

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7615210号
(P7615210)

(45)発行日 令和7年1月16日(2025.1.16)

(24)登録日 令和7年1月7日(2025.1.7)

(51)国際特許分類

F I

H O 4 N 19/70 (2014.01)

H O 4 N 19/70

H O 4 N 19/436 (2014.01)

H O 4 N 19/436

請求項の数 21 外国語出願 (全57頁)

(21)出願番号	特願2023-76273(P2023-76273)	(73)特許権者	504161984
(22)出願日	令和5年5月2日(2023.5.2)		ホアウェイ・テクノロジーズ・カンパニ
(62)分割の表示	特願2021-540001(P2021-540001)		ー・リミテッド
原出願日	令和2年1月9日(2020.1.9)		中華人民共和国・518129・グァン
(65)公開番号	特開2023-101529(P2023-101529)		ドン・シェンツェン・ロンガン・ディス
(43)公開日	令和5年7月21日(2023.7.21)		トリクト・バンティアン・(番地なし)
審査請求日	令和5年6月1日(2023.6.1)		・ホアウェイ・アドミニストレーション
(31)優先権主張番号	62/790,207	(74)代理人	100110364
(32)優先日	平成31年1月9日(2019.1.9)		弁理士 実広 信哉
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	100133569
前置審査			弁理士 野村 進
		(72)発明者	イエークイ・ワン
			アメリカ合衆国・カリフォルニア・92
			130・サン・ディエゴ・サンローズ・
			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】

デコーダ内に実装される方法であって、
ピクチャから区分されたサブピクチャと、パラメータを備えるシーケンス・パラメータ・セット(SPS)とを備えるビットストリームを受信するステップであって、前記パラメータが、前記サブピクチャのサブピクチャサイズおよび前記サブピクチャのサブピクチャ位置についてのものであり、前記サブピクチャ位置が、前記ピクチャ内の前記サブピクチャの位置を示し、前記SPSが、前記ピクチャから区分された各サブピクチャについてのサブピクチャ識別子(ID)をさらに備える、ステップと、
前記SPSを解析して前記サブピクチャサイズおよび前記サブピクチャ位置についての前記パラメータを取得するステップと、
前記サブピクチャサイズおよび前記サブピクチャ位置に基づいて前記サブピクチャを生成してビデオシーケンスを作り出すステップと、
を備える方法。

【請求項2】

前記サブピクチャサイズに基づいてディスプレイのサイズに対する前記サブピクチャのサイズを決定するステップをさらに備える、請求項1に記載の方法。

【請求項3】

前記サブピクチャ位置に基づいてディスプレイに対する前記サブピクチャの位置を決定するステップをさらに備える、請求項1から2のいずれか一項に記載の方法。

【請求項 4】

前記サブピクチャ位置が前記サブピクチャの左上サンプルと前記ピクチャの左上サンプルとの間のオフセット距離を含む、請求項1から3のいずれか一項に記載の方法。

【請求項 5】

前記サブピクチャサイズが、ルマ・コーディング・ツリー・ユニット(CTU)サイズの単位でのサブピクチャ高さおよびルマCTUサイズの単位でのサブピクチャ幅を含む、請求項1から4のいずれか一項に記載の方法。

【請求項 6】

エンコーダ内に実装される方法であって、
ピクチャから区分されたサブピクチャをビットストリームへとエンコードするステップと、

10

前記サブピクチャのサブピクチャサイズおよびサブピクチャ位置を前記ビットストリーム内のシーケンス・パラメータ・セット(SPS)へとエンコードするステップであって、前記サブピクチャ位置が、前記ピクチャ内の前記サブピクチャの位置を示す、ステップと、
前記ピクチャから区分された各サブピクチャについてのサブピクチャ識別子(ID)を前記SPSへとエンコードするステップと、
を備える方法。

【請求項 7】

前記方法が、
前記エンコーダのメモリに、デコーダに向けた通信のために前記ビットストリームを記憶するステップをさらに備える、請求項6に記載の方法。

20

【請求項 8】

前記サブピクチャサイズおよび前記サブピクチャ位置が、前記サブピクチャのレイアウトを示す、請求項6から7のいずれか一項に記載の方法。

【請求項 9】

前記サブピクチャ位置が、前記サブピクチャの左上サンプルと前記ピクチャの左上サンプルとの間のオフセット距離を含む、請求項6から8のいずれか一項に記載の方法。

【請求項 10】

前記サブピクチャサイズが、ルマ・コーディング・ツリー・ユニット(CTU)サイズの単位でのサブピクチャ高さおよびルマCTUサイズの単位でのサブピクチャ幅を含む、請求項6から9のいずれか一項に記載の方法。

30

【請求項 11】

前記ピクチャから区分されたサブピクチャの数を前記SPS内にエンコードするステップをさらに備える、請求項6から10のいずれか一項に記載の方法。

【請求項 12】

ビデオコーディングデバイスによって実行されるとき前記ビデオコーディングデバイスに請求項1から11のいずれか一項に記載の方法を実行させるコンピュータ実行可能命令を備える、非一時的コンピュータ可読媒体。

【請求項 13】

ピクチャから区分されたサブピクチャと、パラメータを備えるシーケンス・パラメータ・セット(SPS)とを備えるビットストリームを受信するように構成された受信ユニットであって、前記パラメータが、前記サブピクチャのサブピクチャサイズおよび前記サブピクチャのサブピクチャ位置についてのものであり、前記サブピクチャ位置が、前記ピクチャ内の前記サブピクチャの位置を示し、前記SPSが、前記ピクチャから区分された各サブピクチャについてのサブピクチャ識別子(ID)をさらに備える、受信ユニットと、

40

前記SPSを解析して前記サブピクチャサイズおよび前記サブピクチャ位置についての前記パラメータを取得するように構成された解析ユニットと、

前記サブピクチャサイズおよび前記サブピクチャ位置に基づいて前記サブピクチャを生成してビデオシーケンスを作り出すように構成された生成ユニットと、

を備えるデコーダ。

50

【請求項 14】

前記デコーダが、請求項2から5のいずれか一項に記載の方法を実行するようにさらに構成された、請求項13に記載のデコーダ。

【請求項 15】

ピクチャから区分されたサブピクチャをビットストリームへとエンコードし、

前記サブピクチャのサブピクチャサイズおよびサブピクチャ位置を前記ビットストリーム内のシーケンス・パラメータ・セット(SPS)へとエンコードし、

前記ピクチャから区分された各サブピクチャについてのサブピクチャ識別子(ID)を前記SPSへとエンコードするように構成されたエンコードユニットであって、前記サブピクチャ位置が、前記ピクチャ内の前記サブピクチャの位置を示す、エンコードユニットを備えるエンコーダ。

10

【請求項 16】

前記エンコーダが、

デコーダに向けた通信のために前記ビットストリームを記憶するように構成された記憶ユニットをさらに備える、請求項15に記載のエンコーダ。

【請求項 17】

前記エンコーダが、請求項8から11のいずれか一項に記載の方法を実行するようにさらに構成された、請求項15から16のいずれか一項に記載のエンコーダ。

【請求項 18】

請求項1から5のいずれか一項に記載の方法を実行するための処理回路を備えるデコーダ。

20

【請求項 19】

請求項6から11のいずれか一項に記載の方法を実行するための処理回路を備えるエンコーダ。

【請求項 20】

コンピュータまたはプロセッサにおいて実行されるとき、請求項1から11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 21】

エンコードされたビットストリームを含むデータを記憶するように構成されたメモリであって、前記ビットストリームは、ピクチャから区分されたサブピクチャと、パラメータを備えるシーケンス・パラメータ・セット(SPS)とを備え、前記パラメータが、前記サブピクチャのサブピクチャサイズおよび前記サブピクチャのサブピクチャ位置についてのものであり、前記サブピクチャ位置が、前記ピクチャ内の前記サブピクチャの位置を示し、前記SPSが、前記ピクチャから区分された各サブピクチャについてのサブピクチャ識別子(ID)をさらに備える、メモリと、

30

要求に応じて前記ビットストリームをデコーダに向けて送信するように構成された送信機と、

を備える電子デバイス。

【発明の詳細な説明】

【技術分野】

40

【0001】

本開示は一般にビデオコーディングに関し、具体的にはビデオコーディングにおけるサブピクチャ管理に関する。

【背景技術】

【0002】

比較的短いビデオでさえ描写するために必要とされるビデオデータの量はかなりであることがあり、これは、限られた帯域幅容量を有する通信ネットワークにわたって、データがストリーミングされ、またはそうでなければ伝達されるとき、困難をもたらし得る。従って、ビデオデータは一般に、現代の電気通信ネットワークにわたって伝達される前に圧縮される。メモリリソースは限られ得るので、ビデオが記憶デバイスに記憶されるとき、

50

ビデオのサイズも問題であり得る。ビデオ圧縮デバイスはしばしば、ソースにおける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、それにより、デジタルビデオ画像を表現するために必要とされるデータの量を減らす。圧縮されたデータは次いで、ビデオデータをデコードするビデオ伸長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、およびより高いビデオ品質のいっそう増加する要求により、画像品質においてほとんどまたはまったく犠牲なく圧縮比を改善する、改善された圧縮および伸長技法が望ましい。

【発明の概要】

【課題を解決するための手段】

【0003】

実施形態では、開示は、デコーダ内に実装される方法であって、デコーダの受信機によって、ピクチャから区分されたサブピクチャを備えるビットストリームと、サブピクチャのサブピクチャサイズおよびサブピクチャのサブピクチャ位置を備えるシーケンス・パラメータ・セット(SPS)とを受信するステップと、デコーダのプロセッサによって、SPSを解析してサブピクチャサイズおよびサブピクチャ位置を取得するステップと、プロセッサによって、サブピクチャサイズおよびサブピクチャ位置に基づいてサブピクチャをデコードしてビデオシーケンスを作り出すステップと、プロセッサによって、表示のためにビデオシーケンスを転送するステップと、を備える方法を含む。タイルおよびサブピクチャはピクチャより小さいので、いくつかのシステムは、タイリング情報およびサブピクチャ情報をピクチャ・パラメータ・セット(PPS)内に含む。しかしながら、サブピクチャは、関心領域(ROI)の適用およびサブピクチャを基にしたアクセス方式をサポートするために使用され得る。これらの適用はピクチャごとに変化しない。開示される例は、PPSの代わりにSPS内にサブピクチャのためのレイアウト情報を含む。サブピクチャ・レイアウト情報は、サブピクチャ位置およびサブピクチャサイズを含む。サブピクチャ位置は、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセットである。サブピクチャサイズは、ルマサンプルにおいて測定されるようなサブピクチャの高さおよび幅である。ビデオシーケンスは、単一のSPS(またはビデオセグメントごとに1つ)を含んでもよく、ピクチャごとに1つのPPSと同数のものを含んでもよい。SPS内にサブピクチャのためのレイアウト情報を配置することは、レイアウトが、各PPSについて冗長にシグナリングされるのではなく、シーケンス／セグメントについて1回のみシグナリングされることを保証する。従って、SPS内でサブピクチャ・レイアウトをシグナリングすることは、コーディング効率を増加させ、従って、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。また、いくつかのシステムは、デコーダによって導出されるサブピクチャ情報を有する。サブピクチャ情報をシグナリングすることは、損失したパケットの場合におけるエラーの確率を低減し、サブピクチャを抽出することに関して追加の機能をサポートする。従って、SPS内でサブピクチャ・レイアウトをシグナリングすることは、エンコーダおよび／またはデコーダの機能を改善する。

【0004】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、サブピクチャが時間動き制約サブピクチャであること、および、サブピクチャサイズおよびサブピクチャ位置が時間動き制約サブピクチャのレイアウトを示すことを提供する。

【0005】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、プロセッサによって、サブピクチャサイズに基づいてディスプレイのサイズに対するサブピクチャのサイズを決定することをさらに備えることを提供する。

【0006】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、プロセッサによって、サブピクチャ位置に基づいてディスプレイに対するサブピクチャの位置を決定することをさらに備えることを提供する。

【0007】

10

20

30

40

50

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、サブピクチャ位置が、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセット距離を含むことを提供する。

【0008】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、サブピクチャサイズが、ルマサンプルにおけるサブピクチャ高さおよびルマサンプルにおけるサブピクチャ幅を含むことを提供する。

【0009】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、SPSが、ピクチャから区分された各サブピクチャについてのサブピクチャ識別子(ID)をさらに備えることを提供する。

10

【0010】

実施形態では、開示は、エンコーダ内に実装される方法であって、エンコーダのプロセッサによって、ピクチャから区分されたサブピクチャをビットストリームへとエンコードするステップと、プロセッサによって、サブピクチャのサブピクチャサイズおよびサブピクチャ位置をビットストリーム内のSPSへとエンコードするステップと、エンコーダのメモリに、デコーダに向けた通信のためにビットストリームを記憶するステップと、を備える方法を含む。タイルおよびサブピクチャはピクチャより小さいので、いくつかのシステムは、タイリング情報およびサブピクチャ情報をピクチャ・パラメータ・セット(PPS)内に含む。しかしながら、サブピクチャは、関心領域(ROI)の適用およびサブピクチャを基にしたアクセス方式をサポートするために使用され得る。これらの適用はピクチャごとに変化しない。開示される例は、PPSの代わりにSPS内にサブピクチャのためのレイアウト情報を含む。サブピクチャ・レイアウト情報は、サブピクチャ位置およびサブピクチャサイズを含む。サブピクチャ位置は、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセットである。サブピクチャサイズは、ルマサンプルにおいて測定されるようなサブピクチャの高さおよび幅である。ビデオシーケンスは、単一のSPS(またはビデオセグメントごとに1つ)を含んでもよく、ピクチャごとに1つのPPSと同数のものを含んでもよい。SPS内にサブピクチャのためのレイアウト情報を配置することは、レイアウトが、各PPSについて冗長にシグナリングされるのではなく、シーケンス／セグメントについて1回のみシグナリングされることを保証する。従って、SPS内でサブピクチャ・レイアウトをシグナリングすることは、コーディング効率を増加させ、従って、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。また、いくつかのシステムは、デコーダによって導出されるサブピクチャ情報を有する。サブピクチャ情報をシグナリングすることは、損失したパケットの場合におけるエラーの確率を低減し、サブピクチャを抽出することに関して追加の機能をサポートする。従って、SPS内でサブピクチャ・レイアウトをシグナリングすることは、エンコーダおよび／またはデコーダの機能を改善する。

20

30

【0011】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、プロセッサによって、サブピクチャが時間動き制約サブピクチャであることを示すために、フラグをSPS内にエンコードすることをさらに備えることを提供する。

40

【0012】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、サブピクチャサイズおよびサブピクチャ位置が時間動き制約サブピクチャのレイアウトを示すことを提供する。

【0013】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、サブピクチャ位置が、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセット距離を含むことを提供する。

【0014】

50

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、サブピクチャサイズが、ルマサンプルにおけるサブピクチャ高さおよびルマサンプルにおけるサブピクチャ幅を含むことを提供する。

【0015】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、プロセッサによって、ピクチャから区分されたサブピクチャの各々についてのサブピクチャIDをSPSへとエンコードすることをさらに備えることを提供する。

【0016】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、プロセッサによって、ピクチャから区分されたサブピクチャの数をSPS内にエンコードすることをさらに備えることを提供する。

10

【0017】

実施形態では、開示は、プロセッサ、メモリ、プロセッサに結合された受信機、およびプロセッサに結合された送信機を備えるビデオコーディングデバイスを含み、プロセッサ、メモリ、受信機、および送信機は、先行する態様のいずれかの方法を実行するように構成される。

【0018】

実施形態では、開示は、ビデオコーディングデバイスによる使用のためのコンピュータプログラム製品を備える非一時的コンピュータ可読媒体を含み、コンピュータプログラム製品は、プロセッサによって実行されるときビデオコーディングデバイスに先行する態様のいずれかの方法を実行させるような、非一時的コンピュータ可読媒体に記憶されたコンピュータ実行可能命令を備える。

20

【0019】

実施形態では、開示は、ピクチャから区分されたサブピクチャを備えるビットストリームと、サブピクチャのサブピクチャサイズおよびサブピクチャのサブピクチャ位置を備えるSPSとを受信するための受信手段と、SPSを解析してサブピクチャサイズおよびサブピクチャ位置を取得するための解析手段と、サブピクチャサイズおよびサブピクチャ位置に基づいてサブピクチャをデコードしてビデオシーケンスを作り出すためのデコード手段と、表示のためにビデオシーケンスを転送するための転送手段と、を備えるデコーダを含む。

【0020】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、デコーダが先行する態様のいずれかの方法を実行するようにさらに構成されることを提供する。

30

【0021】

実施形態では、開示は、ピクチャから区分されたサブピクチャをビットストリームへとエンコードし、サブピクチャのサブピクチャサイズおよびサブピクチャ位置をビットストリーム内のSPSへとエンコードするためのエンコード手段と、デコーダに向けた通信のためにビットストリームを記憶するための記憶手段と、を備えるエンコーダを含む。

【0022】

任意選択で、先行する態様のいずれかにおいて、態様の別の実装は、エンコーダが先行する態様のいずれかの方法を実行するようにさらに構成されることを提供する。

40

【0023】

明確さの目的のために、前述の実施形態のいずれか1つが、本開示の範囲内で新しい実施形態を作り出すために、他の前述の実施形態のいずれか1つまたは複数と組み合わせられ得る。

【0024】

これらおよび他の特徴は、添付の図面および請求項と併せて用いられる以下の詳細な説明からより明確に理解されるであろう。

【0025】

この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用いられる以下の簡単な説明への参照がここで行われ、同様の参照符号は同様の部分を表現する。

50

【図面の簡単な説明】

【0026】

【図1】ビデオ信号をコーディングする一例の方法のフローチャートである。

【図2】ビデオコーディングのための一例のコーディングおよびデコード(コーデック)システムの概略図である。

【図3】一例のビデオエンコーダを例示する概略図である。

【図4】一例のビデオデコーダを例示する概略図である。

【図5】一例のビットストリームおよびそのビットストリームから抽出されるサブビットストリームを例示する概略図である。

【図6】サブピクチャへと区分される一例のピクチャを例示する概略図である。

10

【図7】スライスをサブピクチャ・レイアウトに関連させるための一例の機構を例示する概略図である。

【図8】サブピクチャへと区分される別の例のピクチャを例示する概略図である。

【図9】一例のビデオコーディングデバイスの概略図である。

【図10】サブピクチャの抽出をサポートするためにピクチャのビットストリームにおいてサブピクチャ・レイアウトをエンコードする一例の方法のフローチャートである。

【図11】シグナリングされたサブピクチャ・レイアウトに基づいてサブピクチャのビットストリームをデコードする一例の方法のフローチャートである。

【図12】ビットストリームを介してサブピクチャ・レイアウトをシグナリングするための一例のシステムの概略図である。

20

【発明を実施するための形態】

【0027】

1つまたは複数の実施形態の例示的な実装が以下で提供されるが、開示されるシステムおよび／または方法は、現在知られているか、または存在しているかにかかわらず、任意の数の技法を使用して実装され得ることが始めに理解されるべきである。開示は、ここで例示され説明される典型的な設計および実装を含む、以下で例示される例示的な実装、図、および技法に、いかなる場合も限定されるべきでなく、しかし、それらの等価物の完全な範囲とともに添付の請求項の範囲内で修正され得る。

【0028】

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)、コーディングされたビデオシーケンス(CVS)、ジョイント・ビデオ・エキスパート・チーム(JVET)、動き制約タイルセット(MCTS)、最大伝送単位(MTU)、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)、バーサタイル・ビデオ・コーディング(VVC)、およびワーキング・ドラフト(WD)のような様々な頭字語がここで利用される。

30

【0029】

データの最小損失とともにビデオファイルのサイズを低減するために、多くのビデオ圧縮技法が利用されることが可能である。たとえば、ビデオ圧縮技法は、空間(たとえば、イントラピクチャ)予測および／または時間(たとえば、インターピクチャ)予測を実行して、ビデオシーケンスにおけるデータ冗長性を低減するまたは取り除くことを含むことができる。ブロックを基にしたビデオコーディングのために、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、これは、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)スライス内のビデオブロックは、同じピクチャ内の近隣ブロック内の参照サンプルに関して空間予測を使用してコーディングされる。ピクチャのインターコーディングされた単方向予測(P)または双方向予測(B)スライス内のビデオブロックは、同じピクチャ内の近隣ブロック内の参照サンプルに関して空間予測、または他の参照ピクチャ内の参照サンプルに関して時間予測を利用することによ

40

50

ってコーディングされ得る。ピクチャはフレームおよび／または画像と呼ばれることがあり、参照ピクチャは参照フレームおよび／または参照画像と呼ばれることがある。空間予測または時間予測は、画像ブロックを表現する予測ブロックをもたらす。残差データは、元の画像ブロックと予測ブロックとの間のピクセル差を表現する。従って、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を示す残差データに従ってエンコードされる。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エンコードされる。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得る。これらは、量子化され得る残差変換係数をもたらす。量子化された変換係数は最初に、2次元アレイに配置され得る。量子化された変換係数が、変換係数の1次元ベクトルを作成するために走査され得る。エントロピーコーディングは、よりいっそうの圧縮を達成するために適用され得る。そのようなビデオ圧縮技法は、以下でより詳細に論じられる。

【0030】

エンコードされたビデオが正確にデコードされることが可能であることを保証するために、対応するビデオコーディング規格に従って、ビデオがエンコードされデコードされる。ビデオコーディング規格は、国際電気通信連合(ITU)標準化部門(ITU-T)H.261、国際標準化機構／国際電気標準会議(ISO/IEC)モーション・ピクチャ・エキスパート・グループ(MPEG)-1 Part 2、ITU-T H.262またはISO/IEC MPEG-2 Part 2、ITU-T H.263、ISO/IEC MPEG-4 Part 2、ITU-T H.264またはISO/IEC MPEG-4 Part 10としても知られているアドバンスド・ビデオ・コーディング(AVC)、およびITU-T H.265またはMPEG-H Part 2としても知られてい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・デプス(MVC+D)、ならびに3次元(3D)AVC(3D-AVC)のような拡張を含む。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)のような拡張を含む。ITU-TおよびISO/IECのジョイント・ビデオ・エキスパート・チーム(JVET)は、バーサタイル・ビデオ・コーディング(VVC)と呼ばれるビデオコーディング規格を開発することを開始した。VVCはワーキング・ドラフト(WD)に含まれ、これはJVET-L1001-v9を含む。

【0031】

ビデオ画像をコーディングするために、画像はまず区分され、区分はビットストリームへとコーディングされる。様々なピクチャ区分方式が利用可能である。たとえば、画像は、通常のスライス、従属スライス、タイルへと、および／または、波面並列処理(WPP)に従って区分されることが可能である。簡潔さのために、ビデオコーディングのためにCTBのグループへとスライスを区分するとき、通常のスライス、従属スライス、タイル、WPP、およびそれらの組み合わせのみが使用されることが可能であるように、HEVCはエンコーダを制限する。そのような区分は、最大伝送単位(MTU)サイズのマッチング、並列処理、低減されたエンドツーエンド遅延をサポートするために適用されることが可能である。MTUは、単一のパケットにおいて送信されることが可能であるデータの最大量を表記する。パケットペイロードがMTUを超えるならば、そのペイロードは断片化と呼ばれる処理を通じて2つのパケットへと分割される。

【0032】

単にスライスとも呼ばれる通常のスライスは、ループフィルタリング動作によるいくつかの相互依存性にもかかわらず、同じピクチャ内の他の通常のスライスとは独立に再構築されることが可能である画像の区分された部分である。各々の通常のスライスは、送信のためにそれ自身のネットワーク抽象化レイヤ(NAL)ユニットにカプセル化される。さらに、ピクチャ内予測(イントラサンプル予測、動き情報予測、コーディングモード予測)およびスライス境界にわたるエントロピーコーディング依存性は、独立した再構築をサポートするために無効にされ得る。そのような独立した再構築は並列化をサポートする。たとえば、通常のスライスを基にした並列化は、最小のプロセッサ間またはコア間通信を利用す

10

20

30

40

50

る。しかしながら、各々の通常のスライスが独立であるので、各スライスは別々のスライスヘッダと関連付けられる。通常のスライスの使用は、各スライスについてのスライスヘッダのビットコストにより、およびスライス境界にわたる予測の欠如により、かなりのコーディングオーバーヘッドを招くことがある。さらに、通常のスライスは、MTUサイズの要件についてのマッチングをサポートするために利用され得る。具体的には、通常のスライスは別々のNALユニットにカプセル化され、独立にコーディングされることが可能であるので、各々の通常のスライスは、複数のパケットへとスライスを分けることを避けるために、MTU方式におけるMTUより小さいべきである。それゆえ、並列化の目標およびMTUサイズのマッチングの目標は、ピクチャにおけるスライスレイアウトへの矛盾する要求を課し得る。

【0033】

従属スライスは通常のスライスに類似するが、短縮されたスライスヘッダを有し、ピクチャ内予測を破壊することなく画像ツリーブロック境界の区分を可能にする。従って、従属スライスは、通常のスライスが複数のNALユニットへと断片化されることを可能にし、これは、通常のスライス全体のエンコードが完了する前に通常のスライスの一部が送り出されることを可能にすることによって、低減されたエンドツーエンド遅延を提供する。

【0034】

タイルは、タイルの列と行を作り出す水平および垂直の境界によって作り出される画像の区分された部分である。タイルはラスタ走査順(右から左および上から下)でコーディングされ得る。CTBの走査順はタイル内で局所的である。従って、第1のタイル内のCTBは、次のタイル内のCTBに進む前に、ラスタ走査順でコーディングされる。通常のスライスと類似して、タイルは、エントロピーデコードの依存性とともによりピクチャ内予測の依存性を破壊する。しかしながら、タイルは個々のNALユニットに含まれないことがあり、従って、タイルはMTUサイズのマッチングのために使用されないことがある。各タイルは、1つのプロセッサ／コアによって処理されることが可能であり、近隣のタイルをデコードする処理ユニット間のピクチャ内予測のために利用されるプロセッサ間／コア間通信は、共有されるスライスヘッダを搬送すること(隣接するタイルが同じスライス内にあるとき)、および再構築されたサンプルとメタデータのループフィルタリング関連の共有を実行することに限定され得る。1つより多くのタイルがスライスに含まれるとき、スライス内の最初のエントリ・ポイント・オフセット以外の各タイルについてのエントリ・ポイント・バイト・オフセットが、スライスヘッダ内でシグナリングされ得る。各スライスおよびタイルについて、以下の条件、1)スライス内のすべてのコーディングされたツリーブロックが同じタイルに属する、および2)タイル内のすべてのコーディングされたツリーブロックが同じスライスに属する、のうちの少なくとも1つが満たされるべきである。

【0035】

WPPでは、画像はCTBの単一の行へと区分される。エントロピーデコードおよび予測機構は、他の行におけるCTBからのデータを使用し得る。並列処理は、CTBの行の並列デコードを通じて可能にされる。たとえば、現在の行は、先行する行と並列にデコードされ得る。しかしながら、現在の行のデコードは、2つのCTBによる先行する行のデコードプロセスより遅延する。この遅延は、現在の行における現在のCTBの上にあるCTBおよび右上にあるCTBに関連するデータが、現在のCTBがコーディングされる前に利用可能であることを保証する。この手法は、グラフィカルに表現されるとき波面として現れる。このずらされた開始は、画像が含むCTBの行の数と同じ数までのプロセッサ／コアを用いた並列化を可能にする。ピクチャ内の近隣のツリーブロックの行の間のピクチャ内予測が許容されるので、ピクチャ内予測を有効にするためのプロセッサ間／コア間通信はかなりであることがある。WPPの区分はNALユニットサイズを考慮する。従って、WPPはMTUサイズのマッチングをサポートしない。しかしながら、要求に応じてMTUサイズのマッチングを実装するために、あるコーディングオーバーヘッドを有してWPPと併せて通常のスライスが使用されることが可能である。

【0036】

タイルは動き制約タイルセットも含み得る。動き制約タイルセット(MCTS)は、関連付けられた動きベクトルが、MCTS内部の完全サンプル位置と、補間のためにMCTS内部の完全サンプル位置のみを要求する分数サンプル位置とを指し示すように制限されるように設計されたタイルセットである。さらに、MCTS外部のブロックから導出される時間動きベクトル予測についての動きベクトル候補の使用は許容されない。このようにして、各MCTSは、MCTSに含まれないタイルが存在することなく、独立にデコードされ得る。時間MCTS補足強化情報(SEI)メッセージは、ビットストリームにおけるMCTSの存在を示し、MCTSをシグナリングするために使用され得る。MCTS SEIメッセージは、MCTSセットについて準拠するビットストリームを生成するために、MCTSサブビットストリーム抽出(SEIメッセージのセマンティクスの部分として指定される)において使用されることが可能である補足情報を提供する。情報はいくつかの抽出情報セットを含み、各々が、MCTSセットの数を定義し、MCTSサブビットストリーム抽出プロセスの間に使用されるべき置換ビデオパラメータセット(VPS)、シーケンス・パラメータ・セット(SPS)、およびピクチャ・パラメータ・セット(PPS)のロー・バイト・シーケンス・ペイロード(RBSP)バイトを含む。MCTSサブビットストリーム抽出プロセスに従ってサブビットストリームを抽出するとき、パラメータセット(VPS、SPS、およびPPS)は、書き直され、または置換されてもよく、スライスアドレス関連のシンタックス要素(first_slice_segment_in_pic_flagおよびslice_segment_addressを含む)のうちの1つまたはすべてが、抽出されたサブビットストリームにおいて異なる値を利用し得るので、スライスヘッダは更新されてもよい。

【0037】

ピクチャは、また、1つまたは複数のサブピクチャへと区分され得る。サブピクチャは、0に等しいtile_group_addressを有するタイルグループで開始する、タイルグループ／スライスの長方形セットである。各サブピクチャは、別々のPPSを参照することが可能であり、従って、別々のタイル区分を有することが可能である。サブピクチャは、デコードプロセスにおいてピクチャのように扱われ得る。現在のサブピクチャをデコードするための参照サブピクチャは、デコードされたピクチャバッファにおいて参照ピクチャから現在のサブピクチャと同じに位置付けられるエリアを抽出することによって生成される。抽出されたエリアは、デコードされたサブピクチャとして扱われる。同じサイズのサブピクチャとピクチャ内の同じ位置との間で、インター予測が行われ得る。スライスとしても知られているタイルグループは、ピクチャまたはサブピクチャ内の関連するタイルのシーケンスである。ピクチャ内のサブピクチャの位置を決定するために、いくつかの項目が導出されることが可能である。たとえば、各々の現在のサブピクチャは、ピクチャ境界内に現在のサブピクチャを含むために十分大きいピクチャ内で、CTUラスタ走査順で次の占有されていない位置に配置され得る。

【0038】

さらに、ピクチャ区分は、ピクチャレベルタイルおよびシーケンスレベルタイルに基づき得る。シーケンスレベルタイルは、MCTSの機能を含んでもよく、サブピクチャとして実装され得る。たとえば、ピクチャレベルタイルは、ピクチャ内の特定のタイル列および特定のタイル行内のコーディング・ツリー・ブロックの長方形領域として定義され得る。シーケンスレベルタイルは、異なるフレームに含まれるコーディング・ツリー・ブロックの長方形領域のセットとして定義されてもよく、各長方形領域はさらに1つまたは複数のピクチャレベルタイルを備え、コーディング・ツリー・ブロックの長方形領域のセットは類似する長方形領域の任意の他のセットから独立にデコード可能である。シーケンスレベルタイルグループセット(STGPS)は、そのようなシーケンスレベルタイルのグループである。STGPSは、NALユニットヘッダ内に関連付けられた識別子(ID)を有する非ビデオコーディングレイヤ(VCL)NALユニット内でシグナリングされ得る。

【0039】

先行するサブピクチャを基にした区分方式は、ある課題と関連付けられ得る。たとえば、サブピクチャが有効にされるとき、サブピクチャ内のタイリング(タイルへのサブピクチャの区分)が、並列処理をサポートするために使用されることが可能である。並列処理の目

10

20

30

40

50

的のためのサブピクチャのタイル区分は、ピクチャごとに変化することが可能であり(たとえば、並列処理負荷のバランスをとる目的のために)、従って、ピクチャレベルで(たとえば、PPSにおいて)管理されてもよい。しかしながら、サブピクチャ区分(サブピクチャへのピクチャの区分)は、関心領域(ROI)およびサブピクチャを基にしたピクチャアクセスをサポートするために利用され得る。そのような場合、PPSにおけるサブピクチャまたはMCTSのシグナリングは効率的でない。

【0040】

別の例では、ピクチャ内の任意のサブピクチャが時間動き制約サブピクチャとしてコーディングされるとき、ピクチャ内のすべてのサブピクチャが、時間動き制約サブピクチャとしてコーディングされ得る。そのようなピクチャ区分は限定的であり得る。たとえば、時間動き制約サブピクチャとしてサブピクチャをコーディングすることは、追加の機能と引き換えにコーディング効率を低減し得る。しかしながら、関心領域を基にした適用では、通常はサブピクチャのうちの1つまたは数個のみが、時間動き制約サブピクチャを基にした機能を使用する。従って、残りのサブピクチャは、何も実際上の利益を提供することなく、低減されたコーディング効率を被る。

【0041】

別の例では、サブピクチャのサイズを指定するためのシンタックス要素は、ルマCTUサイズの単位で指定され得る。従って、サブピクチャの幅と高さの両方が、CtbSizeYの整数倍であるべきである。サブピクチャの幅と高さを指定するこの機構は、様々な問題をもたらし得る。たとえば、サブピクチャ区分は、CtbSizeYの整数倍であるピクチャ幅および／またはピクチャ高さを有するピクチャにのみ適用可能である。これは、サブピクチャの区分を、CTbSizeYの整数倍でない寸法を含むピクチャについて利用不可能として表現する。ピクチャ寸法がCtbSizeYの整数倍でないとき、サブピクチャ区分がピクチャの幅および／または高さに適用されるならば、最も右のサブピクチャおよび最も下のサブピクチャについてのルマサンプル単位でのサブピクチャ幅および／またはサブピクチャ高さの導出は正しくないであろう。そのような正しくない導出は、いくつかのコーディングツールにおいて誤った結果を引き起こす。

【0042】

別の例では、ピクチャ内のサブピクチャの位置はシグナリングされなくてもよい。代わりに、位置は以下の規則を使用して導出される。現在のサブピクチャは、ピクチャ境界内にサブピクチャを含むために十分大きいピクチャ内で、CTUラスタ走査順で次のそのような占有されていない位置に配置される。そのようなやり方でサブピクチャ位置を導出することは、いくつかの場合においてエラーを引き起こし得る。たとえば、サブピクチャが送信において損失されるならば、他のサブピクチャの位置が正しくなく導出され、デコードされたサンプルが誤った位置に配置される。サブピクチャが間違った順序で到着するとき、同じ課題が当てはまる。

【0043】

別の例では、サブピクチャをデコードすることは、参照ピクチャ内で同じに位置付けられるサブピクチャの抽出を要求し得る。これは、プロセッサおよびメモリリソースの使用に関して、追加の複雑さおよび結果として生じる負担を強いることがある。

【0044】

別の例では、サブピクチャが時間動き制約サブピクチャとして設計されるとき、サブピクチャ境界を横切るループフィルタが無効にされる。これは、タイル境界を横切るループフィルタが有効にされるかどうかにかかわらず起こる。そのような制約は、制限しすぎることがあり、複数のサブピクチャを利用するビデオピクチャについて視覚的なアーティファクトをもたらすことがある。

【0045】

別の例では、SPS、STGPS、PPS、およびタイルグループヘッダの間の関係は、以下の通りである。STGPSはSPSを参照し、PPSはSTGPSを参照し、タイルグループヘッダ／スライスヘッダはPPSを参照する。しかしながら、STGPSおよびPPSは、PPSがSTGPSを参

10

20

30

40

50

照するのではなく、直交すべきである。先行する構成は、また、同じピクチャのすべてのタイルグループが同じPPSを参照することを許容しないことがある。

【0046】

別の例では、各STGPSは、サブピクチャの4つの側についてIDを含み得る。そのようなIDが使用されて同じ境界を共有するサブピクチャを識別し、それによって、それらの相対的な空間関係が定義されることが可能である。しかしながら、そのような情報は、いくつかの場合において、シーケンスレベルタイルグループセットについて位置およびサイズ情報を導出するために十分でないことがある。他の場合には、位置およびサイズ情報をシグナリングすることは冗長であり得る。

【0047】

別の例では、STGPS IDは、8ビットを使用してVCL NALユニットのNALユニットヘッダ内でシグナリングされ得る。これはサブピクチャ抽出を助け得る。そのようなシグナリングは、NALユニットヘッダの長さを不必要に増加させ得る。別の問題は、シーケンスレベルタイルグループセットが重複を防ぐように制約されない限り、1つのタイルグループが複数のシーケンスレベルタイルグループセットと関連付けられ得ることである。

【0048】

上述の課題のうちの1つまたは複数に対処するために、様々な機構がここで開示される。第1の例では、サブピクチャのためのレイアウト情報が、PPSの代わりにSPS内に含まれる。サブピクチャ・レイアウト情報は、サブピクチャ位置およびサブピクチャサイズを含む。サブピクチャ位置は、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセットである。サブピクチャサイズは、ルマサンプルにおいて測定されるようなサブピクチャの高さと幅である。上で注記されたように、タイルはピクチャごとに変化し得るので、いくつかのシステムはPPS内にタイリング情報を含む。しかしながら、ROI適用およびサブピクチャを基にしたアクセスをサポートするために、サブピクチャが使用され得る。これらの機能は、ピクチャごとに変化しない。さらに、ビデオシーケンスは、単一のSPS(またはビデオセグメントごとに1つ)を含んでもよく、ピクチャごとに1つのPPSと同数のものを含んでもよい。SPS内にサブピクチャのためのレイアウト情報を配置することは、レイアウトが、各PPSについて冗長にシグナリングされるのではなく、シーケンス／セグメントについて1回のみシグナリングされることを保証する。従って、SPS内でサブピクチャ・レイアウトをシグナリングすることは、コーディング効率を増加させ、従って、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。また、いくつかのシステムは、デコーダによって導出されるサブピクチャ情報を有する。サブピクチャ情報をシグナリングすることは、損失したパケットの場合におけるエラーの確率を低減し、サブピクチャを抽出することに関して追加の機能をサポートする。従って、SPS内でサブピクチャ・レイアウトをシグナリングすることは、エンコーダおよび／またはデコーダの機能を改善する。

【0049】

第2の例では、サブピクチャの幅およびサブピクチャの高さは、CTUサイズの倍数に制約される。しかしながら、これらの制約は、サブピクチャがピクチャの右の境界またはピクチャの下境界にそれぞれ配置されるとき、取り除かれる。上で注記されたように、いくつかのビデオシステムは、CTUサイズの倍数である高さおよび幅を含むようにサブピクチャを限定し得る。これは、サブピクチャが多くピクチャレイアウトを用いて正しく動作することを妨げる。下と右のサブピクチャが、CTUサイズの倍数でない高さおよび幅をそれぞれ含むことを可能にすることによって、サブピクチャは、デコードエラーを引き起こすことなく任意のピクチャとともに使用され得る。これは、エンコーダおよびデコーダの機能を増加させることをもたらす。さらに、増加した機能は、エンコーダがより効率的にピクチャをコーディングすることを可能にし、これは、エンコーダおよびデコーダにおけるネットワークリソース、メモリリソース、および／または処理リソースの使用を低減する。

【0050】

第3の例では、サブピクチャは、間隙または重複なしでピクチャを包含するように制約

10

20

30

40

50

される。上で注記されたように、いくつかのビデオコーディングシステムは、サブピクチャが間隙および重複を含むことを許容する。これは、タイルグループ／スライスが複数のサブピクチャと関連付けられる可能性を作り出す。これがエンコーダにおいて許容されるならば、デコーダは、そのデコード方式がまれに使用されるときでも、そのようなコーディング方式をサポートするように構築されなければならない。サブピクチャの間隙および重複を許容しないことによって、サブピクチャのサイズおよび位置を決定するときにデコーダが潜在的な間隙および重複を考慮すること要求されないため、デコーダの複雑さが減らされることが可能である。さらに、サブピクチャの間隙および重複を許容しないことは、ビデオシーケンスについてエンコードを選択するときにエンコーダが間隙と重複の場合を考慮するのを省略できるので、エンコーダにおけるレート歪み最適化(RDO)プロセスの複雑さを低減する。従って、間隙および重複を回避することは、エンコーダおよびデコーダにおけるメモリリソースおよび／または処理リソースの使用を低減し得る。

【0051】

第4の例では、サブピクチャが時間動き制約サブピクチャであるときを示すために、フラグが、SPS内でシグナリングされることが可能である。上で注記されたように、いくつかのシステムは、すべてのサブピクチャを時間動き制約サブピクチャであるようにまとめて設定しても、または、時間動き制約サブピクチャの使用を完全に許容しなくてもよい。そのような時間動き制約サブピクチャは、減らされたコーディング効率という犠牲を伴って、独立抽出機能を提供する。しかしながら、関心領域を基にした適用では、関心領域は独立抽出のためにコーディングされるべきであり、一方、関心領域の外側の領域はそのような機能を必要としない。従って、残りのサブピクチャは、何も実際上の利益を提供することなく、低減されたコーディング効率を被る。従って、フラグは、独立抽出が望まれないときに、増加したコーディング効率のために、独立抽出機能を提供する時間動き制約サブピクチャと非動き制約サブピクチャとの混合を可能にする。従って、フラグは、増加した機能および／または増加したコーディング効率を可能にし、これは、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。

【0052】

第5の例では、サブピクチャIDの完全なセットは、SPS内でシグナリングされ、スライスヘッダは、対応するスライスを含むサブピクチャを示すサブピクチャIDを含む。上で注記されたように、いくつかのシステムは、他のサブピクチャに対するサブピクチャ位置をシグナリングする。これは、サブピクチャが損失され、または別々に抽出されるならば、課題を引き起こす。各サブピクチャをIDによって指定することによって、サブピクチャは、他のサブピクチャを参照することなく配置され、サイズ決めされることが可能である。そして、これは、サブピクチャのいくつかを抽出するのみで、他のサブピクチャを送信することを回避する適用とともに誤り訂正をサポートする。すべてのサブピクチャIDの完全なリストが、関連するサイズ情報とともにSPS内で送信されることが可能である。各スライスヘッダは、対応するスライスを含むサブピクチャを示すサブピクチャIDを含み得る。このようにして、サブピクチャおよび対応するスライスは、他のサブピクチャを参照することなく抽出され配置されることが可能である。従って、サブピクチャIDは、増加した機能および／または増加したコーディング効率をサポートし、これは、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。

【0053】

第6の例では、レベルが各サブピクチャについてシグナリングされる。いくつかのビデオコーディングシステムでは、レベルがピクチャについてシグナリングされる。レベルは、ピクチャをデコードするために必要とされるハードウェアリソースを示す。上で注記されたように、異なるサブピクチャは、いくつかの場合において異なる機能を有することがあり、従って、コーディングプロセスの間に異なって扱われることがある。それゆえ、ピクチャを基にしたレベルは、いくつかのサブピクチャをデコードするために有用でないこ

とがある。従って、本開示は、各サブピクチャについてレベルを含む。このようにして、各サブピクチャは、より複雑でない機構に従ってコーディングされるサブピクチャについてデコード要件を高すぎに設定することによってデコーダに不必要に負担をかけることなく、他のサブピクチャとは独立にコーディングされることが可能である。シグナリングされるサブピクチャレベル情報は、増加した機能および／または増加したコーディング効率をサポートし、これは、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。

【0054】

図1は、ビデオ信号をコーディングする一例の動作方法100のフローチャートである。具体的には、ビデオ信号はエンコーダにおいてエンコードされる。エンコードプロセスは、ビデオファイルサイズを低減するために様々な機構を利用することによってビデオ信号を圧縮する。より小さいファイルサイズは、圧縮されたビデオファイルがユーザに向かって送信されることを可能にし、一方、関連付けられた帯域幅オーバーヘッドを低減する。デコーダは次いで、圧縮されたビデオファイルをデコードして、エンドユーザへの表示のために元のビデオ信号を再構築する。デコードプロセスは一般に、デコーダがビデオ信号を首尾一貫して再構築することを可能にするために、エンコードプロセスを鏡写しにしたものである。

【0055】

ステップ101において、ビデオ信号がエンコーダに入力される。たとえば、ビデオ信号は、メモリに記憶された圧縮されていないビデオファイルであり得る。別の例として、ビデオファイルは、ビデオカメラのようなビデオキャプチャデバイスによって捕捉され、ビデオのライブストリーミングをサポートするためにエンコードされ得る。ビデオファイルは、オーディオ成分とビデオ成分の両方を含み得る。ビデオ成分は、順番に見られるとき動きの視覚的な印象を与える一連の画像フレームを含む。フレームは、ここでルマ成分(またはルマサンプル)と呼ばれる光、およびクロマ成分(またはカラーサンプル)と呼ばれる色に関して表現されるピクセルを含む。いくつかの例では、フレームは、3次元で見ることをサポートするために深度値も含み得る。

【0056】

ステップ103において、ビデオはブロックへと区分される。区分は、各フレーム内のピクセルを、圧縮のために正方形および／または長方形のブロックへと細分することを含む。たとえば、高効率ビデオコーディング(HEVC)(H.265およびMPEG-H Part 2としても知られている)では、フレームは、まずコーディング・ツリー・ユニット(CTU)へと分割されることが可能であり、これはあらかじめ定義されたサイズ(たとえば、64ピクセル×64ピクセル)のブロックである。CTUはルマおよびクロマサンプルの両方を含む。コーディング・ツリーは、CTUをブロックへと分割し、次いで、さらなるエンコードをサポートする構成が達成されるまでブロックを再帰的に細分するために利用され得る。たとえば、フレームのルマ成分は、個々のブロックが比較的均質な照明値を含むまで細分され得る。さらに、フレームのクロマ成分は、個々のブロックが比較的均質な色値を含むまで細分され得る。従って、区分機構はビデオフレームの内容に依存して変化する。

【0057】

ステップ105において、ステップ103において区分された画像ブロックを圧縮するために様々な圧縮機構が利用される。たとえば、インター予測および／またはイントラ予測が利用され得る。インター予測は、共通のシーンにおける物体は連続するフレーム内に出現する傾向があるという事実を利用するように設計される。従って、参照フレーム内の物体を描写するブロックは、隣接フレーム内で繰り返し記述される必要はない。具体的には、テーブルのような物体は、複数のフレーム上で一定の位置にとどまり得る。従って、テーブルは一度記述され、隣接フレームは参照フレームへ戻って参照することができる。複数のフレーム上で物体を照合するために、パターンマッチング機構が利用され得る。さらに、動いている物体は、たとえば物体の動きまたはカメラの動きにより、複数のフレームにわたって表現され得る。特定の例として、ビデオは、複数のフレーム上で画面にわたって

10

20

30

40

50

動く自動車を表し得る。そのような動きを記述するために動きベクトルが利用されることが可能である。動きベクトルは、フレーム内の物体の座標から参照フレーム内の物体の座標へのオフセットを提供する2次元ベクトルである。それゆえ、インター予測は、参照フレーム内の対応するブロックからのオフセットを示す動きベクトルのセットとして、現在のフレーム内の画像ブロックをエンコードすることができる。

【0058】

イントラ予測は共通のフレーム内のブロックをエンコードする。イントラ予測は、ルマおよびクロマ成分がフレーム内で密集する傾向があるという事実を利用する。たとえば、木の一部分における緑の斑点は、類似する緑の斑点に隣接して配置される傾向がある。イントラ予測は、複数の方向性予測モード(たとえば、HEVCでは33個)、平面モード、および直流(DC)モードを利用する。方向性モードは、現在のブロックが対応する方向における近隣ブロックのサンプルと類似する／同じであることを示す。平面モードは、行／列(たとえば、平面)に沿った一連のブロックが行の端にある近隣ブロックに基づいて補間されることが可能であることを示す。平面モードは、事実上、値を変化させる際に比較的一定の勾配を利用することによって、行／列にわたる光／色の滑らかな遷移を示す。DCモードは、境界平滑化のために利用され、方向性予測モードの角度方向と関連付けられるすべての近隣ブロックのサンプルと関連付けられる平均値とブロックが類似する／同じであることを示す。従って、イントラ予測ブロックは、実際の値の代わりに様々な関係する予測モードの値として画像ブロックを表現することができる。さらに、インター予測ブロックは、実際の値の代わりに動きベクトルの値として画像ブロックを表現することができる。いずれの場合でも、予測ブロックは、いくつかの場合において、画像ブロックを厳密に表現しないことがある。あらゆる差分が残差ブロックに格納される。ファイルをさらに圧縮するために、残差ブロックに変換が適用され得る。

【0059】

ステップ107において、様々なフィルタリング技法が適用され得る。HEVCでは、フィルタはループ内フィルタリング方式に従って適用される。上で論じられたブロックを基にした予測は、デコーダにおけるブロック状の画像の生成をもたらす。さらに、ブロックを基にした予測方式は、ブロックをエンコードし、次いで、参照ブロックとしての後の使用のためにエンコードされたブロックを再構築し得る。ループ内フィルタリング方式は、ノイズ抑制フィルタ、デブロックングフィルタ、適応ループフィルタ、およびサンプル適応オフセット(SAO)フィルタをブロック／フレームに反復的に適用する。これらのフィルタは、エンコードされたファイルが正確に再構築されることが可能であるように、そのようなブロックングアーティファクトを軽減する。さらに、これらのフィルタは再構築された参照ブロックにおけるアーティファクトを軽減し、それによって、アーティファクトは、再構築された参照ブロックに基づいてエンコードされる後続のブロックにおいて追加のアーティファクトを作り出す見込みがより少ない。

【0060】

いったんビデオ信号が区分され、圧縮され、フィルタリングされると、ステップ109において、結果として生じるデータがビットストリーム内にエンコードされる。ビットストリームは、デコーダにおける適切なビデオ信号の再構築をサポートするために望まれるあらゆるシグナリングデータとともに上で論じられたデータを含む。たとえば、そのようなデータは、区分データ、予測データ、残差ブロック、およびコーディング命令をデコーダに提供する様々なフラグを含み得る。ビットストリームは、要求に応じたデコーダに向けた送信のためにメモリに記憶され得る。ビットストリームは、また、複数のデコーダに向けたブロードキャストおよび／またはマルチキャストであり得る。ビットストリームの生成は反復的なプロセスである。従って、ステップ101、103、105、107、および109は、多数のフレームおよびブロック上で連続的および／または同時に起こり得る。図1に表される順序は、明確さおよび議論の容易さのために提示され、ビデオコーディングプロセスを特定の順序に限定することは意図されない。

【0061】

ステップ111において、デコーダが、ビットストリームを受信し、デコードプロセスを開始する。具体的には、デコーダは、エントロピーデコード方式を利用して、ビットストリームを対応するシンタックスおよびビデオデータへと変換する。ステップ111において、デコーダが、ビットストリームからのシンタックスデータを利用して、フレームについての区分を決定する。区分は、ステップ103におけるブロック区分の結果と一致すべきである。ステップ111において利用されるようなエントロピーエンコード／デコードがここで説明される。エンコーダは、入力画像における値の空間的な配置に基づいて、いくつかの可能な選択からブロック区分方式を選択することのような、圧縮プロセスの間に多くの選択を行う。厳密な選択のシグナリングは、多数のピンを利用し得る。ここで使用されるように、ピンは、変数として扱われる二進値(たとえば、状況に依存して変化し得るビット値)である。エントロピーコーディングは、特定の場合について明らかに実行可能ではないあらゆる選択肢をエンコーダが廃棄することを可能にし、許容可能な選択肢のセットを残す。次いで、各々の許容可能な選択肢が符号語を割り当てられる。符号語の長さは、許容可能な選択肢の数に基づく(たとえば、2つの選択肢については1つのピン、3つから4つの選択肢については2つのピンなど)。エンコーダは次いで、選択された選択肢について符号語をエンコードする。符号語は、すべての可能な選択肢の潜在的に大きいセットからの選択を一意に示すのとは対照的に、許容可能な選択肢の小さいサブセットからの選択を一意に示すために望まれるぐらいの大きさであるので、この方式は符号語のサイズを低減する。デコーダは次いで、許容可能な選択肢のセットをエンコーダと類似する方式で決定することによって、選択をデコードする。許容可能な選択肢のセットを決定することによって、デコーダは、符号語を読み取り、エンコーダによって行われる選択を決定することができる。

【0062】

ステップ113において、デコーダがブロックデコードを実行する。具体的には、デコーダは、逆変換を利用して残差ブロックを生成する。次いで、デコーダは、残差ブロックおよび対応する予測ブロックを利用して、区分に従って画像ブロックを再構築する。予測ブロックは、ステップ105においてエンコーダにおいて生成されたようなイントラ予測ブロックとインター予測ブロックの両方を含み得る。再構築された画像ブロックは次いで、ステップ111において決定された区分データに従って、再構築されたビデオ信号のフレームへと配置される。ステップ113についてのシンタックスは、また、上で論じられたようなエントロピーコーディングを介してビットストリーム内でシグナリングされ得る。

【0063】

ステップ115において、エンコーダにおいて、ステップ107と類似する方式で、再構築されたビデオ信号のフレームにおいてフィルタリングが実行される。たとえば、ノイズ抑制フィルタ、デブロッキングフィルタ、適応ループフィルタ、およびSAOフィルタが、ブロッキングアーティファクトを取り除くためにフレームに適用され得る。いったんフレームがフィルタリングされると、ビデオ信号は、エンドユーザによって見るために、ステップ117においてディスプレイに出力されることが可能である。

【0064】

図2は、ビデオコーディングのための一例のコーディングおよびデコード(コーデック)システム200の概略図である。具体的には、コーデックシステム200は、動作方法100の実装をサポートするために機能を提供する。コーデックシステム200は、エンコーダとデコーダの両方において利用されるコンポーネントを描写するために一般化されている。コーデックシステム200は、動作方法100においてステップ101および103に関して論じられたようなビデオ信号を受信および区分し、これは区分されたビデオ信号201をもたらす。コーデックシステム200は次いで、方法100においてステップ105、107、および109に関して論じられたようなエンコーダとして動作するとき、区分されたビデオ信号201をコーディングされたビットストリームへと圧縮する。デコーダとして動作するとき、コーデックシステム200は、動作方法100においてステップ111、113、115、および117に関して論じられたようなビットストリームから出力ビデオ信号を生成する。コーデックシス

テム200は、汎用コーダ制御コンポーネント211、変換スケーリングおよび量子化コンポーネント213、イントラピクチャ推定コンポーネント215、イントラピクチャ予測コンポーネント217、動き補償コンポーネント219、動き推定コンポーネント221、スケーリングおよび逆変換コンポーネント229、フィルタ制御分析コンポーネント227、ループ内フィルタコンポーネント225、デコードされたピクチャバッファコンポーネント223、ならびにヘッダフォーマッティングおよびコンテキスト適応バイナリ算術コーディング(CABAC)コンポーネント231を含む。そのようなコンポーネントは表されたように結合される。図2では、黒い線はエンコード／デコードされるべきデータの動きを示し、一方、破線は他のコンポーネントの動作を制御する制御データの動きを示す。コーデックシステム200のコンポーネントは、すべて、エンコーダ内に存在し得る。デコーダは、コーデックシステム200のコンポーネントのサブセットを含み得る。たとえば、デコーダは、イントラピクチャ予測コンポーネント217、動き補償コンポーネント219、スケーリングおよび逆変換コンポーネント229、ループ内フィルタコンポーネント225、ならびにデコードされたピクチャバッファコンポーネント223を含み得る。これらのコンポーネントがここで説明される。

【0065】

区分されたビデオ信号201は、コーディング・ツリーによってピクセルのブロックへと区分された、キャプチャされたビデオシーケンスである。コーディング・ツリーは、様々な分割モードを利用して、ピクセルのブロックをピクセルのより小さいブロックへと細分する。これらのブロックは次いで、より小さいブロックへとさらに細分されることが可能である。ブロックは、コーディング・ツリーにおいてノードと呼ばれ得る。より大きい親ノードは、より小さい子ノードへと分割される。ノードが細分される回数は、ノード／コーディング・ツリーの深度と呼ばれる。いくつかの場合において、分割されたブロックはコーディング・ユニット(CU)に含まれることが可能である。たとえば、CUは、ルマブロック、赤差分クロマ(Cr)ブロック、および青差分クロマ(Cb)ブロックを、CUについての対応するシンタックス命令とともに含む、CTUの下位部分であることが可能である。分割モードは、利用される分割モードに依存して変化する形状のそれぞれ2つ、3つ、または4つの子ノードへとノードを区分するために利用される、二分木(BT)、三分木(TT)、および四分木(QT)を含み得る。区分されたビデオ信号201は、圧縮のために、汎用コーダ制御コンポーネント211、変換スケーリングおよび量子化コンポーネント213、イントラピクチャ推定コンポーネント215、フィルタ制御分析コンポーネント227、ならびに動き推定コンポーネント221に転送される。

【0066】

汎用コーダ制御コンポーネント211は、適用の制約に従って、ビデオシーケンスの画像のビットストリームへのコーディングに関連する決定を行うように構成される。たとえば、汎用コーダ制御コンポーネント211は、再構築品質に対するビットレート／ビットストリームサイズの最適化を管理する。そのような決定は、記憶空間／帯域幅の利用可能性および画像解像度の要求に基づいて行われ得る。汎用コーダ制御コンポーネント211は、また、バッファのアンダーランおよびオーバーランの問題を軽減するために、送信速度を考慮してバッファ利用を管理する。これらの問題を管理するために、汎用コーダ制御コンポーネント211は、他のコンポーネントによる区分、予測、およびフィルタリングを管理する。たとえば、汎用コーダ制御コンポーネント211は、圧縮の複雑さを動的に増加させて、解像度を増加させ、帯域幅使用を増加させ、または、圧縮の複雑さを減らして、解像度および帯域幅使用を減らし得る。従って、汎用コーダ制御コンポーネント211は、コーデックシステム200の他のコンポーネントを制御して、ビデオ信号再構築の品質をビットレートの関心とバランスをとる。汎用コーダ制御コンポーネント211は、制御データを作り出し、これは他のコンポーネントの動作を制御する。制御データは、また、ヘッダフォーマッティングおよびCABACコンポーネント231に転送されて、デコーダにおけるデコードのためのパラメータをシグナリングするためにビットストリーム内にエンコードされる。

【0067】

区分されたビデオ信号201は、また、インター予測のために動き推定コンポーネント221および動き補償コンポーネント219に送信される。区分されたビデオ信号201のフレームまたはスライスは、複数のビデオブロックへと分割され得る。動き推定コンポーネント221および動き補償コンポーネント219は、1つまたは複数の参照フレーム内の1つまたは複数のブロックに対して、受信された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、時間予測を提供する。コーデックシステム200は、複数のコーディングパスを実行して、たとえば、ビデオデータの各ブロックについて適切なコーディングモードを選択し得る。

【0068】

動き推定コンポーネント221および動き補償コンポーネント219は、高度に統合され得るが、概念上の目的のために別々に例示される。動き推定コンポーネント221によって実行される動き推定は、ビデオブロックについての動きを推定する、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、予測ブロックに対するコーディングされたオブジェクトのずれを示し得る。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ブロックに密接に一致することが見いだされるブロックである。予測ブロックは参照ブロックとも呼ばれ得る。そのようなピクセル差分は、絶対値差分和(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。HEVCは、CTU、コーディング・ツリー・ブロック(CTB)、およびCUを含む、いくつかのコーディングされたオブジェクトを利用する。たとえば、CTUはCTBへと分割されることが可能であり、これは、次いで、CU内の包含のためにCBへと分割されることが可能である。CUは、予測データを含む予測ユニット(PU)および／またはCUのための変換された残差データを含む変換ユニット(TU)としてエンコードされることが可能である。動き推定コンポーネント221は、レート歪み最適化プロセスの部分としてレート歪み分析を使用することによって、動きベクトル、PU、およびTUを生成する。たとえば、動き推定コンポーネント221は、現在のブロック／フレームのための複数の参照ブロック、複数の動きベクトルなどを決定してもよく、最良のレート歪み特性を有する参照ブロック、動きベクトルなどを選択してもよい。最良のレート歪み特性は、コーディング効率(たとえば、最終的なエンコードのサイズ)とビデオ再構築の品質(たとえば、圧縮によるデータ損失の量)の両方のバランスをとる。

【0069】

いくつかの例では、コーデックシステム200は、デコードされたピクチャバッファコンポーネント223に記憶されている参照ピクチャの整数より下のピクセル位置についての値を計算し得る。たとえば、ビデオコーデックシステム200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。従って、動き推定コンポーネント221は、完全ピクセル位置と分数ピクセル位置に対する動き探索を実行して、分数ピクセル精度を有する動きベクトルを出力し得る。動き推定コンポーネント221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内のビデオブロックのPUについて動きベクトルを計算する。動き推定コンポーネント221は、計算された動きベクトルを動きデータとしてエンコードのためにヘッダフォーマッティングおよびCABACコンポーネント231に出力し、動きを動き補償コンポーネント219に出力する。

【0070】

動き補償コンポーネント219によって実行される動き補償は、動き推定コンポーネント221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。再び、動き推定コンポーネント221および動き補償コンポーネント219は、いくつかの例では機能的に統合され得る。現在のビデオブロックのPUについての動きベクトルを受信すると、動き補償コンポーネント219は、動きベクトルが指し示す予測ブロックを位置決めし得る。残差ビデオブロックは次いで、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減算し、ピクセル差分値を形成することによって形成される。一般に、動き推定コンポーネント221は、ルマ成分に対する動き推定を実行し、動き補償コンポーネント219は、クロマ成分とルマ成分の両方

10

20

30

40

50

についてルマ成分に基づいて計算される動きベクトルを使用する。予測ブロックおよび残差ブロックは、変換スケーリングおよび量子化コンポーネント213に転送される。

【0071】

区分されたビデオ信号201は、また、イントラピクチャ推定コンポーネント215およびイントラピクチャ予測コンポーネント217に送信される。動き推定コンポーネント221および動き補償コンポーネント219のように、イントラピクチャ推定コンポーネント215およびイントラピクチャ予測コンポーネント217は高度に統合され得るが、概念上の目的のために別々に例示されている。イントラピクチャ推定コンポーネント215およびイントラピクチャ予測コンポーネント217は、上で説明されたように、フレーム間で動き推定コンポーネント221および動き補償コンポーネント219によって実行されるインター予測の代替として、現在のフレームの中のブロックに対して現在のブロックをイントラ予測する。特に、イントラピクチャ推定コンポーネント215は、現在のブロックをエンコードするために使用するイントラ予測モードを決定する。いくつかの例では、イントラピクチャ推定コンポーネント215は、複数のテストされたイントラ予測モードから現在のブロックをエンコードするために適切なイントラ予測モードを選択する。選択されたイントラ予測モードは次いで、エンコードのためにヘッダフォーマットティングおよびCABACコンポーネント231に転送される。

【0072】

たとえば、イントラピクチャ推定コンポーネント215は、様々なテストされたイントラ予測モードについてレート歪み分析を使用してレート歪み値を計算し、テストされたモードの中で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する。レート歪み分析は、一般に、エンコードされたブロックを作成するために使用されるビットレート(たとえば、ビットの数)とともに、エンコードされたブロックと、エンコードされたブロックを作成するためにエンコードされた元のエンコードされていないブロックとの間の歪み(またはエラー)の量を決定する。イントラピクチャ推定コンポーネント215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示すかを決定するために、様々なエンコードされたブロックについての歪みおよびレートから比を計算する。加えて、イントラピクチャ推定コンポーネント215は、レート歪み最適化(RDO)に基づいて、深度モデリングモード(DMM)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得る。

【0073】

イントラピクチャ予測コンポーネント217は、エンコーダにおいて実装されるとき、イントラピクチャ推定コンポーネント215によって決定される選択されたイントラ予測モードに基づいて予測ブロックから残差ブロックを生成し、または、デコーダにおいて実装されるとき、ビットストリームから残差ブロックを読み取り得る。残差ブロックは、行列として表現される、予測ブロックと元のブロックとの間の値における差分を含む。残差ブロックは次いで、変換スケーリングおよび量子化コンポーネント213に転送される。イントラピクチャ推定コンポーネント215およびイントラピクチャ予測コンポーネント217は、ルマおよびクロマ成分の両方に対して動作し得る。

【0074】

変換スケーリングおよび量子化コンポーネント213は、残差ブロックをさらに圧縮するように構成される。変換スケーリングおよび量子化コンポーネント213は、離散コサイン変換(DCT)、離散サイン変換(DST)、または概念的に類似の変換のような変換を残差ブロックに適用し、残差変換係数値を備えるビデオブロックを作成する。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ることが可能である。変換は、残差情報をピクセル値領域から周波数領域のような変換領域に変換し得る。変換スケーリングおよび量子化コンポーネント213は、また、たとえば周波数に基づいて、変換された残差情報をスケーリングするように構成される。そのようなスケーリングは、異なる周波数情報が異なる粒度で量子化されるように、スケール係数を残差情報に適用することを伴い、これは、再構築されたビデオの最終的な視覚的品質に影響し得る。変換スケ

10

20

30

40

50

ーリングおよび量子化コンポーネント213は、また、ビットレートをさらに低減するために変換係数を量子化するように構成される。量子化プロセスは、係数のいくつか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、変換スケーリングおよび量子化コンポーネント213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。量子化された変換係数は、ヘッダフォーマッティングおよびCABACコンポーネント231に転送されて、ビットストリーム内にエンコードされる。

【0075】

スケーリングおよび逆変換コンポーネント229は、動き推定をサポートするために、変換スケーリングおよび量子化コンポーネント213の逆の動作を適用する。スケーリングおよび逆変換コンポーネント229は、逆スケーリング、変換、および／または量子化を適用して、たとえば、別の現在のブロックについて予測ブロックになり得る参照ブロックとしての後の使用のために、ピクセル領域において残差ブロックを再構築する。動き推定コンポーネント221および／または動き補償コンポーネント219は、後のブロック／フレームの動き推定における使用のために残差ブロックを対応する予測ブロックに加算し戻すことによって参照ブロックを計算し得る。スケーリング、量子化、および変換の間に作り出されるアーティファクトを軽減するために、再構築された参照ブロックにフィルタが適用される。そのようなアーティファクトは、そうでなければ、後続のブロックが予測されるときに不正確な予測を引き起こす(および追加のアーティファクトを作り出す)ことがある。

【0076】

フィルタ制御分析コンポーネント227およびループ内フィルタコンポーネント225は、フィルタを残差ブロックおよび／または再構築された画像ブロックに適用する。たとえば、スケーリングおよび逆変換コンポーネント229からの変換された残差ブロックは、元の画像ブロックを再構築するために、イントラピクチャ予測コンポーネント217および／または動き補償コンポーネント219からの対応する予測ブロックと組み合わせられ得る。フィルタは、次いで、再構築された画像ブロックに適用され得る。いくつかの例では、フィルタは、代わりに、残差ブロックに適用され得る。図2の他のコンポーネントのように、フィルタ制御分析コンポーネント227およびループ内フィルタコンポーネント225は高度に統合され、一緒に実装され得るが、概念上の目的のために別々に描写されている。再構築された参照ブロックに適用されるフィルタは、特定の空間領域に適用され、そのようなフィルタがどのように適用されるかを調整するための複数のパラメータを含む。フィルタ制御分析コンポーネント227は、そのようなフィルタがどこで適用されるべきかを決定するために再構築された参照ブロックを分析し、対応するパラメータを設定する。そのようなデータは、エンコードのためにフィルタ制御データとしてヘッダフォーマッティングおよびCABACコンポーネント231に転送される。ループ内フィルタコンポーネント225は、フィルタ制御データに基づいてそのようなフィルタを適用する。フィルタは、デブロッキングフィルタ、ノイズ抑制フィルタ、SAOフィルタ、および適応ループフィルタを含み得る。そのようなフィルタは、例に依存して、空間／ピクセル領域で(たとえば、再構築されたピクセルブロックにおいて)、または周波数領域で適用され得る。

【0077】

エンコーダとして動作するとき、フィルタリングされた再構築された画像ブロック、残差ブロック、および／または予測ブロックは、上で論じられたような動き推定における後の使用のために、デコードされたピクチャバッファコンポーネント223に記憶される。デコーダとして動作するとき、デコードされたピクチャバッファコンポーネント223は、出力ビデオ信号の部分として、再構築されフィルタリングされたブロックを記憶し、ディスプレイに向けて転送する。デコードされたピクチャバッファコンポーネント223は、予測ブロック、残差ブロック、および／または再構築された画像ブロックを記憶することが可能な任意のメモリデバイスであり得る。

【0078】

ヘッダフォーマッティングおよびCABACコンポーネント231は、コーデックシステム2

00の様々なコンポーネントからデータを受信し、デコーダに向けた送信のためにそのようなデータをコーディングされたビットストリームへとエンコードする。具体的には、ヘッダフォーマッティングおよびCABACコンポーネント231は、一般的な制御データおよびフィルタ制御データのような制御データをエンコードするために、様々なヘッダを生成する。さらに、量子化された変換係数データの形式の残差データとともにイントラ予測および動きデータを含む予測データが、すべて、ビットストリーム内にエンコードされる。最終的なビットストリームは、元の区分されたビデオ信号201を再構築するためにデコーダによって望まれるすべての情報を含む。そのような情報は、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)、様々なブロックについてのエンコードコンテキストの定義、最も可能性の高いイントラ予測モードの指示、区分情報の指示なども含み得る。そのようなデータは、エントロピーコーディングを利用することによってエンコードされ得る。たとえば、情報は、コンテキスト適応可変長コーディング(CAVLC)、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利用することによってエンコードされ得る。エントロピーコーディングに続いて、コーディングされたビットストリームは、別のデバイス(たとえば、ビデオデコーダ)に送信され、または、後の送信もしくは取り出しのためにアーカイブされてもよい。

【0079】

図3は、一例のビデオエンコーダ300を例示するブロック図である。ビデオエンコーダ300は、コーデックシステム200のエンコード機能を実装するために、ならびに／または動作方法100のステップ101、103、105、107、および／もしくは109を実装するために利用され得る。エンコーダ300は、入力ビデオ信号を区分し、区分されたビデオ信号201と実質的に類似する区分されたビデオ信号301をもたらす。区分されたビデオ信号301は次いで圧縮され、エンコーダ300のコンポーネントによってビットストリームへとエンコードされる。

【0080】

具体的には、区分されたビデオ信号301は、イントラ予測のためにイントラピクチャ予測コンポーネント317に転送される。イントラピクチャ予測コンポーネント317は、イントラピクチャ推定コンポーネント215およびイントラピクチャ予測コンポーネント217と実質的に類似し得る。区分されたビデオ信号301は、また、デコードされたピクチャバッファコンポーネント323内の参照ブロックに基づくインター予測のために動き補償コンポーネント321に転送される。動き補償コンポーネント321は、動き推定コンポーネント221および動き補償コンポーネント219と実質的に類似し得る。イントラピクチャ予測コンポーネント317および動き補償コンポーネント321からの予測ブロックおよび残差ブロックは、残差ブロックの変換および量子化のために変換および量子化コンポーネント313に転送される。変換および量子化コンポーネント313は、変換スケーリングおよび量子化コンポーネント213と実質的に類似し得る。変換され量子化された残差ブロックおよび対応する予測ブロックは(関連付けられた制御データとともに)、ビットストリームへのコーディングのためにエントロピーコーディングコンポーネント331に転送される。エントロピーコーディングコンポーネント331は、ヘッダフォーマッティングおよびCABACコンポーネント231と実質的に類似し得る。

【0081】

変換され量子化された残差ブロックおよび／または対応する予測ブロックは、また、動き補償コンポーネント321による使用のために参照ブロックへの再構築のために、変換および量子化コンポーネント313から逆変換および量子化コンポーネント329に転送される。逆変換および量子化コンポーネント329は、スケーリングおよび逆変換コンポーネント229と実質的に類似し得る。ループ内フィルタコンポーネント325内のループ内フィルタは、また、例に依存して、残差ブロックおよび／または再構築された参照ブロックに適用される。ループ内フィルタコンポーネント325は、フィルタ制御分析コンポーネント227およびループ内フィルタコンポーネント225と実質的に類似し得る。ループ内フィルタコ

ンポーネント325は、ループ内フィルタコンポーネント225に関して論じられたような複数のフィルタを含み得る。フィルタリングされたブロックは次いで、動き補償コンポーネント321による参照ブロックとしての使用のために、デコードされたピクチャバッファコンポーネント323に記憶される。デコードされたピクチャバッファコンポーネント323は、デコードされたピクチャバッファコンポーネント223と実質的に類似し得る。

【0082】

図4は、一例のビデオデコーダ400を例示するブロック図である。ビデオデコーダ400は、コーデックシステム200のデコード機能を実装するために、ならびに／または動作方法100のステップ111、113、115、および／もしくは117を実装するために利用され得る。デコーダ400は、たとえばエンコーダ300から、ビットストリームを受信し、エンドユーザへの表示のために、再構築された出力ビデオ信号をビットストリームに基づいて生成する。

【0083】

ビットストリームは、エントロピーデコードコンポーネント433によって受信される。エントロピーデコードコンポーネント433は、CAVLC、CABAC、SBAC、PIPEコーディング、または他のエントロピーコーディング技法のようなエントロピーデコード方式を実装するように構成される。たとえば、エントロピーデコードコンポーネント433は、ヘッダ情報を利用して、ビットストリーム内に符号語としてエンコードされる追加のデータを解釈するためにコンテキストを提供し得る。デコードされた情報は、一般的な制御データ、フィルタ制御データ、区分情報、動き情報、予測データ、および残差ブロックからの量子化された変換係数のようなビデオ信号をデコードするための任意の望まれる情報を含む。量子化された変換係数は、残差ブロックへの再構築のために逆変換および量子化コンポーネント429に転送される。逆変換および量子化コンポーネント429は、逆変換および量子化コンポーネント329と類似し得る。

【0084】

再構築された残差ブロックおよび／または予測ブロックは、イントラ予測動作に基づく画像ブロックへの再構築のためにイントラピクチャ予測コンポーネント417に転送される。イントラピクチャ予測コンポーネント417は、イントラピクチャ推定コンポーネント215およびイントラピクチャ予測コンポーネント217と類似し得る。具体的には、イントラピクチャ予測コンポーネント417は、フレーム内で参照ブロックを位置決めするために予測モードを利用し、残差ブロックを結果に適用してイントラ予測された画像ブロックを再構築する。再構築されたイントラ予測された画像ブロックおよび／または残差ブロックならびに対応するインター予測データは、ループ内フィルタコンポーネント425を介してデコードされたピクチャバッファコンポーネント423に転送され、これらは、デコードされたピクチャバッファコンポーネント223およびループ内フィルタコンポーネント225とそれぞれ実質的に類似し得る。ループ内フィルタコンポーネント425は、再構築された画像ブロック、残差ブロック、および／または予測ブロックをフィルタリングし、そのような情報はデコードされたピクチャバッファコンポーネント423に記憶される。デコードされたピクチャバッファコンポーネント423からの再構築された画像ブロックは、インター予測のために動き補償コンポーネント421に転送される。動き補償コンポーネント421は、動き推定コンポーネント221および／または動き補償コンポーネント219と実質的に類似し得る。具体的には、動き補償コンポーネント421は、参照ブロックからの動きベクトルを利用して予測ブロックを生成し、残差ブロックを結果に適用して画像ブロックを再構築する。結果として生じる再構築されたブロックは、また、ループ内フィルタコンポーネント425を介してデコードされたピクチャバッファコンポーネント423に転送され得る。デコードされたピクチャバッファコンポーネント423は、追加の再構築された画像ブロックを記憶し続け、これらは区分情報を介してフレームへと再構築されることが可能である。そのようなフレームは、シーケンスにも配置され得る。シーケンスは、再構築された出力ビデオ信号としてディスプレイに向けて出力される。

【0085】

図5は、一例のビットストリーム500およびビットストリーム500から抽出されるサブビットストリーム501を例示する概略図である。たとえば、ビットストリーム500は、コーデックシステム200および／またはデコーダ400によるデコードのために、コーデックシステム200および／またはエンコーダ300によって生成されることが可能である。別の例として、ビットストリーム500は、ステップ111におけるデコーダによる使用のために、方法100のステップ109においてエンコーダによって生成され得る。

【0086】

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、複数のピクチャ・パラメータ・セット(PPS)512、複数のスライスヘッダ514、画像データ520、および1つまたは複数のSEIメッセージ515を含む。SPS510は、ビットストリーム500に含まれるビデオシーケンス内のすべてのピクチャに共通のシーケンスデータを含む。そのようなデータは、ピクチャサイズ、ビット深度、コーディングツールパラメータ、ビットレート制限などを含むことができる。PPS512は、1つまたは複数の対応するピクチャに固有のパラメータを含む。従って、ビデオシーケンス内の各ピクチャは、1つのPPS512を指し得る。PPS512は、対応するピクチャ内のタイルについて利用可能なコーディングツール、量子化パラメータ、オフセット、ピクチャ固有のコーディングツールパラメータ(たとえば、フィルタ制御)などを示すことができる。スライスヘッダ514は、ピクチャ内の1つまたは複数の対応するスライス524に固有のパラメータを含む。従って、ビデオシーケンス内の各スライス524は、スライスヘッダ514を参照し得る。スライスヘッダ514は、スライスタイプ情報、ピクチャ順序カウンタ(POC)、参照ピクチャリスト、予測重み、タイルエントリポイント、デブロッキングパラメータなどを含み得る。いくつかの例では、スライス524はタイルグループと呼ばれ得る。そのような場合において、スライスヘッダ514は、タイルグループヘッダと呼ばれ得る。SEIメッセージ515は、ブロックデコードのために要求されないメタデータを含む任意選択のメッセージであるが、ピクチャ出力のタイミング、表示設定、損失検出、損失隠蔽などを示すことのような関連する目的のために利用されることが可能である。

【0087】

画像データ520は、対応する変換され量子化される残差データとともにインター予測および／またはイントラ予測に従ってエンコードされるビデオデータを含む。そのような画像データ520は、エンコードの前に画像を区分するために使用される区分に従って分類される。たとえば、ビデオシーケンスはピクチャ521へと分割される。ピクチャ521はサブピクチャ522へとさらに分割されてもよく、これはスライス524へと分割される。スライス524はタイルおよび／またはCTUへとさらに分割され得る。CTUはコーディング・ツリーに基づいてコーディングブロックへとさらに分割される。コーディングブロックは次いで、予測機構に従ってエンコード／デコードされることが可能である。たとえば、ピクチャ521は1つまたは複数のサブピクチャ522を含むことができる。サブピクチャ522は1つまたは複数のスライス524を含み得る。ピクチャ521はPPS512を参照し、スライス524はスライスヘッダ514を参照する。サブピクチャ522は、全体のビデオシーケンス(セグメントとしても知られている)上で首尾一貫して区分されることが可能であり、従って、SPS510を参照し得る。各スライス524は1つまたは複数のタイルを含み得る。各スライス524、従ってピクチャ521およびサブピクチャ522は、複数のCTUも含むことができる。

【0088】

各ピクチャ521は、時間における対応する瞬間のためのビデオシーケンスと関連付けられる視覚データの全体のセットを含み得る。しかしながら、ある適用は、いくつかの場合においてピクチャ521の一部のみを表示することを望み得る。たとえば、仮想現実(VR)システムは、ピクチャ521のユーザ選択された領域を表示してもよく、これは、ピクチャ521において描写されるシーンに存在している感覚を作り出す。ユーザが見ることを望み得る領域は、ビットストリーム500がエンコードされるときに知られていない。従って、ピクチャ521は、ユーザが潜在的に見得る各々の可能性のある領域をサブピクチャ522として含んでもよく、これは、ユーザ入力に基づいて別々にデコードされて表示されることが

可能である。他の適用は関心領域を別々に表示し得る。たとえば、ピクチャインピクチャを有するテレビは、あるビデオシーケンスから特定の領域、従ってサブピクチャ522を、関連のないビデオシーケンスのピクチャ521上に表示することを望み得る。さらに別の例では、遠隔会議システムは、現在話しているユーザの全体のピクチャ521と、現在話していないユーザのサブピクチャ522とを表示し得る。従って、サブピクチャ522は、ピクチャ521の定義された領域を含み得る。時間的に動きが制約されるサブピクチャ522は、ピクチャ521の残りから別々にデコード可能であることが可能である。具体的には、時間動き制約サブピクチャは、時間動き制約サブピクチャの外側のサンプルを参照せずにエンコードされ、従って、ピクチャ521の残りを参照せずに完全なデコードのために十分な情報を含む。

10

【0089】

各スライス524は、左上の角にあるCTUおよび右下の角にあるCTUによって定義される長方形であり得る。いくつかの例では、スライス524は、左から右および上から下に進むラスタ走査順で、一連のタイルおよび／またはCTUを含む。他の例では、スライス524は長方形スライスである。長方形スライスは、ラスタ走査順に従ってピクチャの全体の幅を横切らないことがある。代わりに、長方形スライスは、CTUおよび／またはタイル行、ならびにCTUおよび／またはタイル列に関して定義される、ピクチャ521および／またはサブピクチャ522の長方形および／または正方形領域を含み得る。スライス524は、デコーダによって別々に表示されることが可能である最小単位である。従って、ピクチャ521からのスライス524は、ピクチャ521の望まれる領域を別々に描写するために異なるサブピクチャ522に割り当てられ得る。

20

【0090】

デコーダは、ピクチャ521の1つまたは複数のサブピクチャ523を表示し得る。サブピクチャ523は、サブピクチャ522のユーザ選択されたまたはあらかじめ定義されたサブグループである。たとえば、ピクチャ521は9個のサブピクチャ522へと分割され得るが、デコーダは、サブピクチャ522のグループから単一のサブピクチャ523のみを表示し得る。サブピクチャ523はスライス525を含み、これはスライス524の選択されたまたはあらかじめ定義されたサブグループである。サブピクチャ523の別々の表示を可能にするために、サブビットストリーム501はビットストリーム500から抽出され得る(529)。抽出529は、デコーダがサブビットストリーム501のみを受信するように、エンコーダ側で起こり得る。他の場合には、ビットストリーム500全体がデコーダに送信され、デコーダが別々のデコードのためにサブビットストリーム501を抽出する(529)。サブビットストリーム501は、いくつかの場合において一般にビットストリームとも呼ばれ得ることが留意されるべきである。サブビットストリーム501は、スライスヘッダ514、およびサブピクチャ523および／またはスライス525に関連するSEIメッセージ515とともに、SPS510、PPS512、選択されたサブピクチャ523を含む。

30

【0091】

本開示は、デコーダにおけるサブピクチャ523の選択および表示のためにサブピクチャ522の効率的なコーディングをサポートするために、様々なデータをシグナリングする。SPS510は、サブピクチャサイズ531、サブピクチャ位置532、およびサブピクチャ522の完全なセットに関するサブピクチャID533を含む。サブピクチャサイズ531は、対応するサブピクチャ522についてのルマサンプルにおけるサブピクチャ高さおよびルマサンプルにおけるサブピクチャ幅を含む。サブピクチャ位置532は、対応するサブピクチャ522の左上サンプルとピクチャ521の左上サンプルとの間のオフセット距離を含む。サブピクチャ位置532およびサブピクチャサイズ531は、対応するサブピクチャ522のレイアウトを定義する。サブピクチャID533は、対応するサブピクチャ522を一意に識別するデータを含む。サブピクチャID533は、サブピクチャ522のラスタ走査インデックスまたは他の定義された値であり得る。従って、デコーダは、SPS510を読み取り、各サブピクチャ522のサイズ、位置、およびIDを決定することができる。いくつかのビデオコーディングシステムでは、サブピクチャ522はピクチャ521から区分されるので、サブピクチャ522に

40

50

関連するデータはPPS512に含まれ得る。しかしながら、サブピクチャ522を作り出すために使用される区分は、ビデオシーケンス／セグメント上で首尾一貫したサブピクチャ522の区分に依存する、ROIを基にした適用、VR適用などのような適用により使用され得る。それゆえ、サブピクチャ522の区分は、一般にピクチャごとに変化しない。SPS510内にサブピクチャ522のためのレイアウト情報を配置することは、レイアウトが各PPS512について冗長にシグナリングされる(これは、いくつかの場合において各ピクチャ521についてシグナリングされ得る)のではなく、シーケンス／セグメントについて1回のみシグナリングされることを保証する。また、そのような情報を導出するためにデコーダに依存する代わりに、サブピクチャ522情報をシグナリングすることは、損失したパケットの場合におけるエラーの確率を低減し、サブピクチャ523を抽出することに関して追加の機能をサポートする。従って、SPS510内でサブピクチャ522のレイアウトをシグナリングすることは、エンコーダおよび／またはデコーダの機能を改善する。

【0092】

SPS510は、また、サブピクチャ522の完全なセットに関連する動き制約サブピクチャフラグ534を含む。動き制約サブピクチャフラグ534は、各サブピクチャ522が時間動き制約サブピクチャであるかどうかを示す。従って、デコーダは、動き制約サブピクチャフラグ534を読み取り、サブピクチャ522のいずれが他のサブピクチャ522をデコードすることなく別々に抽出され表示されることが可能であるかを決定することができる。これは、選択されたサブピクチャ522が時間動き制約サブピクチャとしてコーディングされることを可能にし、一方、他のサブピクチャ522が増加したコーディング効率のためにそのような制約なしでコーディングされることを可能にする。

【0093】

サブピクチャID533は、スライスヘッダ514にも含まれる。各スライスヘッダ514は、スライス524の対応するセットに関連するデータを含む。従って、スライスヘッダ514は、スライスヘッダ514と関連付けられるスライス524に対応するサブピクチャID533のみを含む。それゆえ、デコーダは、スライス524を受信し、スライスヘッダ514からサブピクチャID533を取得し、どのサブピクチャ522がスライス524を含むかを決定することができる。デコーダは、また、SPS510内の関連するデータと相関させるために、スライスヘッダ514からのサブピクチャID533を使用することができる。それゆえ、デコーダは、SPS510および関連するスライスヘッダ514を読み取ることによって、サブピクチャ522/523およびスライス524/525をどのように配置するかを決定することができる。これは、いくつかのサブピクチャ522が送信において損失し、またはコーディング効率を増加させるために意図的に省略されても、サブピクチャ523およびスライス525がデコードされることを可能にする。

【0094】

SEIメッセージ515は、また、サブピクチャレベル535を含み得る。サブピクチャレベル535は、対応するサブピクチャ522をデコードするために必要なハードウェアリソースを示す。このようにして、各サブピクチャ522は、他のサブピクチャ522とは独立にコーディングされることが可能である。これは、各サブピクチャ522がデコーダにおいて正しい量のハードウェアリソースを割り振られることが可能であることを保証にする。そのようなサブピクチャレベル535がなしで、各サブピクチャ522は、最も複雑なサブピクチャ522をデコードするために十分なリソースを割り振られる。従って、サブピクチャレベル535は、サブピクチャ522が変化するハードウェアリソース要件と関連付けられるならば、デコーダがハードウェアリソースを過剰に割り振ることを防ぐ。

【0095】

図6は、サブピクチャ622へと区分される一例のピクチャ600を例示する概略図である。たとえば、ピクチャ600は、たとえばコーデックシステム200、エンコーダ300、および／またはデコーダ400によって、ビットストリーム500においてエンコードされ、ビットストリーム500からデコードされることが可能である。さらに、ピクチャ600は、方法100に従ったエンコードおよびデコードをサポートするために、サブビットストリーム50

1において区分され、および／または含まれることが可能である。

【0096】

ピクチャ600は、ピクチャ521と実質的に類似し得る。さらに、ピクチャ600はサブピクチャ622へと区分されてもよく、これはサブピクチャ522と実質的に類似する。サブピクチャ622は、各々、サブピクチャサイズ631を含み、これはサブピクチャサイズ531としてビットストリーム500に含まれ得る。サブピクチャサイズ631は、サブピクチャ幅631aおよびサブピクチャ高さ631bを含む。サブピクチャ幅631aは、ルマサンプルの単位での対応するサブピクチャ622の幅である。サブピクチャ高さ631bは、ルマサンプルの単位での対応するサブピクチャ622の高さである。サブピクチャ622は、各々、サブピクチャID633を含み、これはサブピクチャID633としてビットストリーム500に含まれ得る。サブピクチャID633は、各サブピクチャ622を一意に識別する任意の値であり得る。表される例では、サブピクチャID633はサブピクチャ622のインデックスである。サブピクチャ622は、各々、位置632を含み、これはサブピクチャ位置532としてビットストリーム500に含まれ得る。位置632は、対応するサブピクチャ622の左上サンプルとピクチャ600の左上サンプル642との間のオフセットとして表現される。

【0097】

また表されるように、いくつかのサブピクチャ622は時間動き制約サブピクチャ634であってもよく、他のサブピクチャ622はそうでなくてもよい。表される例では、5のサブピクチャID633を有するサブピクチャ622は、時間動き制約サブピクチャ634である。これは、5として識別されるサブピクチャ622が、いずれの他のサブピクチャ622も参照せずにコーディングされ、従って、他のサブピクチャ622からのデータを考慮することなく抽出され別々にデコードされることが可能であることを示す。どのサブピクチャ622が時間動き制約サブピクチャ634であるかの指示は、ビットストリーム500において動き制約サブピクチャフラグ534においてシグナリングされることが可能である。

【0098】

表されるように、サブピクチャ622は、間隙または重複なしでピクチャ600を包含するように制約されることが可能である。間隙は、いずれのサブピクチャ622にも含まれないピクチャ600の領域である。重複は、1つより多くのサブピクチャ622に含まれるピクチャ600の領域である。図6に表される例では、サブピクチャ622は、間隙と重複の両方を防ぐようにピクチャ600から区分される。間隙は、ピクチャ600のサンプルがサブピクチャ622の外側に残されることを引き起こす。重複は、関連付けられたスライスが複数のサブピクチャ622に含まれることを引き起こす。従って、間隙および重複は、サンプルが、サブピクチャ622が異なってコーディングされるとき、異なる扱いによって影響されることを引き起こし得る。これがエンコーダにおいて許容されるならば、デコーダは、そのデコード方式がまれに使用されるときでも、そのようなコーディング方式をサポートしなければならない。サブピクチャ622の間隙および重複を許容しないことによって、サブピクチャサイズ631および位置632を決定するときにデコーダが潜在的な間隙および重複を考慮することが要求されないの、デコーダの複雑さが減らされることが可能である。さらに、サブピクチャ622の間隙および重複を許容しないことは、エンコーダにおけるRDOプロセスの複雑さを低減する。これは、ビデオシーケンスについてエンコードを選択するときにエンコーダが間隙と重複の事例を考慮することを省略できるからである。従って、間隙および重複を回避することは、エンコーダおよびデコーダにおけるメモリリソースおよび／または処理リソースの使用を低減し得る。

【0099】

図7は、スライス724をサブピクチャ722のレイアウトに関連させるための一例の機構700を例示する概略図である。たとえば、機構700はピクチャ600に適用され得る。さらに、機構700は、たとえばコーデックシステム200、エンコーダ300、および／またはデコーダ400によって、ビットストリーム500内のデータに基づいて適用されることが可能である。さらに、機構700は、方法100に従ったエンコードおよびデコードをサポートするために利用されることが可能である。

【0100】

機構700は、それぞれ、スライス524/525およびサブピクチャ522/523のようなサブピクチャ722内のスライス724に適用されることが可能である。表される例では、サブピクチャ722は、第1のスライス724a、第2のスライス724b、および第3のスライス724cを含む。スライス724の各々についてのスライスヘッダは、サブピクチャ722についてのサブピクチャID733を含む。デコーダは、スライスヘッダからのサブピクチャID733をSPS内のサブピクチャID733と照合することができる。デコーダは次いで、サブピクチャID733に基づいてSPSからサブピクチャ722の位置732およびサイズを決定することができる。サブピクチャ722は、位置732を使用して、ピクチャの左上の角742にある左上サンプルに対して配置されることが可能である。位置732に対するサブピクチャ722の高さおよび幅を設定するために、サイズが使用されることが可能である。そうすると、スライス724がサブピクチャ722に含まれることが可能である。従って、スライス724は、他のサブピクチャを参照せずに、サブピクチャID733に基づいて正しいサブピクチャ722内に配置されることが可能である。これは、他の損失したサブピクチャがサブピクチャ722のデコードを変えないので、エラー訂正をサポートする。これは、また、サブピクチャ722のみを抽出する適用をサポートし、他のサブピクチャを送信することを回避する。従って、サブピクチャID733は、増加した機能および／または増加したコーディング効率をサポートし、これは、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。

【0101】

図8は、サブピクチャ822へと区分される別の例のピクチャ800を例示する概略図である。ピクチャ800は、ピクチャ600と実質的に類似し得る。加えて、ピクチャ800は、たとえばコーデックシステム200、エンコーダ300、および／またはデコーダ400によって、ビットストリーム500内にエンコードされ、ビットストリーム500からデコードされることが可能である。さらに、ピクチャ800は、方法100および／または機構700に従ったエンコードおよびデコードをサポートするために、サブビットストリーム501において区分され、および／または含まれることが可能である。

【0102】

ピクチャ800はサブピクチャ822を含み、これはサブピクチャ522、523、622、および／または722と実質的に類似し得る。サブピクチャ822は複数のCTU825へと分割される。CTU825は、標準化されたビデオコーディングシステムにおける基本コーディング単位である。CTU825はコーディング・ツリーによってコーディングブロックへと細分され、これはインター予測またはイントラ予測に従ってコーディングされる。表されるように、いくつかのサブピクチャ822aは、CTU825のサイズの倍数であるサブピクチャ幅およびサブピクチャ高さを含むように制約される。表される例では、サブピクチャ822aは、6個のCTU825の高さおよび5個のCTU825の幅を有する。この制約は、ピクチャの右の境界801に配置されたサブピクチャ822bについて、およびピクチャの下の境界802に配置されたサブピクチャ822cについて、取り除かれる。表される例では、サブピクチャ822bは、5個と6個のCTU825の間の幅を有する。しかしながら、ピクチャの下の境界802に配置されないサブピクチャ822bは、依然として、CTU825のサイズの倍数であるサブピクチャ高さを維持するように制約される。表される例では、サブピクチャ822cは、6個と7個のCTU825の間の高さを有する。しかしながら、ピクチャの右の境界801に配置されないサブピクチャ822cは、依然として、CTU825のサイズの倍数であるサブピクチャ幅を維持するように制約される。

【0103】

上で注記されたように、いくつかのビデオシステムは、CTU825のサイズの倍数である高さおよび幅を含むようにサブピクチャ822を限定し得る。これは、サブピクチャ822が、多くのピクチャレイアウトとともに、たとえばCTU825のサイズの倍数でない合計の幅または高さを含むピクチャ800とともに、正しく動作することを妨げ得る。下のサブピクチャ822cおよび右のサブピクチャ822bが、CTU825のサイズの倍数でない高さと幅をそ

れぞれ含むことを許容することによって、サブピクチャ822は、デコードエラーを引き起こすことなく任意のピクチャ800とともに使用され得る。これは、エンコーダおよびデコーダの機能を増加させることをもたらす。さらに、増加した機能は、エンコーダがピクチャをより効率的にコーディングすることを可能にし、これは、ネットワークリソース、メモリリソース、および／またはエンコーダとデコーダにおける処理リソースの使用を低減する。

【0104】

ここで説明されるように、本開示は、ビデオコーディングにおけるサブピクチャを基にしたピクチャ区分の設計を説明する。サブピクチャは、ピクチャのために使用されるのと類似したデコードプロセスを使用して独立にデコードされることが可能である、ピクチャ内の長方形エリアである。本開示は、サブピクチャ抽出のためのプロセスとともに、コーディングされたビデオシーケンスおよび／またはビットストリームにおけるサブピクチャのシグナリングに関する。技法の説明は、ITU-TおよびISO/IECのJVETによるVVCに基づく。しかしながら、技法は、他のビデオコーデック仕様にも適用される。以下はここで説明される例示の実施形態である。そのような実施形態は、個々に、または組み合わせて適用され得る。

【0105】

コーディングされたビデオシーケンス(CVS)内に存在し得るサブピクチャに関連する情報は、SPSのようなシーケンスレベルパラメータセット内でシグナリングされ得る。そのようなシグナリングは、以下の情報を含み得る。CVSの各ピクチャに存在するサブピクチャの数は、SPS内でシグナリングされ得る。SPSまたはCVSの文脈では、すべてのアクセసుユニット(AU)について同じに位置付けられるサブピクチャは、集合的にサブピクチャシーケンスと呼ばれ得る。各サブピクチャの属性を記述する情報をさらに指定するためのループも、SPSに含まれ得る。この情報は、サブピクチャ識別情報、サブピクチャの位置(たとえば、サブピクチャの左上角ルマサンプルとピクチャの左上角ルマサンプルとの間のオフセット距離)、およびサブピクチャのサイズを備え得る。加えて、SPSは、各サブピクチャが動き制約サブピクチャ(MCTSの機能を含む)であるかどうかをシグナリングし得る。各サブピクチャについてのプロファイル、階層、およびレベル情報も、デコーダにおいてシグナリングされ、または導出可能であり得る。そのような情報は、元のビットストリームからサブピクチャを抽出することによって作り出されるビットストリームについてのプロファイル、階層、およびレベル情報を決定するために利用され得る。各サブピクチャのプロファイルおよび階層は、ビットストリーム全体のプロファイルおよび階層と同じであるように導出され得る。各サブピクチャについてのレベルは明示的にシグナリングされ得る。そのようなシグナリングは、SPSに含まれるループに存在し得る。シーケンスレベルの仮想参照デコーダ(HRD)パラメータは、各サブピクチャ(または等価的に、各サブピクチャシーケンス)についてのSPSのビデオ使用可能性情報(VUI)セクションにおいてシグナリングされ得る。

【0106】

ピクチャが2つ以上のサブピクチャへと区分されないとき、サブピクチャの属性(たとえば、位置、サイズなど)は、サブピクチャIDを除き、ビットストリームにおいて存在しない／シグナリングされないことがある。CVS内のピクチャのサブピクチャが抽出されるとき、新しいビットストリーム内の各アクセసుユニットはサブピクチャを含まないことがある。この場合、新しいビットストリーム内の各AU内のピクチャは、複数のサブピクチャへと区分されない。従って、SPS内の位置およびサイズのようなサブピクチャ属性をシグナリングする必要はなく、なぜなら、そのような情報はピクチャ属性から導出されることが可能であるからである。しかしながら、IDは抽出されたサブピクチャに含まれるVCL NALユニット／タイルグループにより参照され得るので、サブピクチャ識別情報は依然としてシグナリングされてもよい。これは、サブピクチャIDが、サブピクチャを抽出するときに同じままであることを可能にし得る。

【0107】

ピクチャ内のサブピクチャの位置(xオフセットおよびyオフセット)は、ルマサンプルの単位でシグナリングされることが可能である。位置は、サブピクチャの左上角ルマサンプルとピクチャの左上角ルマサンプルとの間の距離を表現する。代替的に、ピクチャ内のサブピクチャの位置は、最小コーディングルマブロックサイズ(MinCbSizeY)の単位でシグナリングされることが可能である。代替的に、サブピクチャ位置オフセットの単位は、パラメータセット内のシンタックス要素によって明示的に示され得る。単位は、CtbSizeY、MinCbSizeY、ルマサンプル、または他の値であり得る。

【0108】

サブピクチャのサイズ(サブピクチャ幅およびサブピクチャ高さ)は、ルマサンプルの単位でシグナリングされることが可能である。代替的に、サブピクチャのサイズは、最小コーディングルマブロックサイズ(MinCbSizeY)の単位でシグナリングされることが可能である。代替的に、サブピクチャサイズ値の単位は、パラメータセット内のシンタックス要素によって明示的に示されることが可能である。単位は、CtbSizeY、MinCbSizeY、ルマサンプル、または他の値であり得る。サブピクチャの右の境界がピクチャの右の境界と一致しないとき、サブピクチャの幅は、ルマCTUサイズ(CtbSizeY)の整数倍であることが要求され得る。同様に、サブピクチャの下の境界がピクチャの下の境界と一致しないとき、サブピクチャの高さは、ルマCTUサイズ(CtbSizeY)の整数倍であることが要求され得る。サブピクチャの幅がルマCTUサイズの整数倍でないならば、サブピクチャは、ピクチャ内の最も右の位置に位置決めされることが要求され得る。同様に、サブピクチャの高さがルマCTUサイズの整数倍でないならば、サブピクチャは、ピクチャ内の最も下の位置に位置決めされることが要求され得る。いくつかの場合において、サブピクチャの幅はルマCTUサイズの単位でシグナリングされることが可能であるが、サブピクチャの幅はルマCTUサイズの整数倍でない。この場合、ルマサンプルにおける実際の幅が、サブピクチャのオフセット位置に基づいて導出されることが可能である。サブピクチャの幅はルマCTUサイズに基づいて導出されることが可能であり、ピクチャの高さはルマサンプルに基づいて導出されることが可能である。同様に、サブピクチャの高さはルマCTUサイズの単位でシグナリングされ得るが、サブピクチャの高さはルマCTUサイズの整数倍でない。そのような場合、ルマサンプルにおける実際の高さは、サブピクチャのオフセット位置に基づいて導出されることが可能である。サブピクチャの高さはルマCTUサイズに基づいて導出されることが可能であり、ピクチャの高さはルマサンプルに基づいて導出されることが可能である。

【0109】

任意のサブピクチャについて、サブピクチャIDは、サブピクチャインデックスと異なり得る。サブピクチャインデックスは、SPS内のサブピクチャのループにおいてシグナリングされるようなサブピクチャのインデックスであり得る。サブピクチャIDは、ピクチャ内のサブピクチャラスタ走査順でのサブピクチャのインデックスであり得る。各サブピクチャのサブピクチャIDの値がサブピクチャインデックスと同じであるとき、サブピクチャIDはシグナリングまたは導出され得る。各サブピクチャのサブピクチャIDがサブピクチャインデックスと異なるとき、サブピクチャIDは明示的にシグナリングされる。サブピクチャIDのシグナリングのためのビットの数は、サブピクチャ属性を含む同じパラメータセット内で(たとえば、SPS内で)シグナリングされ得る。サブピクチャIDについてのいくつかの値が、ある目的のために予約され得る。たとえば、タイルグループヘッダが、どのサブピクチャがタイルグループを含むかを指定するためにサブピクチャIDを含むとき、エミュレーション防止コードの偶然の包含を防ぐためにタイルグループヘッダの最初の数ビットがすべて0ではないことを保証するために、サブピクチャのために値0が予約され、使用されないことがある。ピクチャのサブピクチャが、間隙なし、かつ重複なしでピクチャのエリア全体を包含しない任意選択の場合において、値(たとえば、値1)が、任意のサブピクチャの部分でないタイルグループのために予約され得る。代替的に、残りのエリアのサブピクチャIDが明示的にシグナリングされる。サブピクチャIDをシグナリングするためのビットの数は、以下のように制約され得る。値の範囲は、サブピクチャIDの予約された値を含めて、ピクチャ内のすべてのサブピクチャを一意に識別するために十分であるべきである

。たとえば、サブピクチャIDについてのビットの最小数は、 $\text{Ceil}(\text{Log}_2(\text{ピクチャ内のサブピクチャの数} + \text{予約されたサブピクチャIDの数}))$ の値であることが可能である。

【0110】

サブピクチャの和集合が、間隙なし、かつ重複なしでピクチャ全体を包含しなければならないことが制約され得る。この制約が適用されるとき、各サブピクチャについて、サブピクチャが動き制約サブピクチャであるかどうかを指定するためのフラグが存在してもよく、これは、サブピクチャが抽出されることが可能であることを示す。代替的に、サブピクチャの和集合は、ピクチャ全体を包含しないことがあるが、重複が許容されないことがある。

【0111】

NALユニットビットの残りを解析することを抽出器に要求することなくサブピクチャ抽出プロセスを助けるために、サブピクチャIDがNALユニットヘッダの直後に存在し得る。VCL NALユニットについて、サブピクチャIDは、タイルグループヘッダの最初のビット内に存在し得る。非VCL NALユニットについて、以下が当てはまり得る。SPSについて、サブピクチャIDは、NALユニットヘッダの直後に存在する必要はない。PPSについて、同じピクチャのすべてのタイルグループが同じPPSを参照するように制約されるならば、サブピクチャIDは、そのNALユニットヘッダの直後に存在する必要はない。同じピクチャのタイルグループが異なるPPSを参照することが許容されるならば、サブピクチャIDは、PPSの最初のビット内(たとえば、NALユニットヘッダの直後)に存在し得る。この場合、1つのピクチャの任意のタイルグループが、同じPPSを共有することが許容され得る。代替的に、同じピクチャのタイルグループが異なるPPSを参照することが許容され、同じピクチャの異なるタイルグループが同じPPSを共有することも許容されるとき、サブピクチャIDはPPSシンタックス内に存在しないことがある。代替的に、同じピクチャのタイルグループが異なるPPSを参照することが許容され、同じピクチャの異なるタイルグループも同じPPSを共有することが許容されるとき、サブピクチャIDのリストがPPSシンタックス内に存在し得る。このリストは、PPSが適用されるサブピクチャを示す。他の非VCL NALユニットについて、非VCLユニット(たとえば、access unit delimiter、end of sequence、end of bitstreamなど)がピクチャレベルまたはそれより上に適用されるならば、サブピクチャIDは、NALユニットヘッダの直後に存在しなくてもよい。そうでなければ、サブピクチャIDはNALユニットヘッダの直後に存在してもよい。

【0112】

上記のSPSシグナリングを用いて、個々のサブピクチャ内のタイル区分は、PPS内でシグナリングされ得る。同じピクチャ内のタイルグループは、異なるPPSを参照することが許容され得る。この場合、タイルグルーピングは、各サブピクチャ内のみであり得る。タイルグルーピングの概念は、サブピクチャのタイルへの区分である。

【0113】

代替的に、個々のサブピクチャ内のタイル区分を記述するためのパラメータセットが定義される。そのようなパラメータセットは、サブピクチャ・パラメータ・セット(SPPS)と呼ばれ得る。SPPSはSPSを参照する。SPS IDを参照するシンタックス要素はSPPS内に存在する。SPPSはサブピクチャIDを含み得る。サブピクチャ抽出の目的のために、サブピクチャIDを参照するシンタックス要素は、SPPS内の最初のシンタックス要素である。SPPSはタイル構造(たとえば、列の数、行の数、均一なタイルの間隔など)を含む。SPPSは、ループフィルタが関連付けられたサブピクチャ境界にわたって有効にされているかどうかを示すためにフラグを含み得る。代替的に、各サブピクチャについてのサブピクチャ属性は、SPS内の代わりにSPPS内でシグナリングされ得る。個々のサブピクチャ内のタイル区分は、依然として、PPS内でシグナリングされ得る。同じピクチャ内のタイルグループは、異なるPPSを参照することが許容される。いったんSPPSがアクティブ化されると、SPPSはデコード順で連続するAUのシーケンスについて続く。しかしながら、SPPSは、CVSの開始でないAUにおいて非アクティブ化／アクティブ化され得る。いくつかのAUにおける複数のサブピクチャを有する単層のビットストリームのデコードプロセスの間の任意の瞬

間において、複数のSPPSがアクティブであり得る。SPPSは、AUの異なるサブピクチャによって共有され得る。代替的に、SPPSおよびPPSは1つのパラメータセットへと合併され得る。そのような場合、同じピクチャのすべてのタイルグループが同じPPSを参照することは要求されなくてもよい。同じサブピクチャ内のすべてのタイルグループが、SPPSとPPSの間の合併から結果として生じる同じパラメータセットを参照し得るような制約が適用され得る。

【0114】

サブピクチャIDをシグナリングするために使用されるビットの数は、NALユニットヘッダ内でシグナリングされ得る。NALユニットヘッダ内に存在するとき、そのような情報は、NALユニットのペイロードの最初(たとえば、NALユニットヘッダの直後の最初の数ビット)においてサブピクチャID値を解析する際に、サブピクチャ抽出プロセスを助け得る。そのようなシグナリングのために、NALユニットヘッダ内の予約されたビット(たとえば、7個の予約されたビット)のいくつか、NALユニットヘッダの長さを増加させることを避けるために使用され得る。そのようなシグナリングのためのビットの数は、sub-picture-ID-bit-lenの値を包含し得る。たとえば、VVC NALユニットヘッダの7個の予約されたビットのうちの4ビットが、この目的のために使用され得る。

【0115】

サブピクチャをデコードするとき、各コーディング・ツリー・ブロックの位置(たとえば、xCtbおよびyCtb)が、サブピクチャ内のルマサンプル位置の代わりにピクチャ内の実際のルマサンプル位置に調整され得る。このようにして、コーディング・ツリー・ブロックがサブピクチャの代わりにピクチャを参照してデコードされるので、各参照ピクチャから同じに位置付けられるサブピクチャの抽出が回避されることが可能である。コーディング・ツリー・ブロックの位置を調整するために、変数SubpictureXOffsetおよびSubpictureYOffsetが、サブピクチャの位置(subpic_x_offsetおよびsubpic_y_offset)に基づいて導出されることが可能である。変数の値は、サブピクチャ内の各コーディング・ツリー・ブロックの、ルマサンプル位置xおよびyの座標の値にそれぞれ加算され得る。

【0116】

サブピクチャ抽出プロセスは以下のように定義されることが可能である。プロセスへの入力、抽出されるべき標的サブピクチャである。これは、サブピクチャIDまたはサブピクチャ位置の形式であることが可能である。入力がサブピクチャの位置であるとき、関連付けられたサブピクチャIDが、SPS内のサブピクチャ情報を解析することによって解決されることが可能である。非VCL NALユニットについて、以下が当てはまる。ピクチャサイズおよびレベルに関するSPS内のシンタックス要素が、サブピクチャのサイズおよびレベル情報を用いて更新され得る。以下の非VCL NALユニット、すなわち、PPS、Access Unit Delimiter (AUD)、End of Sequence (EOS)、End of Bitstream (EOB)、およびピクチャレベルまたはそれより上に適用可能である任意の他の非VCL NALユニットは、変更なしで保持される。標的サブピクチャIDに等しくないサブピクチャIDを有する残りの非VCL NALユニットは、取り除かれ得る。標的サブピクチャIDに等しくないサブピクチャIDを有するVCL NALユニットも、取り除かれ得る。

【0117】

SEIメッセージをネストするシーケンスレベルサブピクチャは、サブピクチャのセットについてAUレベルまたはサブピクチャレベルSEIメッセージのネストのために使用され得る。これは、バッファリング期間、ピクチャタイミング、および非HRD SEIメッセージを含み得る。SEIメッセージをネストするこのサブピクチャのシンタックスおよびセマンティクスは、以下の通りであることが可能である。無方向性メディアフォーマット(OMAF)環境内でのようなシステム動作のために、ビューポートを包含するサブピクチャシーケンスのセットは、OMAFプレーヤによって要求されデコードされ得る。従って、シーケンスレベルSEIメッセージは、長方形ピクチャ領域を集合的に包含するサブピクチャシーケンスのセットの情報を搬送するために使用される。この情報はシステムによって使用されることが可能であり、この情報は、サブピクチャシーケンスのセットのビットレートとともに

に要求されるデコード能力を示す。この情報は、サブピクチャシーケンスのセットのみを含むビットストリームのレベルを示す。この情報は、また、サブピクチャシーケンスのセットのみを含むビットストリームのビットレートを示す。任意選択で、サブビットストリーム抽出プロセスは、サブピクチャシーケンスのセットについて指定され得る。これを行うことの利益は、サブピクチャシーケンスのセットのみを含むビットストリームも準拠することになり得ることである。欠点は、異なるビューポートサイズの可能性を考慮する際に、すでに大きな可能性のある数の個々のサブピクチャシーケンスに加えて、多くのそのようなセットが存在し得ることである。

【0118】

一例の実施形態では、開示される例の1つまたは複数は以下のように実装され得る。サブピクチャが、ピクチャ内の1つまたは複数のタイルグループの長方形領域として定義され得る。許容される二分割プロセスは以下のように定義され得る。このプロセスへの入力、二分割モードbtSplit、コーディングブロック幅cbWidth、コーディングブロック高さcbHeight、ピクチャの左上ルマサンプルに対する考慮されるコーディングブロックの左上ルマサンプルの位置(x0,y0)、マルチタイプ木深度mttDepth、オフセットを有する最大マルチタイプ木深度maxMttDepth、最大二分木サイズmaxBtSize、および区分インデックスpartIdxである。このプロセスの出力は、変数allowBtSplitである。

【0119】

【表1】

	btSplit == SPLIT_BT_VER	btSplit == SPLIT_BT_HOR
parallelTtSplit	SPLIT_TT_VER	SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

btSplitに基づく parallelTtSplit および cbSize の仕様

【0120】

変数parallelTtSplitおよびcbSizeは、上で指定されたように導出される。変数allowBtSplitは以下のように導出される。以下の条件、すなわち、cbSizeがMinBtSizeY以下である、cbWidthがmaxBtSizeより大きい、cbHeightがmaxBtSizeより大きい、およびmttDepthがmaxMttDepth以上である、のうちの1つまたは複数が真であるならば、allowBtSplitはFALSEに等しく設定される。そうでなければ、以下の条件、すなわち、btSplitがSPLIT_BT_VERに等しい、およびy0+cbHeightがSupPicBottomBorderInPicより大きい、のすべてが真であるならば、allowBtSplitはFALSEに等しく設定される。そうでなければ、以下の条件、すなわち、btSplitがSPLIT_BT_HORに等しい、x0+cbWidthがSupPicRightBorderInPicより大きい、およびy0+cbHeightがSubPicBottomBorderInPic以下である、のすべてが真であるならば、allowBtSplitはFALSEに等しく設定される。そうでなければ、以下の条件、すなわち、mttDepthが0より大きい、partIdxが1に等しい、およびMttSplitMode[x0][y0][mttDepth-1]がparallelTtSplitに等しい、のすべてが真であるならば、allowBtSplitはFALSEに等しく設定される。そうでなければ、以下の条件、すなわち、btSplitがSPLIT_BT_VERに等しい、cbWidthがMaxTbSizeY以下である、およびcbHeightがMaxTbSizeYより大きい、のすべてが真であるならば、allowBtSplitはFALSEに等しく設定される。そうでなければ、以下の条件、すなわち、btSplitがSPLIT_BT_HORに等しい、cbWidthがMaxTbSizeYより大きい、およびcbHeightがMaxTbSizeY以下である、のすべてが真であるならば、allowBtSplitはFALSEに等しく設定される。そうでなければ、allowBtSplitはTRUEに等しく設定される。

【0121】

許容される三分割プロセスは、以下のように定義され得る。このプロセスへの入力、三分割モードttSplit、コーディングブロック幅cbWidth、コーディングブロック高さcbHeight、ピクチャの左上ルマサンプルに対する考慮されるコーディングブロックの左上ル

マサンプルの位置(x0,y0)、マルチタイプ木深度mttDepth、オフセットを有する最大マルチタイプ木深度maxMttDepth、および最大二分木サイズmaxTtSizeである。このプロセスの出力は、変数allowTtSplitである。

【 0 1 2 2 】

【表 2】

	ttSplit == SPLIT_TT_VER	ttSplit == SPLIT_TT_HOR
cbSize	cbWidth	cbHeight

ttSplit に基づく cbSize の仕様

10

【 0 1 2 3 】

変数cbSizeは上で指定されたように導出される。変数allowTtSplitは以下のように導出される。以下の条件、すなわち、cbSizeが $2 * \text{MinTtSizeY}$ 以下である、cbWidthが $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$ より大きい、cbHeightが $\text{Min}(\text{MaxTbSizeY}, \text{maxTtSize})$ より大きい、mttDepthがmaxMttDepth以上である、 $x0 + \text{cbWidth}$ がSupPicRightBoderInPicより大きい、および $y0 + \text{cbHeight}$ がSubPicBottomBorderInPicより大きい、のうちの1つまたは複数が真であるならば、allowTtSplitはFALSEに等しく設定される。そうでなければ、allowTtSplitはTRUEに等しく設定される。

【 0 1 2 4 】

シーケンス・パラメータ・セットRBSPのシンタックスおよびセマンティクスは、以下の通りである。

【 0 1 2 5 】

20

30

40

50

【表 3】

seq_parameter_set_rbsp() {	記述子
sps_seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
num_subpic_minus1	ue(v)
subpic_id_len_minus1	ue(v)
for (i = 0; i <= num_subpic_minus1; i++) {	
subpic_id[i]	u(v)
if(num_subpic_minus1 > 0) {	
subpic_level_idc[i]	u(8)
subpic_x_offset[i]	ue(v)
subpic_y_offset[i]	ue(v)
subpic_width_in_luma_samples[i]	ue(v)
subpic_height_in_luma_samples[i]	ue(v)
subpic_motion_constrained_flag[i]	u(1)
}	
}	
...	
}	

10

20

【 0 1 2 6 】

30

pic_width_in_luma_samplesは、ルマサンプルの単位で各々のデコードされたピクチャの幅を指定する。pic_width_in_luma_samplesは、0に等しくないものとし、MinCbSizeYの整数倍であるものとする。pic_height_in_luma_samplesは、ルマサンプルの単位で各々のデコードされたピクチャの高さを指定する。pic_height_in_luma_samplesは、0に等しくないものとし、MinCbSizeYの整数倍であるものとする。num_subpicture_minus1に1を足したものは、コーディングされたビデオシーケンスに属するコーディングされたピクチャ内で区分されるサブピクチャの数を指定する。subpic_id_len_minus1に1を足したものは、SPS内のシンタックス要素subpic_id[i]、SPSを参照するSPS内のspps_subpic_id、およびSPSを参照するタイルグループヘッダ内のtile_group_subpic_idを表現するために使用されるビットの数を指定する。subpic_id_len_minus1の値は、両端を含めてCeil(Log2(num_subpic_minus1+2))から8の範囲内であるものとする。subpic_id[i]は、SPSを参照するピクチャのi番目のサブピクチャのサブピクチャIDを指定する。subpic_id[i]の長さは、subpic_id_len_minus1+1ビットである。subpic_id[i]の値は0より大きいものとする。subpic_level_idc[i]は、i番目のサブピクチャの抽出から結果として生じたCVSが、指定されたリソース要件に準拠するレベルを示す。ビットストリームは、指定されるもの以外のsubpic_level_idc[i]の値を含まないものとする。subpic_level_idc[i]の他の値は予約される。存在しないとき、subpic_level_idc[i]の値は、general_level_idcの値に等しいと推測される。

40

【 0 1 2 7 】

subpic_x_offset[i]は、ピクチャの左上の角に対して、i番目のサブピクチャの左上の

50

角の水平オフセットを指定する。存在しないとき、subpic_x_offset[i]の値は0に等しいと推測される。サブピクチャxのオフセットの値は、SubpictureXOffset[i]=subpic_x_offset[i]のように導出される。subpic_y_offset[i]は、ピクチャの左上の角に対して、i番目のサブピクチャの左上の角の垂直オフセットを指定する。存在しないとき、subpic_y_offset[i]の値は、0に等しいと推測される。サブピクチャyのオフセットの値は、SubpictureYOffset[i]=subpic_y_offset[i]のように導出される。subpic_width_in_luma_samples[i]は、このSPSがアクティブなSPSであるi番目のデコードされたサブピクチャの幅を指定する。SubpictureXOffset[i]とsubpic_width_in_luma_samples[i]の和がpic_width_in_luma_samplesより小さいとき、subpic_width_in_luma_samples[i]の値はCtbSizeYの整数倍であるものとする。存在しないとき、subpic_width_in_luma_samples[i]の値は、pic_width_in_luma_samplesの値に等しいと推測される。subpic_height_in_luma_samples[i]は、このSPSがアクティブなSPSであるi番目のデコードされたサブピクチャの高さを指定する。SubpictureYOffset[i]とsubpic_height_in_luma_samples[i]の和がpic_height_in_luma_samplesより小さいとき、subpic_height_in_luma_samples[i]の値はCtbSizeYの整数倍であるものとする。存在しないとき、subpic_height_in_luma_samples[i]の値は、pic_height_in_luma_samplesの値に等しいと推測される。

【0128】

サブピクチャの和集合が重複および間隙なしでピクチャのエリア全体を包含するものであることが、ビットストリーム準拠の要件である。1に等しいsubpic_motion_constrained_flag[i]は、i番目のサブピクチャが時間動き制約サブピクチャであることを指定する。0に等しいsubpic_motion_constrained_flag[i]は、i番目のサブピクチャが時間動き制約サブピクチャであってもなくてもよいことを指定する。存在しないとき、subpic_motion_constrained_flagの値は0に等しいと推測される。

【0129】

変数SubpicWidthInCtbsY、SubpicHeightInCtbsY、SubpicSizeInCtbsY、SubpicWidthInMinCbsY、SubpicHeightInMinCbsY、SubpicSizeInMinCbsY、SubpicSizeInSamplesY、SubpicWidthInSamplesC、およびSubpicHeightInSamplesCは、以下のように導出される。

```
SubpicWidthInLumaSamples[i]=subpic_width_in_luma_samples[i]
SubpicHeightInLumaSamples[i]=subpic_height_in_luma_samples[i]
SubPicRightBorderInPic[i]=SubpictureXOffset[i]+PicWidthInLumaSamples[i]
SubPicBottomBorderInPic[i]=SubpictureYOffset[i]+PicHeightInLumaSamples[i]
SubpicWidthInCtbsY[i]=Ceil(SubpicWidthInLumaSamples[i]÷CtbSizeY)
SubpicHeightInCtbsY[i]=Ceil(SubpicHeightInLumaSamples[i]÷CtbSizeY)
SubpicSizeInCtbsY[i]=SubpicWidthInCtbsY[i]*SubpicHeightInCtbsY[i]
SubpicWidthInMinCbsY[i]=SubpicWidthInLumaSamples[i]/MinCbSizeY
SubpicHeightInMinCbsY[i]=SubpicHeightInLumaSamples[i]/MinCbSizeY
SubpicSizeInMinCbsY[i]=SubpicWidthInMinCbsY[i]*SubpicHeightInMinCbsY[i]
SubpicSizeInSamplesY[i]=SubpicWidthInLumaSamples[i]*SubpicHeightInLumaSamples[i]
SubpicWidthInSamplesC[i]=SubpicWidthInLumaSamples[i]/SubWidthC
SubpicHeightInSamplesC[i]=SubpicHeightInLumaSamples[i]/SubHeightC
```

【0130】

サブピクチャ・パラメータ・セットRBSPのシンタックスおよびセマンティクスは、以下の通りである。

【0131】

10

20

30

40

50

【表 4】

sub_pic_parameter_set_rbsp() {	記述子
spps_subpic_id	u(v)
spps_subpic_parameter_set_id	ue(v)
spps_seq_parameter_set_id	ue(v)
single_tile_in_subpic_flag	u(1)
if(!single_tile_in_subpic_flag) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_tile_spacing_flag	u(1)
if(!uniform_tile_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
tile_column_width_minus1[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
tile_row_height_minus1[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
}	
if(loop_filter_across_tiles_enabled_flag)	
loop_filter_across_subpic_enabled_flag	u(1)
rbsp_trailing_bits()	
}	

10

20

30

【 0 1 3 2 】

spps_subpic_idは、SPPSが属するサブピクチャを識別する。spps_subpic_idの長さは、subpic_id_len_minus1+1ビットである。spps_subpic_parameter_set_idは、他のシンタックス要素による参照のためにSPPSを識別する。spps_subpic_parameter_set_idの値は、両端を含めて0から63の範囲内であるものとする。spps_seq_parameter_set_idは、アクティブなSPSについてsps_seq_parameter_set_idの値を指定する。spps_seq_parameter_set_idの値は、両端を含めて0から15の範囲内であるものとする。1に等しいsingle_tile_in_subpic_flagは、SPPSを参照する各サブピクチャ内に1つのみのタイルがあることを指定する。0に等しいsingle_tile_in_subpic_flagは、SPPSを参照する各サブピクチャ内に1つより多くのタイルがあることを指定する。num_tile_columns_minus1に1を足したものは、サブピクチャを区分するタイル列の数を指定する。num_tile_columns_minus1は、両端を含めて0からPicWidthInCtbsY[spps_subpic_id]-1の範囲内であるものとする。存在しないとき、num_tile_columns_minus1の値は0に等しいと推測される。num_tile_rows_minus1に1を足したものは、サブピクチャを区分するタイル行の数を指定する。num_tile_rows_minus1は、両端を含めて0からPicHeightInCtbsY[spps_subpic_id]-1の範囲内であるものとする。存在しないとき、num_tile_rows_minus1の値は0に等しいと推測される。変数NumTilesInPicは、(num_tile_columns_minus1+1)*(num_tile_rows_minus1+1)に等しく設定される。

40

50

【0133】

single_tile_in_subpic_flagが0に等しいとき、NumTilesInPicは0より大きいものとする。1に等しいuniform_tile_spacing_flagは、タイル列の境界と、同様にタイル行の境界とが、サブピクチャにわたって均一に分散されることを指定する。0に等しいuniform_tile_spacing_flagは、タイル列の境界と、同様にタイル行の境界とが、サブピクチャにわたって均一に分散されないが、シンタックス要素tile_column_width_minus1[i]およびtile_row_height_minus1[i]を使用して明示的にシグナリングされることを指定する。存在しないとき、uniform_tile_spacing_flagの値は1に等しいと推測される。tile_column_width_minus1[i]に1を足したものは、CTBの単位でi番目のタイル列の幅を指定する。tile_row_height_minus1[i]に1を足したものは、CTBの単位でi番目のタイル行の高さを指定する。

10

【0134】

以下の変数、すなわち、CTBの単位でi番目のタイル列の幅を指定する、両端を含めて0からnum_tile_columns_minus1の範囲にわたるiについてのリストColWidth[i]、CTBの単位でj番目のタイル行の高さを指定する、両端を含めて0からnum_tile_rows_minus1の範囲にわたるjについてのリストRowHeight[j]、CTBの単位でi番目のタイル列境界の位置を指定する、両端を含めて0からnum_tile_columns_minus1+1の範囲にわたるiについてのリストColBd[i]、CTBの単位でj番目のタイル行境界の位置を指定する、両端を含めて0からnum_tile_rows_minus1+1の範囲にわたるjについてのリストRowBd[j]、ピクチャのCTBラスタ走査におけるCTBアドレスからタイル走査におけるCTBアドレスへの変換を指定する、両端を含めて0からPicSizeInCtbsY-1の範囲にわたるctbAddrRsについてのリストCtbAddrRsToTs[ctbAddrRs]、タイル走査におけるCTBアドレスからピクチャのCTBラスタ走査におけるCTBアドレスへの変換を指定する、両端を含めて0からPicSizeInCtbsY-1の範囲にわたるctbAddrTsについてのリストCtbAddrTsToRs[ctbAddrTs]、タイル走査におけるCTBアドレスからタイルIDへの変換を指定する、両端を含めて0からPicSizeInCtbsY-1の範囲にわたるctbAddrTsについてのリストTileId[ctbAddrTs]、タイルインデックスからタイル内のCTUの数への変換を指定する、両端を含めて0からPicSizeInCtbsY-1の範囲にわたるtileIdxについてのリストNumCtusInTile[tileIdx]、タイルIDからタイル内の最初のCTBのタイル走査におけるCTBアドレスへの変換を指定する、両端を含めて0からNumTilesInPic-1の範囲にわたるtileIdxについてのリストFirstCtbAddrTs[tileIdx]、ルマサンプルの単位でi番目のタイル列の幅を指定する、両端を含めて0からnum_tile_columns_minus1の範囲にわたるiについてのリストColumnWidthInLumaSamples[i]、および、ルマサンプルの単位でj番目のタイル行の高さを指定する、両端を含めて0からnum_tile_rows_minus1の範囲にわたるjについてのリストRowHeightInLumaSamples[j]は、CTBラスタおよびタイル走査変換プロセスを呼び出すことによって導出される。両端を含めて0からnum_tile_columns_minus1の範囲にわたるiについてのColumnWidthInLumaSamples[i]、および両端を含めて0からnum_tile_rows_minus1の範囲にわたるjについてのRowHeightInLumaSamples[j]の値は、すべて、0より大きいものとする。

20

30

【0135】

1に等しいloop_filter_across_tiles_enabled_flagは、ループ内フィルタリング動作がSPPSを参照するサブピクチャ内のタイル境界にわたって実行され得ることを指定する。0に等しいloop_filter_across_tiles_enabled_flagは、ループ内フィルタリング動作がSPPSを参照するサブピクチャ内のタイル境界にわたって実行されないことを指定する。ループ内フィルタリング動作は、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動作を含む。存在しないとき、loop_filter_across_tiles_enabled_flagの値は1に等しいと推測される。1に等しいloop_filter_across_subpic_enabled_flagは、ループ内フィルタリング動作がSPPSを参照するサブピクチャ内のサブピクチャ境界にわたって実行され得ることを指定する。0に等しいloop_filter_across_subpic_enabled_flagは、ループ内フィルタリング動作がSPPSを参照するサブピクチャ内

40

50

のサブピクチャ境界にわたって実行されないことを指定する。ループ内フィルタリング動作は、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動作を含む。存在しないとき、`loop_filter_across_subpic_enabled_flag`の値は、`loop_filter_across_tiles_enabled_flag`の値に等しいと推測される。

【0 1 3 6】

一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クスは、以下の通りである。

【0 1 3 7】

【表 5】

<code>tile_group_header() {</code>	記述子
<code>tile_group_subpic_id</code>	<code>u(v)</code>
<code>tile_group_subpic_parameter_set_id</code>	<code>u(v)</code>
<code>...</code>	
<code>}</code>	

10

【0 1 3 8】

タイルグループヘッダシンタックス要素`tile_group_pic_parameter_set_id`および`tile_group_pic_order_cnt_lsb`の値は、コーディングされたピクチャのすべてのタイルグループヘッダ内で同じであるものとする。タイルグループヘッダシンタックス要素`tile_group_subpic_id`の値は、コーディングされたサブピクチャのすべてのタイルグループヘッダ内で同じであるものとする。`tile_group_subpic_id`は、タイルグループが属するサブピクチャを識別する。`tile_group_subpic_id`の長さは、`subpic_id_len_minus1+1`ビットである。`tile_group_subpic_parameter_set_id`は、使用中のSPPSについての`spps_subpic_parameter_set_id`の値を指定する。`tile_group_spps_parameter_set_id`の値は、両端を含めて0から63の範囲内であるものとする。

20

【0 1 3 9】

以下の変数が導出され、アクティブなSPSから導出されたそれぞれの変数を上書きする。

```

PicWidthInLumaSamples=SubPicWidthInLumaSamples[tile_group_subpic_id]
PicHeightInLumaSamples=PicHeightInLumaSamples[tile_group_subpic_id]
SubPicRightBorderInPic=SubPicRightBorderInPic[tile_group_subpic_id]
SubPicBottomBorderInPic=SubPicBottomBorderInPic[tile_group_subpic_id]
PicWidthInCtbsY=SubPicWidthInCtbsY[tile_group_subpic_id]
PicHeightInCtbsY=SubPicHeightInCtbsY[tile_group_subpic_id]
PicSizeInCtbsY=SubPicSizeInCtbsY[tile_group_subpic_id]
PicWidthInMinCbsY=SubPicWidthInMinCbsY[tile_group_subpic_id]
PicHeightInMinCbsY=SubPicHeightInMinCbsY[tile_group_subpic_id]
PicSizeInMinCbsY=SubPicSizeInMinCbsY[tile_group_subpic_id]
PicSizeInSamplesY=SubPicSizeInSamplesY[tile_group_subpic_id]
PicWidthInSamplesC=SubPicWidthInSamplesC[tile_group_subpic_id]
PicHeightInSamplesC=SubPicHeightInSamplesC[tile_group_subpic_id]

```

30

40

【0 1 4 0】

コーディング・ツリー・ユニット・シンタックスは、以下の通りである。

【0 1 4 1】

50

【表 6】

coding_tree_unit() {	記述子
xCtb = (CtbAddrInRs % PicWidthInCtbsY) << CtbLog2SizeY + SubpictureXOffset	
yCtb = (CtbAddrInRs / PicWidthInCtbsY) << CtbLog2SizeY + SubpictureYOffset	
...	
}	

10

【 0 1 4 2 】

【表 7】

dual_tree_implicit_qt_split(x0, y0, log2CbSize, cqtDepth) {	記述子
if(log2CbSize > 6) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
dual_tree_implicit_qt_split(x0, y0, log2CbSize - 1, cqtDepth + 1)	
if(x1 < SubPicRightBorderInPic)	
dual_tree_implicit_qt_split(x1, y0, log2CbSize - 1, cqtDepth + 1)	
if(y1 < SubPicBottomBorderInPic)	
dual_tree_implicit_qt_split(x0, y1, log2CbSize - 1, cqtDepth + 1)	
if(x1 < SubPicRightBorderInPic && y1 < SubPicBottomBorderInPic)	
dual_tree_implicit_qt_split(x1, y1, log2CbSize - 1, cqtDepth + 1)	
} else {	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_LUMA)	
coding_quadtree(x0, y0, log2CbSize, cqtDepth, DUAL_TREE_CHROMA)	
}	
}	

20

30

40

【 0 1 4 3 】

コーディング四分木のシンタックスおよびセマンティクスは、以下の通りである。

【 0 1 4 4 】

50

【表 8】

coding_quadtree(x0, y0, log2CbSize, cqtDepth, treeType) {	記述子
minQtSize = (treeType == DUAL_TREE_CHROMA) ? MinQtSizeC : MinQtSizeY	
maxBtSize = (treeType == DUAL_TREE_CHROMA) ? MaxBtSizeC : MaxBtSizeY	
if(((x0 + (1 << log2CbSize) <= PicWidthInLumaSamples) ? 1 : 0) + ((y0 + (1 << log2CbSize) <= PicHeightInLumaSamples) ? 1 : 0) + (((1 << log2CbSize) <= maxBtSize) ? 1 : 0)) >= 2 && (1 << log2CbSize) > minQtSize)	10
qt_split_cu_flag[x0][y0]	ae(v)
if(cu_qp_delta_enabled_flag && cqtDepth <= diff_cu_qp_delta_depth) {	
IsCuQpDeltaCoded = 0	
CuQpDeltaVal = 0	
CuQgTopLeftX = x0	20
CuQgTopLeftY = y0	
}	
if(qt_split_cu_flag[x0][y0]) {	
x1 = x0 + (1 << (log2CbSize - 1))	
y1 = y0 + (1 << (log2CbSize - 1))	
coding_quadtree(x0, y0, log2CbSize - 1, cqtDepth + 1, treeType)	
if(x1 < SubPicRightBorderInPic)	
coding_quadtree(x1, y0, log2CbSize - 1, cqtDepth + 1, treeType)	
if(y1 < SubPicBottomBorderInPic)	30
coding_quadtree(x0, y1, log2CbSize - 1, cqtDepth + 1, treeType)	
if(x1 < SubPicRightBorderInPic && y1 < SubPicBottomBorderInPic)	
coding_quadtree(x1, y1, log2CbSize - 1, cqtDepth + 1, treeType)	
} else	
multi_type_tree(x0, y0, 1 << log2CbSize, 1 << log2CbSize, cqtDepth, 0, 0, 0, treeType)	
}	40

【 0 1 4 5 】

qt_split_cu_flag[x0][y0]は、コーディング・ユニットが半分の水平および垂直サイズを有するコーディング・ユニットへと分割されるかどうかを指定する。アレイインデックスx0、y0は、ピクチャの左上ルマサンプルに対して、考慮されるコーディングブロックの左上ルマサンプルの位置(x0,y0)を指定する。qt_split_cu_flag[x0][y0]が存在しないとき、以下が当てはまる。以下の条件のうちの1つまたは複数が真であるならば、qt_split_cu_flag[x0][y0]の値は1に等しいと推測される。treeTypeがDUAL_TREE_CHROMAに等しいか、またはそうでなければMaxBtSizeYより大きいならば、x0+(1 << log2CbSize)がSubPicRightBorderInPicより大きく、(1 << log2CbSize)がMaxBtSizeCより大き

10

20

30

40

50

い。treeTypeがDUAL_TREE_CHROMAに等しいか、またはそうでなければMaxBtSizeYより大きいならば、 $y0 + (1 \ll \log2CbSize)$ がSubPicBottomBorderInPicより大きく、 $(1 \ll \log2CbSize)$ がMaxBtSizeCより大きい。

【0146】

そうでなければ、以下の条件のすべてが真であるならば、qt_split_cu_flag[x0][y0]の値は1に等しいと推測される。treeTypeがDUAL_TREE_CHROMAに等しいか、またはそうでなければMinQtSizeYより大きいならば、 $x0 + (1 \ll \log2CbSize)$ がSubPicRightBorderInPicより大きく、 $y0 + (1 \ll \log2CbSize)$ がSubPicBottomBorderInPicより大きく、 $(1 \ll \log2CbSize)$ がMinQtSizeCより大きい。そうでなければ、qt_split_cu_flag[x0][y0]の値は0に等しいと推測される。

10

【0147】

マルチタイプ木のシンタックスおよびセマンティクスは、以下の通りである。

【0148】

20

30

40

50

【表 9 A】

multi_type_tree(x0, y0, cbWidth, cbHeight, cqtDepth, mttDepth, depthOffset, partIdx, treeType) {	記述子	
<pre> if((allowSplitBtVer allowSplitBtHor allowSplitTtVer allowSplitTtHor) && (x0 + cbWidth <= SubPicRightBorderInPic) && (y0 + cbHeight <= SubPicBottomBorderInPic)) </pre>		10
mtt_split_cu_flag	ae(v)	
<pre> if(cu_qp_delta_enabled_flag && (cqtDepth + mttDepth) <= diff_cu_qp_delta_depth) { </pre>		
IsCuQpDeltaCoded = 0		
CuQpDeltaVal = 0		
CuQgTopLeftX = x0		
CuQgTopLeftY = y0		
}		20
if(mtt_split_cu_flag) {		
<pre> if((allowSplitBtHor allowSplitTtHor) && (allowSplitBtVer allowSplitTtVer)) </pre>		
mtt_split_cu_vertical_flag	ae(v)	
<pre> if((allowSplitBtVer && allowSplitTtVer && mtt_split_cu_vertical_flag) (allowSplitBtHor && allowSplitTtHor && !mtt_split_cu_vertical_flag)) </pre>		30
mtt_split_cu_binary_flag	ae(v)	
<pre> if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_VER) { </pre>		
depthOffset += (x0 + cbWidth > SubPicRightBorderInPic) ? 1 : 0		
x1 = x0 + (cbWidth / 2)		
multi_type_tree(x0, y0, cbWidth / 2, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)		40

10

20

30

40

50

【表 9 B】

if(x1 < SubPicRightBorderInPic)	
multi_type_tree(x1, y0, cbWidth / 2, cbHeightY, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_BT_HOR) {	
depthOffset += (y0 + cbHeight > SubPicBottomBorderInPic) ? 1 : 0	10
y1 = y0 + (cbHeight / 2)	
multi_type_tree(x0, y0, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)	
if(y1 < SubPicBottomBorderInPic)	
multi_type_tree(x0, y1, cbWidth, cbHeight / 2, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	20
} else if(MttSplitMode[x0][y0][mttDepth] == SPLIT_TT_VER) {	
x1 = x0 + (cbWidth / 4)	
x2 = x0 + (3 * cbWidth / 4)	
multi_type_tree(x0, y0, cbWidth / 4, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 0, treeType)	30
multi_type_tree(x1, y0, cbWidth / 2, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 1, treeType)	
multi_type_tree(x2, y0, cbWidth / 4, cbHeight, cqtDepth, mttDepth + 1, depthOffset, 2, treeType)	
} else { /* SPLIT_TT_HOR */	40

【表 9 C】

$y1 = y0 + (cbHeight / 4)$	
$y2 = y0 + (3 * cbHeight / 4)$	
<code>multi_type_tree(x0, y0, cbWidth, cbHeight / 4,</code> <code>cqtDepth, mttDepth + 1, depthOffset, 0, treeType)</code>	
<code>multi_type_tree(x0, y1, cbWidth, cbHeight / 2,</code> <code>cqtDepth, mttDepth + 1, depthOffset, 1, treeType)</code>	10
<code>multi_type_tree(x0, y2, cbWidth, cbHeight / 4,</code> <code>cqtDepth, mttDepth + 1, depthOffset, 2, treeType)</code>	
<code>}</code>	
<code>} else</code>	
<code>coding_unit(x0, y0, cbWidth, cbHeight, treeType)</code>	20
<code>}</code>	

【0149】

0に等しい`mtt_split_cu_flag`は、コーディング・ユニットが分割されないことを指定する。1に等しい`mtt_split_cu_flag`は、シンタックス要素`mtt_split_cu_binary_flag`によって示されるように、コーディング・ユニットが二分割を使用して2つのコーディング・ユニットへと、または三分割を使用して3つのコーディング・ユニットへと分割されることを指定する。二分割または三分割は、シンタックス要素`mtt_split_cu_vertical_flag`によって示されるように、垂直または水平のいずれかであることが可能である。`mtt_split_cu_flag`が存在しないとき、`mtt_split_cu_flag`の値は以下のように推測される。以下の条件、すなわち、 $x0 + cbWidth$ が`SubPicRightBorderInPic`より大きい、および $y0 + cbHeight$ が`SubPicBottomBorderInPic`より大きい、のうちの1つまたは複数が真であるならば、`mtt_split_cu_flag`の値は1に等しいと推測される。そうでなければ、`mtt_split_cu_flag`の値は0に等しいと推測される。

【0150】

時間的なルマ動きベクトル予測のための導出プロセスは、以下の通りである。このプロセスの出力は、1/16分数サンプル精度での動きベクトル予測`mvLXCol`、および利用可能性フラグ`availableFlagLXCol`である。変数`currCb`は、ルマ位置(xCb, yCb)における現在のルマコーディングブロックを指定する。変数`mvLXCol`および`availableFlagLXCol`は以下のように導出される。`tile_group_temporal_mvp_enabled_flag`が0に等しいならば、または参照ピクチャが現在のピクチャであるならば、`mvLXCol`の両方の成分が0に等しく設定され、`availableFlagLXCol`が0に等しく設定される。それ以外ならば(`tile_group_temporal_mvp_enabled_flag`が1に等しく、参照ピクチャが現在のピクチャではない)、以下の順序付けられたステップが適用される。右下の同じに位置付けられる動きベクトルは以下のように導出される。

$xColBr = xCb + cbWidth$ (8-355)

$yColBr = yCb + cbHeight$ (8-356)

【0151】

$yCb > CtbLog2SizeY$ が $yColBr > CtbLog2SizeY$ に等しく、 $yColBr$ が`SubPicBott`

10

20

30

40

50

omBorderInPicより小さく、xColBrがSubPicRightBorderInPicより小さいならば、以下が当てはまる。変数colCbは、ColPicによって指定される同じに位置付けられるピクチャ内部の $((xColBr) \gg 3) \ll 3, (yColBr) \gg 3) \ll 3$ によって与えられる修正された位置を包含するルマコーディングブロックを指定する。ルマ位置(xColCb,yColCb)は、ColPicによって指定される同じに位置付けられるピクチャの左上ルマサンプルに対して、colCbによって指定される同じに位置付けられるルマコーディングブロックの左上サンプルに等しく設定される。同じに位置付けられる動きベクトルのための導出プロセスは、0に等しく設定されたcurrCb、colCb、(xColCb,yColCb)、refIdxLX、およびsbFlagを入力として用いて呼び出され、出力はmvLXColおよびavailableFlagLXColに割り当てられる。そうでなければ、mvLXColの両方の成分が0に等しく設定され、availableFlagLXColが0に等しく設定される。

10

【0152】

時間的な三角形マージ候補のための導出プロセスは、以下の通りである。変数mvLXColC0、mvLXColC1、availableFlagLXColC0、およびavailableFlagLXColC1は、以下のように導出される。tile_group_temporal_mvp_enabled_flagが0に等しいならば、mvLXColC0とmvLXColC1の両方の成分が0に等しく設定され、availableFlagLXColC0およびavailableFlagLXColC1は0に等しく設定される。そうでなければ(tile_group_temporal_mvp_enabled_flagが1に等しい)、以下の順序付けられたステップが適用される。右下の同じに位置付けられる動きベクトルmvLXColC0は、以下のように導出される。

xColBr=xCb+cbWidth

(8-392)

20

yColBr=yCb+cbHeight

(8-393)

【0153】

yCb) > CtbLog2SizeYがyColBr) > CtbLog2SizeYに等しく、yColBrがSubPicBottomBorderInPicより小さく、xColBrがSubPicRightBorderInPicより小さいならば、以下が当てはまる。変数colCbは、ColPicによって指定される同じに位置付けられるピクチャの内部の $((xColBr) \gg 3) \ll 3, (yColBr) \gg 3) \ll 3$ によって与えられる修正された位置を包含するルマコーディングブロックを指定する。ルマ位置(xColCb,yColCb)は、ColPicによって指定される同じに位置付けられるピクチャの左上ルマサンプルに対して、colCbによって指定される同じに位置付けられるルマコーディングブロックの左上サンプルに等しく設定される。同じに位置付けられる動きベクトルのための導出プロセスは、0に等しく設定されたcurrCb、colCb、(xColCb,yColCb)、refIdxLXC0、およびsbFlagを入力として用いて呼び出され、出力はmvLXColC0およびavailableFlagLXColC0に割り当てられる。そうでなければ、mvLXColC0の両方の成分が0に等しく設定され、availableFlagLXColC0が0に等しく設定される。

30

【0154】

構築されたアフィン制御点動きベクトルマージ候補のための導出プロセスは、以下の通りである。Xを0および1として、第4の(同じに位置付けられる右下の)制御点動きベクトルcpMvLXCorner[3]、参照インデックスrefIdxLXCorner[3]、予測リスト利用フラグpredFlagLXCorner[3]、および利用可能性フラグavailableFlagCorner[3]が以下のように導出される。Xを0または1として、時間マージ候補についての参照インデックスrefIdxLXCorner[3]が0に等しく設定される。Xを0または1として、変数mvLXColおよびavailableFlagLXColが以下のように導出される。tile_group_temporal_mvp_enabled_flagが0に等しいならば、mvLXColの両方の成分が0に等しく設定され、availableFlagLXColが0に等しく設定される。そうでなければ(tile_group_temporal_mvp_enabled_flagが1に等しい)、以下が当てはまる。

40

xColBr=xCb+cbWidth

(8-566)

yColBr=yCb+cbHeight

(8-567)

【0155】

yCb) > CtbLog2SizeYがyColBr) > CtbLog2SizeYに等しく、yColBrがSubPicBottomBorderInPicより小さく、xColBrがSubPicRightBorderInPicより小さいならば、以

50

下が当てはまる。変数colCbは、ColPicによって指定される同じに位置付けられるピクチャの内部の $((xColBr) \gg 3) \ll (3, (yColBr) \gg 3) \ll (3)$ によって与えられる修正された位置を包含するルマコーディングブロックを指定する。ルマ位置 $(xColCb, yColCb)$ は、ColPicによって指定される同じに位置付けられるピクチャの左上ルマサンプルに対して、colCbによって指定される同じに位置付けられるルマコーディングブロックの左上サンプルに等しく設定される。同じに位置付けられる動きベクトルのための導出プロセスは、0に等しく設定されたcurrCb、colCb、 $(xColCb, yColCb)$ 、refIdxLX、およびsbFlagを入力として用いて呼び出され、出力はmvLXColおよびavailableFlagLXColに割り当てられる。そうでなければ、mvLXColの両方の成分が0に等しく設定され、availableFlagLXColが0に等しく設定される。pic_width_in_luma_samplesのすべての出現を、PicWidthInLumaSamplesで置換する。pic_height_in_luma_samplesのすべての出現を、PicHeightInLumaSamplesで置換する。

【0156】

第2の例の実施形態では、シーケンス・パラメータ・セットRBSPのシンタックスおよびセマンティクスは、以下の通りである。

【0157】

【表10】

seq_parameter_set_rbsp() {	記述子
sps_seq_parameter_set_id	ue(v)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
num_subpic_minus1	ue(v)
subpic_id_len_minus1	ue(v)
for (i = 0; i <= num_subpic_minus1; i++) {	
subpic_id[i]	u(v)
if(num_subpic_minus1 > 0) {	
subpic_level_idc[i]	u(8)
subpic_x_offset[i]	ue(v)
subpic_y_offset[i]	ue(v)
subpic_width_in_luma_samples[i]	ue(v)
subpic_height_in_luma_samples[i]	ue(v)
}	
}	
...	
}	

【0158】

subpic_id_len_minus1に1を足したものは、SPS内のシンタックス要素subpic_id[i]を表現するために使用されるビットの数、SPSを参照するSPPS内のspps_subpic_id、およびSPSを参照するタイルグループヘッダ内のtile_group_subpic_idを指定する。subpic_id_len_minus1の値は、両端を含めてCeil(Log2(num_subpic_minus1+3))から8の範囲内であるものとする。両端を含めて0からnum_subpic_minus1のiについてのsubpicture[i]の間で重複がないものとする。ビットストリーム準拠の要件である。各サ

ブピクチャは、時間動き制約サブピクチャであり得る。

【0159】

一般的なタイルグループヘッダのセマンティクスは、以下の通りである。tile_group_subpic_idは、タイルグループが属するサブピクチャを識別する。tile_group_subpic_idの長さは、subpic_id_len_minus1+1ビットである。1に等しいtile_group_subpic_idは、タイルグループがいずれのサブピクチャにも属さないことを示す。

【0160】

第3の例の実施形態では、NALユニットヘッダのシンタックスとセマンティクスは、以下の通りである。

【0161】

【表11】

nal_unit_header() {	記述子
forbidden_zero_bit	f(1)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
nuh_subpicture_id_len	u(4)
nuh_reserved_zero_4bits	u(3)
}	

【0162】

nuh_subpicture_id_lenは、サブピクチャIDを指定するシンタックス要素を表現するために使用されるビットの数を指定する。nuh_subpicture_id_lenの値が0より大きいとき、nuh_reserved_zero_4bits内の後の最初のnuh_subpicture_id_len番目のビットは、NALユニットのペイロードが属するサブピクチャのIDを指定する。nuh_subpicture_id_lenが0より大きいとき、nuh_subpicture_id_lenの値は、アクティブなSPS内のsubpic_id_len_minus1の値に等しいものとする。非VCL NALユニットについてのnuh_subpicture_id_lenの値は、以下のように制約される。nal_unit_typeがSPS_NUTまたはPPS_NUTに等しいならば、nuh_subpicture_id_lenは0に等しいものとする。nuh_reserved_zero_3bitsは「000」に等しいものとする。デコーダは、nuh_reserved_zero_3bitsの値が「000」に等しくないNALユニットを無視する(たとえば、ビットストリームから取り除いて廃棄する)ものとする。

【0163】

第4の例の実施形態では、サブピクチャ・ネスティング・シンタックスは、以下の通りである。

【0164】

10

20

30

40

50

【表 1 2】

sub-picture_nesting(payloadSize) {	記述子
all_sub_pictures_flag	u(1)
if(!all_sub_pictures_flag) {	
nesting_num_sub_pictures_minus1	ue(v)
for(i = 0; i <= nesting_num_sub_pictures_minus1; i++)	
nesting_sub_picture_id[i]	u(v)
}	
while(!byte_aligned())	
sub_picture_nesting_zero_bit /*0 に等しい*/	u(1)
do	
sei_message()	
while(more_rbsp_data())	
}	

10

20

【0 1 6 5】

1に等しいall_sub_pictures_flagは、ネストされたSEIメッセージがすべてのサブピクチャに適用されることを指定する。1に等しいall_sub_pictures_flagは、ネストされたSEIメッセージが適用されるサブピクチャが後続のシンタックス要素によって明示的にシグナリングされることを指定する。nesting_num_sub_pictures_minus1に1を足したものは、ネストされたSEIメッセージが適用されるサブピクチャの数を指定する。nesting_sub_picture_id[i]は、ネストされたSEIメッセージが適用されるi番目のサブピクチャのサブピクチャIDを示す。nesting_sub_picture_id[i]シンタックス要素は、Ceil(Log2(nesting_num_sub_pictures_minus1+1))ビットによって表現される。sub_picture_nesting_zero_bitは0に等しいものとする。

30

【0 1 6 6】

図9は、一例のビデオコーディングデバイス900の概略図である。ビデオコーディングデバイス900は、ここで説明されるような開示される例／実施形態を実装するために適している。ビデオコーディングデバイス900は、ネットワーク上でデータアップストリームおよび／またはダウンストリームを通信するための送信機および／または受信機を含む、ダウンストリームポート920、アップストリームポート950、および／またはトランシーバユニット(Tx/Rx)910を備える。ビデオコーディングデバイス900は、また、データを処理するための論理ユニットおよび／または中央処理ユニット(CPU)を含むプロセッサ930、およびデータを記憶するためのメモリ932を含む。ビデオコーディングデバイス900は、また、電気、光-電気(OE)コンポーネント、電気-光(EO)コンポーネント、および／または、電気、光、またはワイヤレス通信ネットワークを介したデータの通信のためにアップストリームポート950および／またはダウンストリームポート920に結合されたワイヤレス通信コンポーネントを備え得る。ビデオコーディングデバイス900は、また、ユーザへのおよびユーザからの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960を含み得る。I/Oデバイス960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ような出力デバイスを含み得る。I/Oデバイス960は、また、キーボード、マウス、トラックボールなどのような入力デバイス、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み得る。

40

50

【0167】

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)、およびデジタル信号プロセッサ(DSP)として実装され得る。プロセッサ930は、ダウンストリームポート920、Tx/Rx910、アップストリームポート950、およびメモリ932と通信する。プロセッサ930はコーディングモジュール914を備える。コーディングモジュール914は、ビットストリーム500、ピクチャ600、および／またはピクチャ800を利用し得る方法100、1000、1100、および／または機構700のような上で説明された開示された実施形態を実装する。コーディングモジュール914は、ここで説明される任意の他の方法／機構も実装し得る。さらに、コーディングモジュール914は、コーデックシステム200、エンコーダ300、および／またはデコーダ400を実装し得る。たとえば、コーディングモジュール914は、SPS内のサブピクチャの位置とサイズをシグナリングおよび／または取得するために利用され得る。別の例では、コーディングモジュール914は、そのようなサブピクチャがピクチャの右の境界またはピクチャの下の境界にそれぞれ配置されない限り、CTUサイズの倍数であるようにサブピクチャ幅およびサブピクチャ高さを制約し得る。別の例では、コーディングモジュール914は、間隙または重複なしでピクチャを包含するようにサブピクチャを制約し得る。別の例では、コーディングモジュール914は、いくつかのサブピクチャが時間動き制約サブピクチャであり、他のサブピクチャがそうでないことを示すデータを、シグナリングおよび／または取得するために利用され得る。別の例では、コーディングモジュール914は、SPS内でサブピクチャIDの完全なセットをシグナリングし、対応するスライスを含むサブピクチャを示すためにサブピクチャIDを各スライスヘッダに含め得る。別の例では、コーディングモジュール914は、各サブピクチャについてのレベルをシグナリングし得る。それゆえ、コーディングモジュール914は、ビデオコーディングデバイス900に、追加の機能を提供させ、ビデオデータを区分してコーディングするときに、処理オーバーヘッドを低減し、および／またはコーディング効率を増加させるために、ある処理を回避させる。従って、コーディングモジュール914は、ビデオコーディングの技術に特有である課題に対処するとともにビデオコーディングデバイス900の機能を改善する。さらに、コーディングモジュール914は、異なる状態へのビデオコーディングデバイス900の変換をもたらす。代替的に、コーディングモジュール914は、メモリ932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(たとえば、非一時的媒体に記憶されたコンピュータプログラム製品として)実装されることが可能である。

【0168】

メモリ932は、ディスク、テープドライブ、ソリッドステートドライブ、リードオンリメモリ(ROM)、ランダムアクセスメモリ(RAM)、フラッシュメモリ、三値連想メモリ(TCAM)、スタティックランダムアクセスメモリ(SRAM)などのような1つまたは複数のメモリタイプを備える。メモリ932は、プログラムを、実行のためにそのようなプログラムが選択されるときに記憶するために、およびプログラム実行の間に読み取られる命令およびデータを記憶するために、オーバーフローデータ記憶デバイスとして使用され得る。

【0169】

図10は、サブピクチャ522、523、622、722、および／または822のようなサブピクチャの抽出をサポートするために、ピクチャの、ビットストリーム500のようなビットストリーム内にサブピクチャ・レイアウトをエンコードする一例の方法1000のフローチャートである。方法1000は、方法100を実行するとき、コーデックシステム200、エンコーダ300、および／またはビデオコーディングデバイス900のようなエンコーダによって利用され得る。

【0170】

方法1000は、エンコーダが複数のピクチャを含むビデオシーケンスを受信し、たとえばユーザ入力に基づいてそのビデオシーケンスをビットストリームへとエンコードすることを決定するとき、開始し得る。ビデオシーケンスは、エンコードの前にさらなる区分の

ためにピクチャ／画像／フレームへと区分される。ステップ1001において、ピクチャは、サブピクチャとして以後表記される現在のサブピクチャを含む複数のサブピクチャへと区分される。ステップ1003において、サブピクチャはビットストリームへとエンコードされる。

【0171】

ステップ1005において、サブピクチャのサブピクチャサイズおよびサブピクチャ位置が、ビットストリーム内のSPSへとエンコードされる。サブピクチャ位置は、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセット距離を含む。サブピクチャサイズは、ルマサンプルにおけるサブピクチャ高さおよびルマサンプルにおけるサブピクチャ幅を含む。サブピクチャが動き制約サブピクチャであることを示すために、フラグもSPS内にエンコードされ得る。そのような場合、サブピクチャサイズおよびサブピクチャ位置は、動き制約サブピクチャのレイアウトを示す。

【0172】

ステップ1007において、サブピクチャIDが、ピクチャから区分されたサブピクチャの各々についてSPSへとエンコードされる。ピクチャから区分されたサブピクチャの数も、SPSへとエンコードされ得る。ステップ1009において、ビットストリームがデコーダに向けた通信のために記憶される。ビットストリームは次いで、望まれるようにデコーダに向けて送信され得る。いくつかの例では、サブビットストリームは、エンコードされたビットストリームから抽出され得る。そのような場合、送信されるビットストリームはサブビットストリームである。他の例では、エンコードされたビットストリームは、デコーダにおけるサブビットストリームの抽出のために送信され得る。さらに他の例では、エンコードされたビットストリームは、サブビットストリームの抽出なしでデコードされ表示され得る。これらの例のいずれにおいても、サブピクチャのサイズ、位置、ID、数、および／または動き制約サブピクチャフラグが、サブピクチャ・レイアウトをデコーダに効率的にシグナリングするために使用され得る。

【0173】

図11は、シグナリングされたサブピクチャ・レイアウトに基づいて、サブピクチャ522、523、622、722、および／または822のようなサブピクチャの、ビットストリーム500および／またはサブビットストリーム501のようなビットストリームをデコードする方法1100のフローチャートである。方法1100は、方法100を実行するとき、コーデックシステム200、デコーダ400、および／またはビデオコーディングデバイス900のようなデコーダによって利用され得る。たとえば、方法1100は、方法1000の結果として作り出されるビットストリームをデコードするために適用され得る。

【0174】

方法1100は、デコーダがサブピクチャを含むビットストリームを受信することを開始するとき開始し得る。ビットストリームは完全なビデオシーケンスを含んでもよく、または、ビットストリームは別々の抽出のためにサブピクチャの低減されたセットを含むサブビットストリームであってもよい。ステップ1101において、ビットストリームが受信される。ビットストリームは、ピクチャから区分されたサブピクチャを備える。ビットストリームはSPSも備える。SPSは、サブピクチャサイズおよびサブピクチャ位置を備える。いくつかの例では、サブピクチャは時間動き制約サブピクチャである。そのような場合、サブピクチャサイズおよびサブピクチャ位置は、動き制約サブピクチャのレイアウトを示す。いくつかの例では、SPSは、ピクチャから区分された各サブピクチャについてのサブピクチャIDをさらに備え得る。

【0175】

ステップ1103において、SPSが、サブピクチャサイズおよびサブピクチャ位置を取得するために解析される。サブピクチャサイズは、ルマサンプルにおけるサブピクチャ高さおよびルマサンプルにおけるサブピクチャ幅を含み得る。サブピクチャ位置は、サブピクチャの左上サンプルとピクチャの左上サンプルとの間のオフセット距離を含み得る。サブピクチャは、また、時間動き制約サブピクチャフラグおよび／またはサブピクチャIDのよう

な他のサブピクチャ関連のデータを取得するために解析され得る。

【0176】

ステップ1105において、サブピクチャのサイズが、サブピクチャサイズに基づいてディスプレイのサイズに対して決定されることが可能である。さらに、サブピクチャの位置が、サブピクチャ位置に基づいてディスプレイに対して決定されることが可能である。デコーダは、また、時間動き制約サブピクチャフラグに基づいて、サブピクチャが独立にデコードされることが可能であるかどうかを決定することができる。従って、デコーダは、SPSからの解析されたデータおよび／またはサブピクチャに含まれるスライスと関連付けられるスライスヘッダからの対応するデータに基づいて、サブピクチャのレイアウトを決定することができる。

10

【0177】

ステップ1107において、サブピクチャが、サブピクチャサイズ、サブピクチャ位置、および／または、SPS、PPS、スライスヘッダ、SEIメッセージなどから取得される他の情報に基づいてデコードされる。サブピクチャは、ビデオシーケンスを作り出すためにデコードされる。ステップ1109において、ビデオシーケンスが次いで、表示のために転送されることが可能である。

【0178】

図12は、サブピクチャ522、523、622、722、および／または822のためのレイアウトのようなサブピクチャ・レイアウトを、ビットストリーム500および／またはサブビットストリーム501のようなビットストリームを介してシグナリングするための一例のシステム1200の概略図である。システム1200は、コーデックシステム200、エンコーダ300、デコーダ400、および／またはビデオコーディングデバイス900のようなエンコーダおよびデコーダによって実装され得る。さらに、システム1200は、方法100、1000、および／または1100を実装するときを利用され得る。

20

【0179】

システム1200はビデオエンコーダ1202を含む。ビデオエンコーダ1202は、現在のサブピクチャを含む複数のサブピクチャへとピクチャを区分するための区分モジュール1201を備える。ビデオエンコーダ1202は、ピクチャから区分されたサブピクチャをビットストリームへとエンコードし、サブピクチャのサブピクチャサイズおよびサブピクチャ位置をビットストリーム内のSPSへとエンコードするためのエンコードモジュール1203をさらに備える。ビデオエンコーダ1202は、デコーダに向けた通信のためにビットストリームを記憶するための記憶モジュール1205をさらに備える。ビデオエンコーダ1202は、サブピクチャ、サブピクチャサイズ、およびサブピクチャ位置を含むビットストリームをデコーダに向けて送信するための送信モジュール1207をさらに備える。ビデオエンコーダ1202は、方法1000のステップのいずれかを実行するようにさらに構成され得る。

30

【0180】

システム1200はビデオデコーダ1210も含む。ビデオデコーダ1210は、ピクチャから区分されたサブピクチャを備えるビットストリームと、サブピクチャのサブピクチャサイズおよびサブピクチャのサブピクチャ位置を備えるSPSとを受信するための受信モジュール1211を備える。ビデオデコーダ1210は、SPSを解析してサブピクチャサイズおよびサブピクチャ位置を取得するための解析モジュール1213をさらに備える。ビデオデコーダ1210は、サブピクチャサイズおよびサブピクチャ位置に基づいてサブピクチャをデコードしてビデオシーケンスを作り出すためのデコードモジュール1215をさらに備える。ビデオデコーダ1210は、表示のためにビデオシーケンスを転送するための転送モジュール1217をさらに備える。ビデオデコーダ1210は、方法1100のステップのいずれかを実行するようにさらに構成され得る。

40

【0181】

第1のコンポーネントと第2のコンポーネントとの間の線、トレース、または別の媒体を除き、介在するコンポーネントがないとき、第1のコンポーネントは第2のコンポーネントに直接に結合される。第1のコンポーネントと第2のコンポーネントとの間に線、トレース

50

、または別の媒体以外の介在するコンポーネントがあるとき、第1のコンポーネントは第2のコンポーネントに間接に結合される。用語「結合される」およびその変形は、直接に結合されるおよび間接に結合されるの両方を含む。用語「約」の使用は、そうでなく述べられるのでなければ、後続の数字の±10%を含む範囲を意味する。

【0182】

ここに記載される典型的な方法のステップは、必ずしも説明された順序で実行されることは要求されないことも理解されるべきであり、そのような方法のステップの順序は単に典型的であると理解されるべきである。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首尾一貫する方法において、あるステップが省略または組み合わされてもよい。

【0183】

いくつかの実施形態が本開示において提供されたが、開示されたシステムおよび方法は、本開示の精神または範囲から逸脱することなく、多くの他の特定の形式で具現化され得ることが理解され得る。本例は、例示的であり、制限的でないとして考慮されるべきであり、意図はここで与えられる詳細に限定されることでない。たとえば、別のシステムでは様々な要素またはコンポーネントが組み合わされ、または統合されてもよく、またはある特徴が省略され、または実装されなくてもよい。

【0184】

加えて、様々な実施形態において個別または別々として説明され例示される技法、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、コンポーネント、技法、または方法と組み合わされ、または統合されてもよい。改変、置換、および変更の他の例は、この技術分野の当業者によって確認可能であり、ここで開示される精神および範囲から逸脱することなく行われ得る。

【符号の説明】

【0185】

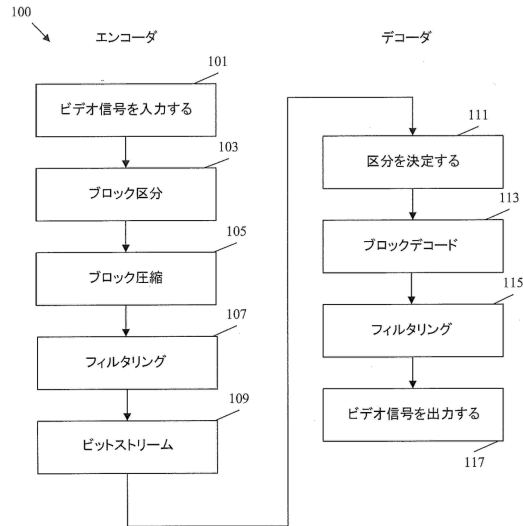
- 200 コーデックシステム
- 201 区分されたビデオ信号
- 211 汎用コーダ制御コンポーネント
- 213 変換スケーリングおよび量子化コンポーネント
- 215 イントラピクチャ推定コンポーネント
- 217 イントラピクチャ予測コンポーネント
- 219 動き補償コンポーネント
- 221 動き推定コンポーネント
- 223 デコードされたピクチャバッファコンポーネント
- 225 ループ内フィルタコンポーネント
- 227 フィルタ制御分析コンポーネント
- 229 スケーリングおよび逆変換コンポーネント
- 231 ヘッドフォーマッティングおよびCABACコンポーネント
- 301 区分されたビデオ信号
- 313 変換および量子化コンポーネント
- 317 イントラピクチャ予測コンポーネント
- 321 動き補償コンポーネント
- 323 デコードされたピクチャバッファコンポーネント
- 325 ループ内フィルタコンポーネント
- 329 逆変換および量子化コンポーネント
- 331 エントロピーコーディングコンポーネント
- 417 イントラピクチャ予測コンポーネント
- 421 動き補償コンポーネント
- 423 デコードされたピクチャバッファコンポーネント
- 425 ループ内フィルタコンポーネント

429	逆変換および量子化コンポーネント	
433	エントロピーデコードコンポーネント	
500	ビットストリーム	
501	サブビットストリーム	
510	SPS	
512	PPS	
514	スライスヘッダ	
515	SEIメッセージ	
520	画像データ	
521	ピクチャ	10
522	サブピクチャ	
523	サブピクチャ	
524	スライス	
525	スライス	
531	サブピクチャサイズ	
532	サブピクチャ位置	
533	サブピクチャID	
534	動き制約サブピクチャフラグ	
535	サブピクチャレベル	
600	ピクチャ	20
622	サブピクチャ	
631	サブピクチャサイズ	
631a	サブピクチャ幅	
631b	サブピクチャ高さ	
632	位置	
633	サブピクチャID	
634	時間動き制約サブピクチャ	
642	左上サンプル	
700	機構	
722	サブピクチャ	30
724	スライス	
733	サブピクチャID	
742	左上の角	
801	ピクチャの右の境界	
802	ピクチャの下の境界	
822	サブピクチャ	
825	CTU	
900	ビデオコーディングデバイス	
910	送信機／受信機	
914	コーディングモジュール	40
920	ダウンストリームポート	
930	プロセッサ	
932	メモリ	
950	アップストリームポート	
960	I/Oデバイス	
1200	システム	
1201	区分モジュール	
1202	ビデオエンコーダ	
1203	エンコードモジュール	
1205	記憶モジュール	50

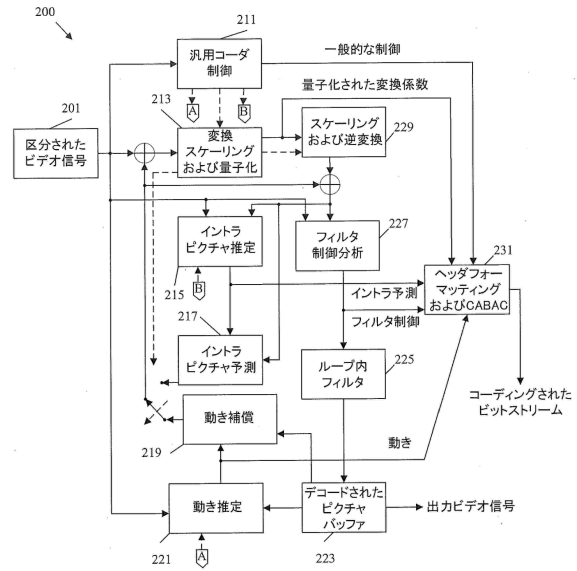
- 1207 送信モジュール
 1210 ビデオデコーダ
 1211 受信モジュール
 1213 解析モジュール
 1215 デコードモジュール
 1217 転送モジュール

【図面】

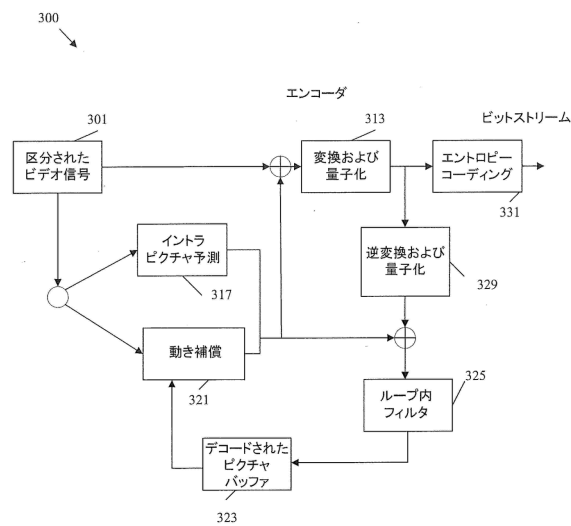
【図 1】



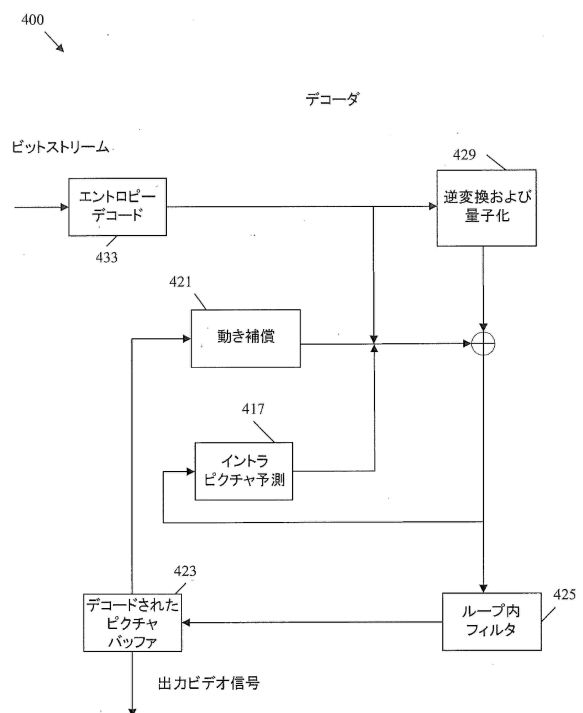
【図 2】



【図 3】



【図 4】



10

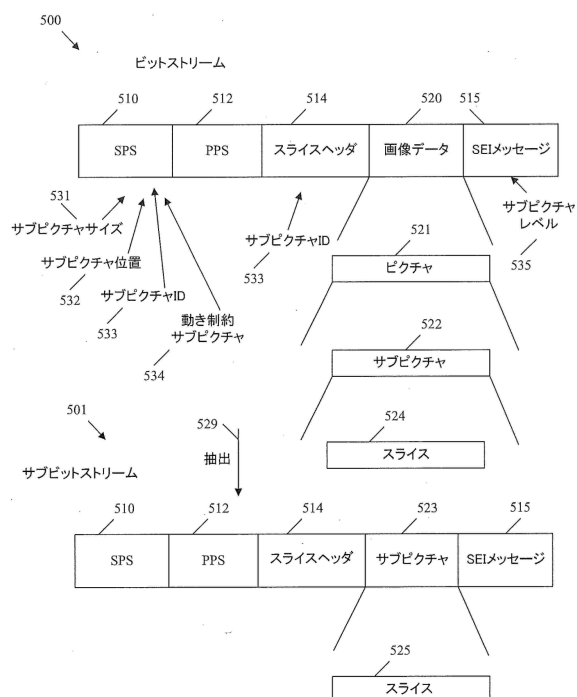
20

30

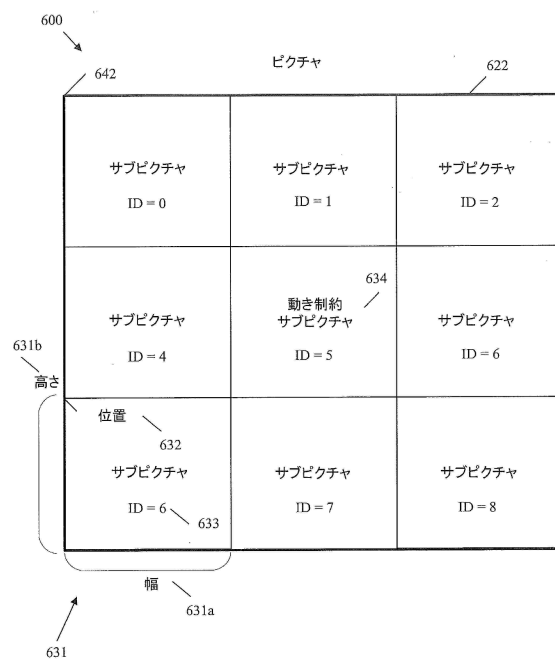
40

50

【図 5】



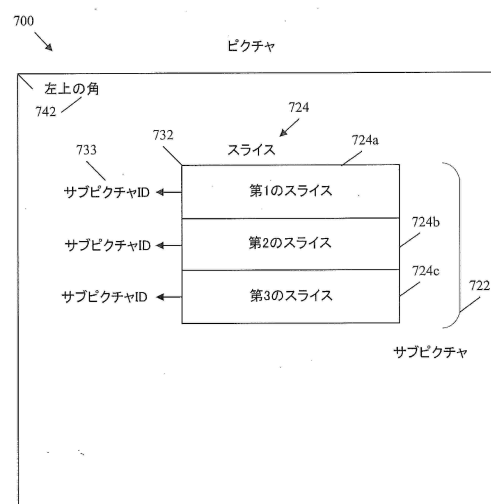
【圖 6】



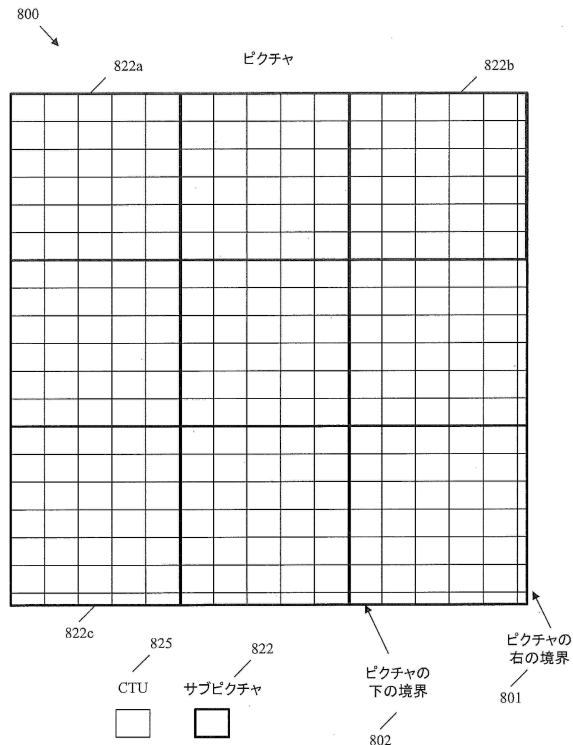
10

20

【図 7】



【図 8】

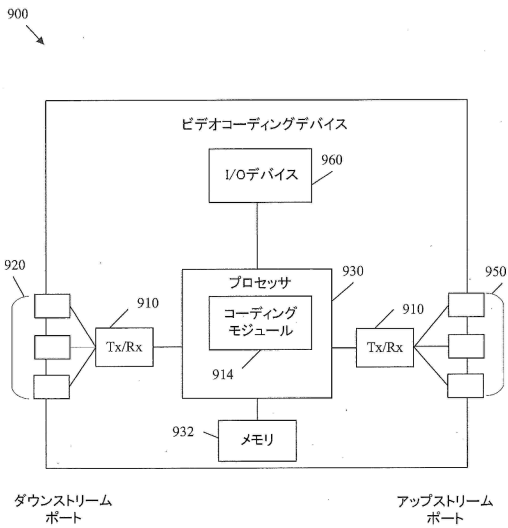


30

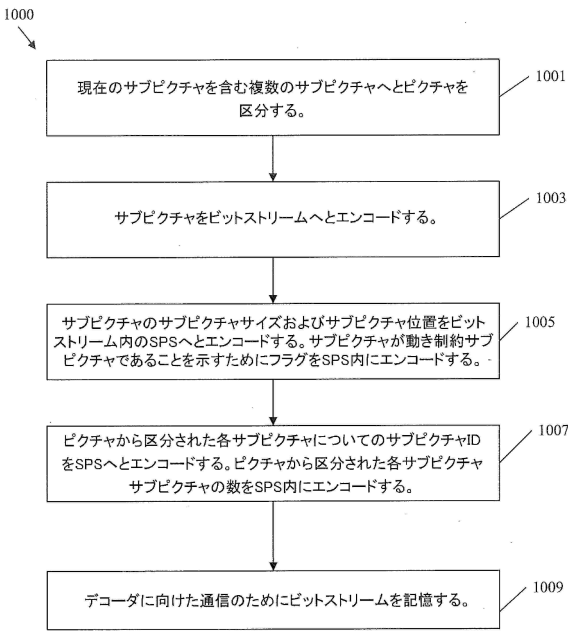
40

50

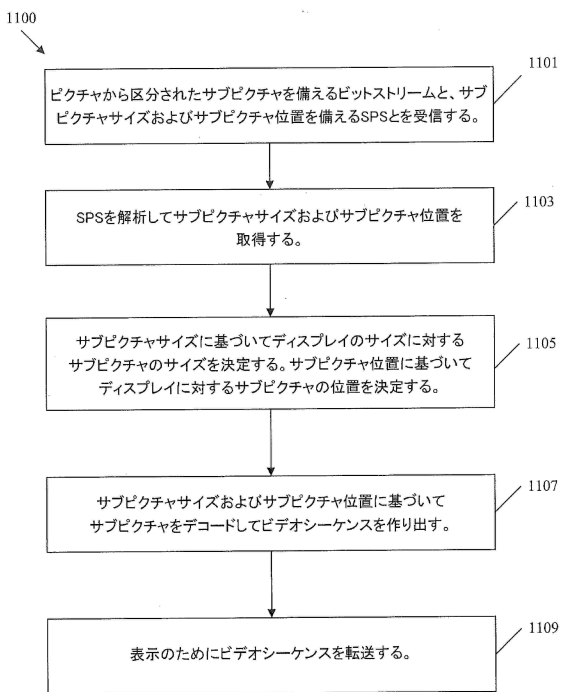
【図 9】



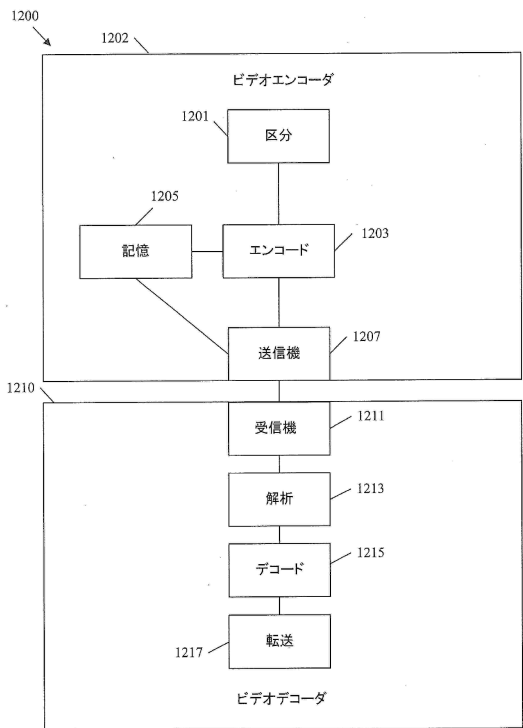
【図 10】



【図 11】



【図 12】



10

20

30

40

50

フロントページの続き

- クレスト・ウェイ・ 6 2 6 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・ 9 2 1 2 9 ・サン・ディエゴ・フォックスクロフト・プレイス
・ 8 5 8 4
審査官 岩井 健二
(56)参考文献 国際公開第 2 0 1 8 / 1 7 2 2 3 4 (WO, A 2)
国際公開第 2 0 1 7 / 1 3 7 4 4 4 (WO, A 1)
米国特許出願公開第 2 0 1 8 / 0 1 0 1 9 6 7 (US, A 1)
Miska M. Hannuksela and Alireza Aminlou, AHG12: On grouping of tiles, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-M0261, 13th Meeting: Marrakech, MA, 2019年01月03日, pp.1-11
Rickard Sjoberg, Mitra Damghanian, and Martin Pettersson, AHG12: Flexible tiles to support MCTS use cases, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-M0374-v2, 13th Meeting: Marrakesh, MA, 2019年01月04日, pp.1-6
Byeongdoo Choi, Stephan Wenger and Shan Liu, AHG12: On sub-picture partitioning, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N0055-v2, 14th Meeting: Geneva, CH, 2019年03月, pp.1-6
Ye-Kui Wang, Hendry, and Jianle Chen, AHG12: Sub-picture-based coding for VVC, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-N0107-v1, 14th Meeting: Geneva, CH, 2019年03月, pp.1-4
(58)調査した分野 (Int.Cl., DB名)
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8