの`refreshed_region_flag`が0に等しい場合、変数`leftBoundary`、`rightBoundary`、`topBoundary`、および`botBoundary`は、それぞれ、`TGRefreshedLeftBoundary`、`TGRefreshedRightBoundary`、`TGRefreshedTopBoundary`、および`TGRefreshedBotBoundary`に等しく設定される。

【 0 4 3 9 】

- そうでない場合、変数`leftBoundary`、`rightBoundary`、`topBoundary`、および`botBoundary`は、それぞれ、`PicRefreshedLeftBoundaryPos`、`PicRefreshedRightBoundaryPos`、`PicRefreshedTopBoundaryPos`、および`PicRefreshedBotBoundaryPos`に等しく設定される。

【 0 4 4 0 】

$h_x = \text{Clip3}(\text{leftBoundary}, \text{rightBoundary}, x)$ (8-1140)

【 0 4 4 1 】

$v_y = \text{Clip3}(\text{topBoundary}, \text{botBoundary}, y)$ (8-1141)

【 0 4 4 2 】

- そうでない場合、以下のことが適用される。

【 0 4 4 3 】

$h_x = \text{Clip3}(0, \text{pic_width_in_luma_samples} - 1, x)$ (8-1145)

【 0 4 4 4 】

$v_y = \text{Clip3}(0, \text{pic_height_in_luma_samples} - 1, y)$ (8-1146)

【 0 4 4 5 】

クロマサンプルのためのコーディングツリーブロックフィルタ処理プロセスが説明される。

【 0 4 4 6 】

...

【 0 4 4 7 】

フィルタ処理済みの再構成クロマサンプル`alfPicture[ x ][ y ]`の導出のために、現在のクロマコーディングツリーブロックの内側の各再構成クロマサンプル`recPicture[ x ][ y ]`が、 $x, y = 0..ctbSizeC - 1$ を伴って次のようにフィルタ処理される。

【 0 4 4 8 】

- クロマサンプルの所与のアレイrecPictureの内側の対応するクロマサンプル(x , y)の各々に対するロケーション(h_x , v_y)が、次のように導出される。

【 0 4 4 9 】

- `gdr_enabled_flag`が1に等しく、`loop_filter_across_refreshed_region_enabled_flag`が0に等しく、ロケーション(x , y)におけるルーマサンプルを含むタイルグループtgAの`refreshed_region_flag`が1に等しい場合、以下のことが適用される。

【 0 4 5 0 】

- ロケーション(h_x , v_y)が別のタイルグループtgBの中に位置し、かつtgBのrefreshed_region_flagが0に等しい場合、変数leftBoundary、rightBoundary、topBoundary、およびbotBoundaryは、それぞれ、TGRefreshedLeftBoundary、TGRefreshedRightBoundary、TGRefreshedTopBoundary、およびTGRefreshedBotBoundaryに等しく設定される。

【 0 4 5 1 】

- そうでない場合、変数leftBoundary、rightBoundary、topBoundary、およびbotBoundaryは、それぞれ、PicRefreshedLeftBoundaryPos、PicRefreshedRightBoundaryPos、PicRefreshedTopBoundaryPos、およびPicRefreshedBotBoundaryPosに等しく設定される。

10

【 0 4 5 2 】

$h_x = \text{Clip3}(\text{leftBoundary} / \text{SubWidthC}, \text{rightBoundary} / \text{SubWidthC}, \text{xCtbC} + \text{x})$ (8-1140)

【 0 4 5 3 】

$v_y = \text{Clip3}(\text{topBoundary} / \text{SubWidthC}, \text{botBoundary} / \text{SubWidthC}, \text{yCtbC} + \text{y})$ (8-1141)

【 0 4 5 4 】

- そうでない場合、以下のことが適用される。

【 0 4 5 5 】

$h_x = \text{Clip3}(0, \text{pic_width_in_luma_samples} / \text{SubWidthC} - 1, \text{xCtbC} + \text{x})$ (8-1177)

20

【 0 4 5 6 】

$v_y = \text{Clip3}(0, \text{pic_height_in_luma_samples} / \text{SubHeightC} - 1, \text{yCtbC} + \text{y})$ (8-1178)

【 0 4 5 7 】

図7は、本開示の一実施形態に係る、漸進的復号リフレッシュ(GDR)技法700を実施するように構成されたビデオビットストリーム750を示す。GDR技法700は、図5のGDR技法500と類似であってもよい。本明細書で使用するビデオビットストリーム750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も呼ばれてもよい。図7に示すように、ビットストリーム750は、シーケンスパラメータセット(SPS:sequence parameter set)752、ピクチャパラメータセット(PPS:picture parameter set)754、スライスヘッダ756、および画像データ758を備える。

30

【 0 4 5 8 】

SPS752は、ピクチャのシーケンス(SOP:sequence of pictures)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、PPS754は、ピクチャ全体に共通のデータを含む。スライスヘッダ756は、たとえば、スライスタイプ、参照ピクチャのうちのどれが使用されるのかなどの、現在スライスについての情報を含む。SPS752およびPPS754は、パラメータセットと総称されてもよい。SPS752、PPS754、およびスライスヘッダ756は、ネットワークアブストラクションレイヤ(NAL)ユニットのタイプである。NALユニットは、後続すべきデータ(たとえば、コーディングされたビデオデータ)のタイプの表示を含むシンタックス構造である。NALユニットは、ビデオコーディングレイヤ(VCL)NALユニットおよび非VCL NALユニットに分類される。VCL NALユニットは、ビデオピクチャの中のサンプルの値を表すデータを含み、非VCL NALユニットは、パラメータセット(多数のVCL NALユニットに適用され得る重要なヘッダデータ)などの関連する任意の追加情報、および補足エンハンスメント情報(タイミング情報、および復号ビデオ信号の有用性を向上させ得るが、ビデオピクチャの中のサンプルの値を復号するために必要でない、他の追加データ)を含む。ビットストリーム750が、実際の適用例では他のパラメータおよび情報を含んでもよいことを、当業者は諒解されよう。

40

【 0 4 5 9 】

図7の画像データ758は、符号化中または復号中の画像またはビデオに関連するデータ

50

を備える。画像データ758は、単に、ペイロード、またはビットストリーム750の中で搬送中のデータと呼ばれてもよい。一実施形態では、画像データ758は、GDRピクチャ702、1つ以上のトレーリングピクチャ704、およびリカバリポイントピクチャ706を含む、CVS708(または、CLVS)を備える。一実施形態では、トレーリングピクチャ704は、GDR期間の中でリカバリポイントピクチャ706に先行するので、GDRピクチャの形式と見なされてもよい。

【0460】

一実施形態では、GDRピクチャ702、トレーリングピクチャ704、およびリカバリポイントピクチャ706は、CVS708の中のGDR期間を規定してもよい。一実施形態では、復号順序は、GDRピクチャ702で始まり、トレーリングピクチャ704に続き、次いで、リカバリピクチャ706に進む。

10

【0461】

ユーザインターフェース84を介してビデオデコーダ30によって値(たとえば、1)が受け取られると、GDRピクチャ702および出力順序(たとえば、提示順序410)でGDRピクチャ702とリカバリポイントピクチャ706との間の任意のトレーリングピクチャ704が出力されることを防止するために、第1のフラグはユーザインターフェース(たとえば、外部入力)によって提供される値に等しく設定され、第2のフラグは第1のフラグに等しく設定される。そうではなく、ユーザインターフェース84を介してビデオデコーダ30によって値が受け取られないとき、第1のフラグおよび第2のフラグは異なる値(たとえば、0)に等しく設定される。一実施形態では、第1のフラグが、ユーザインターフェースによって提供される値に等しく設定されるとき、GDRピクチャ702だけが出力されることを防止される。

20

【0462】

CVS708は、ビデオビットストリーム750の中のすべてのコーディングされたレイヤビデオシーケンス(CLVS)に対するコーディングされたビデオシーケンスである。特に、ビデオビットストリーム750が単一のレイヤを含むとき、CVSとCLVSとは同じである。ビデオビットストリーム750が複数のレイヤを含むときのみ、CVSとCLVSとは異なる。

【0463】

図7に示すように、GDR技法700または原理は、GDRピクチャ702で開始しリカバリポイントピクチャ706で終了する一連のピクチャにわたって機能する。GDR技法700、GDRピクチャ702、トレーリングピクチャ704、およびリカバリポイントピクチャ706は、図5のGDR技法500、GDRピクチャ502、トレーリングピクチャ504、およびリカバリポイントピクチャ506と類似である。したがって、簡潔のために、GDR技法700が実施される方式は、図7に関して繰り返さない。

30

【0464】

図7に示すように、CVS708の中のGDRピクチャ702、トレーリングピクチャ704、およびリカバリポイントピクチャ706は各々、それら自体のVCL NALユニット730内に含まれる。CVS708の中のVCL NALユニット730のセットは、アクセスユニットと呼ばれてもよい。

【0465】

CVS708の中のGDRピクチャ702を含むNALユニット730は、GDR NALユニットタイプ(GDR_NUT)を有する。すなわち、一実施形態では、CVS708の中のGDRピクチャ702を含むNALユニット730は、トレーリングピクチャ704およびリカバリポイントピクチャ706に対してそれ自体の固有のNALユニットタイプを有する。一実施形態では、GDR_NUTは、ビットストリーム750がIRAPピクチャで始まる必要があるのではなく、ビットストリーム750がGDRピクチャ702で始まることを可能にする。GDRピクチャ702のVCL NALユニット730をGDR_NUTとして指定することは、CVS708の中の初期VCL NALユニット730がGDRピクチャ702を含むことを、たとえば、デコーダに示してもよい。

40

【0466】

一実施形態では、GDRピクチャ702はCVS708の中の初期ピクチャである。一実施形態では、GDRピクチャ702はGDR期間の中の初期ピクチャである。一実施形態では、GDRピ

50

クチャ702は0に等しい時間識別子(ID)を有する。時間IDは、他のピクチャに対するピクチャの位置または順序を識別する値または数である。一実施形態では、GDR_NUTを有するVCL NALユニット730を含むアクセスユニットは、GDRアクセスユニットと指定される。一実施形態では、GDRピクチャ702は、別の(たとえば、より大きい)GDRピクチャのコーディングされたスライスである。すなわち、GDRピクチャ702は、より大きいGDRピクチャの一部であってもよい。

【0467】

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を復号する方法800の一実施形態である。方法800は、ビデオエンコーダ(たとえば、ビデオエンコーダ20)から復号ビットストリームが直接または間接的に受信された後に実行されてもよい。方法800は、IRAPピクチャを使用する必要なく順次イントラリフレッシュがランダムアクセスを有効化することを可能とするので、本方法は復号プロセスを改善する。GDRピクチャが出力されることを防止するために、フラグの値はユーザインターフェース(たとえば、図3の中のユーザインターフェース84、またはいくつかの他の外部手段)を介してビデオデコーダのユーザによって設定される。一実施形態では、GDRピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、フラグの値はユーザインターフェース(たとえば、図3の中のユーザインターフェース84、またはいくつかの他の外部手段)を介してビデオデコーダのユーザによって設定される。このようにしてフラグを設定することは、潜在的にダーティなデータがディスプレイに出力されることを防止し、ビデオデコーダがユーザの選好に従って動作することを可能にする。したがって、実際問題として、コーデックの性能が改善され、そのことはより良好なユーザエクスペリエンスにつながる。

【0468】

ブロック802において、ビデオデコーダは、第1のフラグに対する値が外部入力(たとえば、図3の中のユーザインターフェース84、またはいくつかの他の外部手段)によって提供されるかどうかを決定する。一実施形態では、外部入力は、ビデオデコーダのグラフィックユーザインターフェース(GUI)である。一実施形態では、ビデオデコーダのユーザが、外部入力を使用して第1のフラグの値を提供する。一実施形態では、第1のフラグはHandleGdrAsCvsStartFlagと指定される。

【0469】

ブロック804において、ビデオデコーダは、第1のフラグに対する値が外部入力によって提供されるとき、漸進的復号リフレッシュ(GDR)ピクチャ(たとえば、GDRピクチャ702)および出力順序でGDRピクチャ702とリカバリポイントピクチャ706との間の任意のトレーリングピクチャ704が出力されることを防止するために、第1のフラグを、外部入力によって提供される値に等しく、かつ第2のフラグを第1のフラグに等しく設定する。一実施形態では、GDRピクチャおよび出力順序でGDRピクチャとリカバリポイントピクチャとの間の任意のトレーリングピクチャが出力されることを防止するために、第1のフラグの値は1に設定される。一実施形態では、第1のフラグに対する値が外部入力によって提供されないとき、第1のフラグの値は0に設定される。

【0470】

一実施形態では、GDRピクチャは、コーディングされたビデオビットストリームのCVSの中の初期ピクチャである。一実施形態では、GDRピクチャは、コーディングされたビデオビットストリームのレイヤの中の初期ピクチャである。一実施形態では、レイヤは、コーディングされたビデオビットストリームのCVSのCLVSである。

【0471】

ブロック806において、ビデオデコーダは、GDRピクチャを復号する。トレーリングピクチャおよびリカバリポイントピクチャが、次いで、順に復号される。ブロック808において、ビデオデコーダは、GDRピクチャを復号ピクチャバッファ(DPB)の中に記憶する。一実施形態では、第1および第2のフラグの設定によって、GDRピクチャが出力されること

を制限されていないとき、GDRピクチャに基づいて生成された画像が、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)のユーザのために表示されてもよい。

【0472】

図9は、本開示の一実施形態に係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、本明細書で説明するような、開示する実施形態を実施するのに適している。ビ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、または中央演算処理装置(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はまた、光信号または電気信号の出口または入口のために、入口ポート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、光電気(OE:optical-to-electrical)構成要素および電気光(EO:electrical-to-optical)構成要素を備えてもよい。

10

【0473】

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つ以上の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)、およびデジタル信号プロセッサ(DSP)として実装されてもよい。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、上記で説明した開示する実施形態を実施する。たとえば、コーディングモジュール970は、様々なコーデック機能を実施、処理、準備、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の機能への大幅な改善をもたらし、異なる状態へのビデオコーディングデバイス900の変換に影響を及ぼす。代替的に、コーディングモジュール970は、メモリ960の中に記憶されプロセッサ930によって実行される命令として実装される。

20

【0474】

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通信するための入力および/または出力(I/O)デバイス980を含んでもよい。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、出力デバイスを含んでもよい。I/Oデバイス980はまた、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相互作用するための対応するインターフェースを含んでもよい。一実施形態では、I/Oデバイス980は、第1のフラグの値を入力するためにビデオコーディングデバイス900のユーザによって利用される外部手段である。

30

【0475】

メモリ960は、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、またプログラム実行中に読み取られる命令およびデータを記憶するために、1つ以上のディスク、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、オーバーフローデータ記憶デバイスとして使用されてもよい。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)、3元連想メモリ(TCAM:ternary content-addressable memory)、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。

40

【0476】

図10は、コーディングするための手段1000の一実施形態の概略図である。一実施形態では、コーディングするための手段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)の中に実装される。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

50

信するか、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へ送信するか、または復号画像を表示手段(たとえば、I/Oデバイス1080のうちの1つ)へ送信するように構成される。

【0477】

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、本明細書で開示する方法を実行するために、記憶手段1003の中に記憶された命令を実行するよ

10

【0478】

本明細書に記載する例示的な方法のステップが、必ずしも説明した順序で実行されることを必要とされるとは限らないことも理解されたく、そのような方法のステップの順序は、単に例であるものと理解されたい。同様に、そのような方法の中に追加のステップが含まれてもよく、いくつかのステップは、本開示の様々な実施形態に一致する方法の中で省略されても、組み合わせられてもよい。

【0479】

本開示ではいくつかの実施形態が提供されているが、開示するシステムおよび方法が、本開示の趣旨または範囲から逸脱することなく、多くの他の特定の形態で具現されてもよいことを理解されたい。本例は限定的ではなく例示的と見なされるべきであり、その意図は本明細書において与えられる詳細に限定されない。たとえば、様々な要素または構成要素が別のシステムの中で組み合わせられても、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ても、もしくは実施されなくてもよい。

20

【0480】

加えて、様々な実施形態において個別または別個として説明および図示される技法、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、または方法と組み合わせられても、または統合されてもよい。結合されるかもしくは直接結合されるか、または互いに通信するものとして、図示または説明される他の項目は、電気的か、機械的か、または別の方法であるかにかかわらず、いくつかのインターフェース、デバイス、または中間構成要素を通じて、間接的に結合されても、または通信していてもよい。変更、置換、および改変の他の例は、当業者によって確認可能であり、本明細書で開示する趣旨および範囲から逸脱することなく行うことができる。

30

【符号の説明】

【0481】

- 10 コーディングシステム
- 12 ソースデバイス、ビデオデバイス
- 14 宛先デバイス
- 16 コンピュータ可読媒体
- 18 ビデオソース
- 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 ディスプレイデバイス
- 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

40

50

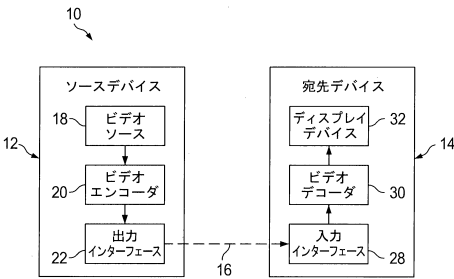
48	区分ユニット	
50	加算器	
52	変換処理ユニット	
54	量子化ユニット	
56	エントロピーコーディングユニット、エントロピー符号化ユニット	
58	逆量子化ユニット	
60	逆変換ユニット	
62	加算器	
64	参照フレームメモリ	
70	エントロピー復号ユニット	10
72	動き補償ユニット	
74	イントラ予測ユニット	
76	逆量子化ユニット	
78	逆変換ユニット	
80	加算器	
82	参照フレームメモリ	
84	ユーザインターフェース(UI)	
402	IRAPピクチャ	
404	リーディングピクチャ	
406	トレーリングピクチャ	20
408	復号順序	
410	提示順序	
502	GDRピクチャ	
504	トレーリングピクチャ	
506	リカバリポイントピクチャ	
508	コーディングされたビデオシーケンス	
510	リフレッシュ済みの/クリーンな領域	
512	リフレッシュされていない/ダーティな領域	
602	現在ピクチャ	
604	参照ピクチャ、リフレッシュ済みの領域	30
606	リフレッシュ済みの領域	
608	リフレッシュされていない領域、リフレッシュ済みの領域	
610	動きベクトル	
612	参照ブロック	
614	現在ブロック	
702	GDRピクチャ	
704	トレーリングピクチャ	
706	リカバリポイントピクチャ	
708	CVS	
730	NALユニット	40
750	ビデオビットストリーム	
752	シーケンスパラメータセット(SPS)	
754	ピクチャパラメータセット(PPS)	
756	スライスヘッダ	
758	画像データ	
900	ビデオコーディングデバイス	
910	入口ポート	
920	受信機ユニット(Rx)	
930	プロセッサ、論理ユニット、中央演算処理装置(CPU)	
940	送信機ユニット(Tx)	50

- 950 出口ポート
- 960 メモリ
- 970 コーディングモジュール
- 980 入力および/または出力(I/O)デバイス
- 1000 コーディングするための手段
- 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

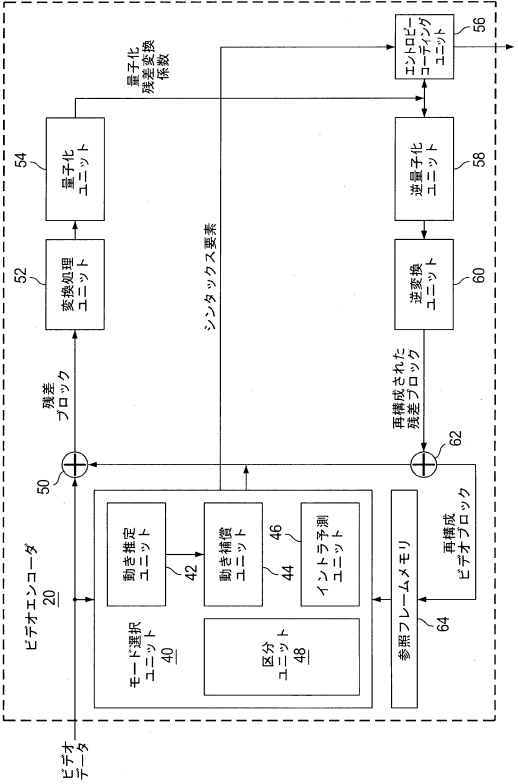
10

【図面】

【図 1】



【図 2】



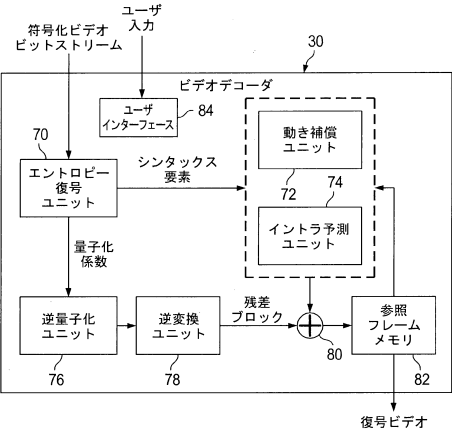
20

30

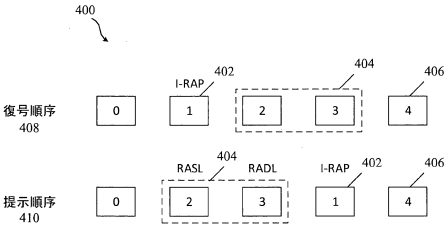
40

50

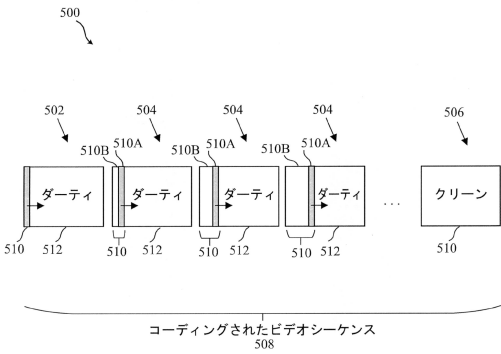
【図 3】



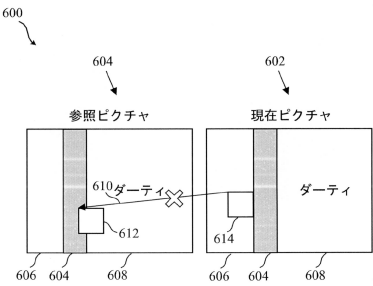
【図 4】



【図 5】



【図 6】



10

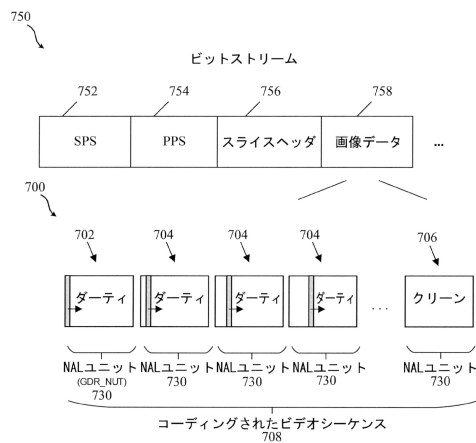
20

30

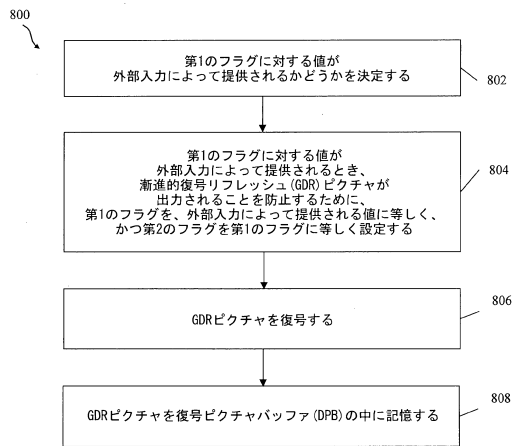
40

50

【図 7】

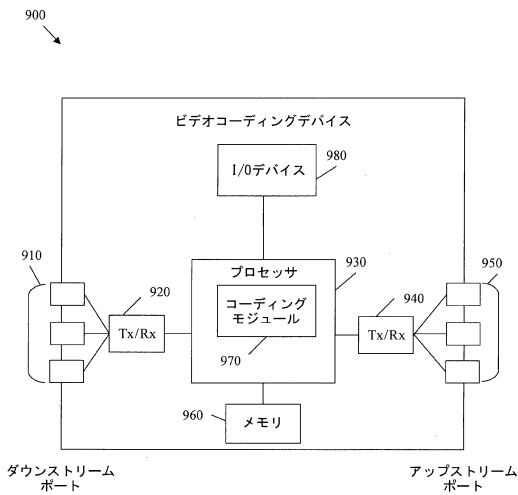


【図 8】

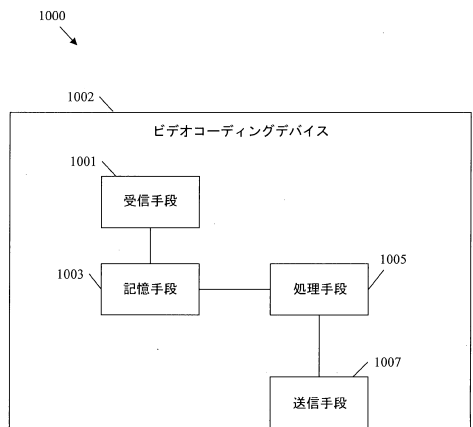


10

【図 9】



【図 10】



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)

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

(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・92130・サン・ディエゴ・コルト・デ・ティブロン・10
756

審査官 間宮 嘉誉

(56)参考文献 特表2022-524618(JP, A)

米国特許出願公開第2016/0112712(US, A1)

国際公開第2013/163569(WO, A1)

BOYCE, Jill, BoG Report on High Level Syntax, JVET-M0816 (version 3), ITU, 2019年0
1月15日, pp.1-17SKUPIN, Robert et al., AHG17: On VVC HLS, JVET-M0101 (version 3), ITU, 2019年01
月12日, pp.1-11

(58)調査した分野 (Int.Cl., DB名)

H04N 7/12

H04N 19/00-19/98

IEEE Explore