

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7486576号
(P7486576)

(45)発行日 令和6年5月17日(2024.5.17)

(24)登録日 令和6年5月9日(2024.5.9)

(51)国際特許分類 F I
H 0 4 N 19/174 (2014.01) H 0 4 N 19/174
H 0 4 N 19/70 (2014.01) H 0 4 N 19/70

請求項の数 8 (全32頁)

(21)出願番号	特願2022-517854(P2022-517854)	(73)特許権者	598036300
(86)(22)出願日	令和2年6月26日(2020.6.26)		テレフォンアクチャーボラゲット エルエム
(65)公表番号	特表2022-549798(P2022-549798 A)		エリクソン(パブル)
(43)公表日	令和4年11月29日(2022.11.29)		スウェーデン国 ストックホルム エス - 1 6 4 8 3
(86)国際出願番号	PCT/SE2020/050668	(74)代理人	100109726
(87)国際公開番号	WO2021/061033		弁理士 園田 吉隆
(87)国際公開日	令和3年4月1日(2021.4.1)	(74)代理人	100150670
審査請求日	令和4年6月3日(2022.6.3)		弁理士 小椋 晴美
(31)優先権主張番号	62/904,286	(74)代理人	100199705
(32)優先日	令和1年9月23日(2019.9.23)		弁理士 仙波 和之
(33)優先権主張国・地域又は機関	米国(US)	(74)代理人	100194294
前置審査			弁理士 石岡 利康
		(72)発明者	ダムガニアン, ミトラ
			スウェーデン国 1 9 7 9 2 ウップラン
			最終頁に続く

(54)【発明の名称】 サブピクチャスライス位置導出を伴うセグメント位置シグナリング

(57)【特許請求の範囲】

【請求項1】

デコーダによって実行される方法(600)であって、前記方法は、
符号化ビデオビットストリームを受信することであって、前記符号化ビデオビットストリ
ームが、スライスアドレスの第1の部分を表す第1の値を符号化する第1のコードワ
ードを含むスライスヘッダを含み、前記第1の値は、スライスが属するサブピクチャを示すサ
ブピクチャIDであり、前記スライスヘッダが、前記スライスアドレスの第2の部分を表
す第2の値を符号化する第2のコードワードを含み、前記第2の値は、前記スライスが属
するサブピクチャ位置に対する、前記スライスの空間配置を示すローカルスライスアドレ
スであり、前記スライスアドレスが、ピクチャ内のスライスの空間ロケーションを指定す
る、符号化ビデオビットストリームを受信することと、
前記第1のコードワードから前記第1の値を復号することと、
前記第2のコードワードから前記第2の値を復号することと、
前記第1の値と前記第2の値とを使用して、前記スライスアドレスを導出することと、
前記スライスを復号するために前記スライスアドレスを使用することと、
を含み、
前記スライスアドレスを導出することが、
パラメータセット中のシンタックスエレメントからマッピングリストを導出することと、
前記第1の値を、前記第1の値とは異なるマッピングされる値にマッピングするために前
記マッピングリストを使用すること、

10

20

前記スライスアドレスを導出するために、前記マッピングされる値を使用することと、を含む、方法。

【請求項 2】

前記方法は、

第 3 のコードワードから第 3 の値を復号することであって、前記第 3 の値が前記スライスアドレスの第 3 の部分を表し、前記スライスアドレスの前記第 3 の部分が、前記サブピクチャレベルよりも低いスライスレベルにおけるアドレスを表す、第 3 の値を復号することと、

前記スライスアドレスを導出するために前記第 1 の値と前記第 2 の値と前記第 3 の値とを使用することと

10

をさらに含む、請求項 1 に記載の方法。

【請求項 3】

前記方法は、

前記符号化ビデオビットストリームにおけるコードワードから数 N を復号することをさらに含み、

前記第 1 の値または前記第 2 の値を前記復号することが、前記符号化ビデオビットストリームから前記数 N を復号することを含む、

請求項 1 または 2 に記載の方法。

【請求項 4】

前記数 N が、前記サブピクチャ中のスライスの数を表すか、または

20

前記数 N が、前記サブピクチャ中のスライスの最大数を表す、

請求項 3 に記載の方法。

【請求項 5】

前記方法が、

前記符号化ビデオビットストリームが参照するパラメータセット中のフラグからフラグ値を復号することをさらに含み、

前記フラグ値が第 1 のフラグ値に等しい場合、前記符号化ビデオビットストリーム中の各サブピクチャ中に 1 つのスライスのみがあり、

前記フラグ値が第 2 のフラグ値に等しい場合、前記符号化ビデオビットストリーム中のサブピクチャ中に 2 つ以上のスライスがあり得る、請求項 1 から 4 のいずれか一項に記載の方法。

30

【請求項 6】

処理回路 (8 0 2) によって実行されたとき、前記処理回路 (8 0 2) に、請求項 1 から 5 のいずれか一項に記載の方法を実行させる命令 (8 4 4) を備える、コンピュータプログラム (8 4 3) 。

【請求項 7】

請求項 6 に記載のコンピュータプログラムを記憶しているコンピュータ可読記憶媒体 (8 4 2) 。

【請求項 8】

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ように適応された復号装置 (8 0 0) 。

40

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、ビデオコーディングおよび復号に関する。

【背景技術】

【0002】

1. HEVC および VVC

【0003】

高効率ビデオコーディング (HEVC) は、時間予測と空間予測の両方を利用する、 I

50

TU - TおよびMPEGによって規格化されたブロックベースビデオコーデックである。空間予測は、現在ピクチャ内からのイントラ（I）予測を使用して達成される。時間予測は、前に復号された参照ピクチャから、ブロックレベルでの単方向（P）予測または双方向インター（B）予測を使用して達成される。エンコーダでは、残差と呼ばれる、元のピクセルデータと予測されたピクセルデータとの間の差は、周波数ドメインに変換され、量子化され、次いで、同じくエントロピーコーディングされる、予測モードおよび動きベクトルなど、必要な予測パラメータとともに送信される前に、エントロピーコーディングされる。デコーダは、エントロピー復号と、逆量子化と、逆方向変換とを実施して残差を取得し、次いで、その残差をイントラ予測またはインター予測に追加してピクチャを再構築する。

10

【0004】

MPEGおよびITU - Tは、ジョイントビデオエクスプロラトリチーム（JVET：Joint Video Exploratory Team）内でHEVCの後継に取り組んでいる。開発中のこのビデオコーデックの名前は、多用途ビデオコーディング（VVC）である。これを書いているとき、VVCドラフト仕様の現在のバージョンは、「多用途ビデオコーディング（ドラフト6）」（JVET - 02001 - vE）であった。本明細書でVVCが言及されるとき、VVCは、VVC仕様のドラフト6を指す。

【0005】

2. 成分

【0006】

ビデオシーケンスは、各ピクチャが1つまたは複数の成分からなる一連のピクチャからなる。各成分は、サンプル値の2次元矩形アレイとして説明され得る。ビデオシーケンス中のピクチャは、3つの成分、すなわち、サンプル値がルーマ値である1つのルーマ成分（Y）と、サンプル値がクロマ値である2つのクロマ成分（Cb）および（Cr）とからなることが一般的である。クロマ成分の次元は、各次元においてルーマ成分よりも1/2だけ小さいことが一般的である。たとえば、HDピクチャのルーマ成分のサイズは1920×1080となり、クロマ成分は、各々、960×540の次元を有するであろう。成分は色成分と呼ばれることがある。本明細書では、ビデオシーケンスの符号化および復号のために有用な方法について説明する。しかしながら、説明される技法が静止画像の符号化および復号のためにも使用され得ることを理解されたい。

20

30

【0007】

3. ブロックおよびユニット

【0008】

ブロックは、サンプルの2次元アレイである。ビデオコーディングでは、各成分が1つまたは複数のブロックにスプリットされ、コード化ビデオビットストリームは一連のブロックである。

【0009】

ビデオコーディングでは、ピクチャは、特定のエリアをカバーするユニットにスプリットされることが一般的である。各ユニットは、その特定のエリアをなすすべてのブロックからなり、各ブロックは、1つのユニットのみに完全に属する。HEVCおよびVVCにおけるコーディングユニット（CU）は、そのようなユニットの一例である。コーディングツリーユニット（CTU）は、いくつかのCUにスプリットされ得る論理ユニットである。

40

【0010】

HEVCでは、CUは正方形であり、すなわち、CUはN×Nルーマサンプルのサイズを有し、ここで、Nは、64、32、16または8の値を有することができる。現在のH.266テストモデル多用途ビデオコーディング（VVC）では、CUは、矩形でもあり、すなわち、N×Mルーマサンプルのサイズを有することができ、ここで、NはMとは異なる。

【0011】

50

4. NALユニット

【0012】

HEVCとVVCの両方が、ネットワークアブストラクションレイヤ(NAL)を規定する。HEVCおよびVVCにおけるすべてのデータ、すなわち、ビデオコーディングレイヤ(VCL)データまたは非VCLデータの両方が、NALユニット中にカプセル化される。VCL NALユニットは、ピクチャサンプル値を表すデータを含んでいる。非VCL NALユニットは、パラメータセットおよび補足エンハンスメント情報(SEI: supplemental enhancement information)メッセージなど、追加の関連するデータを含んでいる。HEVCおよびVVCの現在のバージョンにおけるNALユニットは、NALユニットヘッダと呼ばれるヘッダから始まる。HEVC 10
 CについてのNALユニットヘッダについてのシンタックスは、表1に示されており、スタートコードエミュレーションを防ぐために常に0に等しいものとするforbidden_zero_bitから開始する。それがなければ、いくつかのMPEGシステムは、HEVCビデオビットストリームを他のデータと混同し得るが、NALユニットヘッダ中の0ビットは、すべての可能なHEVCビットストリームをHEVCビットストリームとして一意に識別可能にする。nal_unit_type、nuh_layer_idおよびnuh_temporal_id_plus1コードワードは、それぞれ、どんなタイプのデータがNALユニット中で搬送されるかを識別するNALユニットのNALユニットタイプ、NALユニットが属するレイヤID、およびNALユニットが属する時間IDを指定する。NALユニットタイプは、NALユニットがどのようにパースおよび復号されるべきであることを示し、指定する。VVCの現在のバージョンにおけるNALユニットヘッダは、HEVCにおけるNALユニットヘッダとまったく同様であるが、nal_unit_typeのために1ビット少なく使用し、代わりに、将来の使用のためにこのビットを予約する。 20

【0013】

NALユニットのバイトの残りは、NALユニットタイプによって示されるタイプのペイロードである。ビットストリームは、一連の連結されたNALユニットからなる。

表1 - HEVC NALユニットヘッダシンタックス

記述子	
nal_unit_header() {	
forbidden_zero_bit	f(1)
nal_unit_type	u(6)
nuh_layer_id	u(6)
nuh_temporal_id_plus1	u(3)
}	

30

表2 - VVCの現在のバージョンのNALユニットヘッダシンタックス

記述子	
nal_unit_header() {	
forbidden_zero_bit	f(1)
nuh_reserved_zero_bit	u(1)
nuh_layer_id	u(6)
nal_unit_type	u(5)
nuh_temporal_id_plus1	u(3)
}	

40

【0014】

デコーダまたはビットストリームパーサは、NALユニットヘッダを見た後にNALユニットがどのようにハンドリング、たとえばパースおよび復号されるべきであることを結論付けることができる。NALユニットのバイトの残りは、NALユニットタイプによって示されるタイプのペイロードである。ビットストリームは、一連の連結されたNALユニットからなる。 50

【 0 0 1 5 】

NALユニットタイプは、NALユニットがどのようにパースおよび復号されるべきであることを示し、規定する。VCL NALユニットは、現在ピクチャのピクチャタイプに関する情報を提供する。VVCドラフトの現在のバージョンのNALユニットタイプは、表3に示されている。

【 0 0 1 6 】

復号順序は、ビットストリーム内のNALユニットの順序と同じである、NALユニットが復号されるものとする順序である。復号順序は出力順序とは異なり得、出力順序は、復号されたピクチャがデコーダによって表示などのために出力されるべきである順序である。

表3 - VVCドラフトの現在のバージョンにおけるNALユニットタイプ

nal_unit_type	nal_unit_type の名前	NALユニットのコンテンツおよびRBSP シンタックス構造	NALユニット タイプクラス
0	TRAIL_NUT	トレーリングピクチャのコード化スライス slice_layer_rbsp()	VCL
1	STSA_NUT	STSAピクチャのコード化スライス slice_layer_rbsp()	VCL
2	RASL_NUT	RASLピクチャのコード化スライス slice_layer_rbsp()	VCL
3	RADL_NUT	RADLピクチャのコード化スライス slice_layer_rbsp()	VCL
4..7	RSV_VCL4.. RSV_VCL7	予約済み非IRAP VCL NALユニットタイプ	VCL
8	IDR_W_RADL	IDRピクチャのコード化スライス slice_layer_rbsp()	VCL
9	IDR_N_LP		
10	CRA_NUT	CRAピクチャのコード化スライス silce_layer_rbsp()	VCL
11	GDR_NUT	GDRピクチャのコード化スライス slice_layer_rbsp()	VCL
12	RSV_IRAP_VCL1	予約済みIRAP VCL NALユニットタイプ	VCL
13	2 RSV_IRAP_VCL1 3		
14..15	RSV_VCL14.. RSV_VCL15	予約済み非IRAP VCL NALユニットタイプ	VCL
16	SPS_NUT	シーケンスパラメータセット seq_parameter_set_rbsp()	非VCL
17	PPS_NUT	ピクチャパラメータセット pic_parameter_set_rbsp()	非VCL
18	APS_NUT	適応パラメータセット adaptation_parameter_set_rbsp()	非VCL
19	AUD_NUT	アクセスユニットデリミタ access_unit_delimiter_rbsp()	非VCL
20	EOS_NUT	シーケンスの終了 end_of_seq_rbsp()	非VCL
21	EOB_NUT	ビットストリームの終了 end_of_bitstream_rbsp()	非VCL
22, 23	PREFIX_SEI_NU T SUFFIX_SEI_NU T	補足エンハンスメント情報 sei_rbsp()	非VCL
24	DPS_NUT	復号パラメータセット decoding_parameter_set_rbsp()	非VCL
25..27	RSV_NVCL25.. RSV_NVCL27	予約済み非VCL NALユニットタイプ	非VCL
28..31	UNSPEC28.. UNSPEC31	指定されない非VCL NALユニットタイプ	非VCL

10

20

30

40

50

【 0 0 1 7 】

5 . イントラランダムアクセスポイント (I R A P) ピクチャおよびコード化ビデオシーケンス (C V S)

【 0 0 1 8 】

H E V C におけるイントラランダムアクセスポイント (I R A P) ピクチャは、その復号プロセスにおける予測のために、それ自体以外のいかなるピクチャも参照しないピクチャである。H E V C における復号順序におけるビットストリーム中の第 1 のピクチャは、I R A P ピクチャでなければならないが、I R A P ピクチャは、さらに、ビットストリームの後半においても現れ得る。H E V C は、3 つのタイプの I R A P ピクチャ、すなわち、切断リンクアクセス (B L A) ピクチャと、瞬時デコーダリフレッシュ (I D R) ピクチャと、クリーンランダムアクセス (C R A) ピクチャとを指定する。

10

【 0 0 1 9 】

H E V C におけるコード化ビデオシーケンス (C V S) は、I R A P アクセスユニットにおいて開始し、復号順序における次の I R A P アクセスユニットまでの、ただし次の I R A P アクセスユニットを含まない、一連のアクセスユニットである。

【 0 0 2 0 】

I D R ピクチャは、常に、新しい C V S を開始する。I D R ピクチャは、関連するランダムアクセス復号可能リーディング (R A D L : r a n d o m a c c e s s d e c o d a b l e l e a d i n g) ピクチャを有し得る。I D R ピクチャは、関連するランダムアクセススキップリーディング (R A S L : r a n d o m a c c e s s s k i p p e d l e a d i n g) ピクチャを有しない。

20

【 0 0 2 1 】

H E V C における B L A ピクチャはまた、新しい C V S を開始し、復号プロセスに、I D R ピクチャと同じ影響を及ぼす。しかしながら、H E V C における B L A ピクチャは、参照ピクチャの空でないセットを指定するシンタックスエレメントを含んでいることがある。B L A ピクチャは関連する R A S L ピクチャを有し得、関連する R A S L ピクチャは、関連する R A S L ピクチャが、ビットストリーム中に存在しないことがあるピクチャへの参照を含んでいることがあるとき、デコーダによって出力されず、復号可能でないことがある。B L A ピクチャは、復号される関連する R A D L ピクチャをも有し得る。B L A ピクチャは、V V C の現在のバージョンにおいて規定されない。

30

【 0 0 2 2 】

C R A ピクチャは、関連する R A D L または R A S L ピクチャを有し得る。B L A ピクチャの場合と同様に、C R A ピクチャは、参照ピクチャの空でないセットを指定するシンタックスエレメントを含んでいることがある。C R A ピクチャでは、関連する R A S L ピクチャが、ビットストリーム中に存在しないピクチャへの参照を含んでいることがあるとき、関連する R A S L ピクチャが復号可能でないことがあるので、関連する R A S L ピクチャがデコーダによって出力されないことを指定するように、フラグがセットされ得る。C R A は、C V S を開始し得る。

【 0 0 2 3 】

V V C ドラフトの現在のバージョンでは、C V S は、C V S 開始 (C V S S) アクセスユニットにおいて開始され、C V S S アクセスユニットは、I R A P ピクチャ、すなわち、I D R または C R A ピクチャ、あるいは漸進的復号リフレッシュ (G D R : g r a d u a l d e c o d i n g r e f r e s h) ピクチャを含んでいることがある。

40

【 0 0 2 4 】

G D R ピクチャは、本質的に、低遅延コーディングのために符号化されたビットストリーム中のランダムアクセスのために使用され、ここで、完全な I R A P ピクチャは、あまりに多くの遅延を引き起こすことになる。G D R ピクチャは、ビデオピクチャごとに更新する漸進的イントラリフレッシュを使用し得、ここで、各ピクチャは、部分的にイントラコーディングされるにすぎない。ビットストリームが G D R ピクチャにおいてチューニングされたことを仮定すれば、ビデオがいつ完全にリフレッシュされ、出力の準備ができて

50

いるかが、GDRピクチャによりシグナリングされる。GDRは、CVSを開始し得る。

【0025】

6. パラメータセット

【0026】

HEVCおよびVVCは、3つのタイプのパラメータセット、すなわち、ピクチャパラメータセット(PPS)と、シーケンスパラメータセット(PPS)と、ビデオパラメータセット(VPS)とを指定する。PPSは、1つまたは複数のピクチャについて共通であるデータを含んでおり、SPSは、コード化ビデオシーケンス(CVS)について共通であるデータを含んでおり、VPSは、複数のCVSについて共通であるデータを含んでいる。ビットストリーム中でランダムアクセスポイントを提供するために、IRAPピクチャまたはGDRピクチャとしてピクチャを周期的に符号化することが一般的であり、ここで、各そのようなピクチャは、復号するために必要なパラメータセット(VPS、SPS、PPS)に先行される。

10

【0027】

VVCの現在のバージョンは、2つの追加のパラメータセット、すなわち、適応パラメータセット(APS)とデコーダパラメータセット(DPS)とをも指定する。

【0028】

APSは、適応ループフィルタ(ALF)ツールと、ルーママッピングおよびクロマスケールリング(LMCS:luma mapping and chroma scaling)ツールとのために必要とされるパラメータを搬送する。

20

【0029】

DPSは、復号セッション中に変化しないことがあり、デコーダが知るのに良好であり得る、情報、たとえば許容されるサブレイヤの最大数を含んでいる。DPS中の情報は、復号プロセスの動作のために必要でない。

【0030】

7. タイルおよびブリック

【0031】

ドラフトVVCビデオコーディング規格は、ピクチャを、タイルと呼ばれることがある、矩形の空間的に独立した領域に分割するタイルツールを含む。ドラフトVVCコーディング規格におけるタイルは、HEVCにおいて使用されるタイルと同様であるが、2ステップ区分機構をもつ。タイルツールを使用して、HEVCにおけるピクチャがサンプルの行および列に区分され得、ここで、タイルは行と列との交差部である。図9Aは、ピクチャのための合計20個のタイルを生じる、4つのタイル行と5つのタイル列とを使用する区分の一例を示す。

30

【0032】

タイル構造は、行の高さと列の幅とを指定することによって、ピクチャパラメータセット(PPS)中でシグナリングされる。個々の行および列は、異なるサイズを有することができるが、区分は、常に、それぞれ左から右に、および上から下に、ピクチャ全体にわたってスパンする。

【0033】

同じピクチャのタイル間に復号依存性はない。これは、イントラ予測と、エントロピーコーディングのためのコンテキスト選択と、動きベクトル予測とを含む。1つの例外は、ループ内フィルタ処理依存性が概してタイル間で許容されることである。

40

【0034】

VVCにおける2ステップタイル区分は、HEVCの場合のように、ピクチャをタイルに区分することによって開始する。次いで、各タイルは、図9Bの右側に示されているように、随意に、水平境界によってブリックに区分され得る。現在のVVC仕様ドラフトでは、ブリックという単語は、さらに区分されないタイルのためにも使用され、これは、図9Bの右側のピクチャが9つのブリックからなることを意味する。

【0035】

50

8 . スライス

【 0 0 3 6 】

H E V Cにおけるスライスの概念は、ピクチャを独立してコーディングされたスライスに分割し、ここで、ピクチャ中の1つのスライスの復号は、同じピクチャ中の他のスライスから独立している。異なるコーディングタイプが同じピクチャのスライスのために使用され得る、すなわち、スライスは、Iスライス、Pスライス、またはBスライスのいずれかであり得る。スライスの1つの目的は、データ損失の場合に再同期を可能にすることである。

【 0 0 3 7 】

V V Cの現在のバージョンでは、スライスは、いくつかの完全なタイル、または1つのタイルの完全なブリックの連続するシーケンスのみのいずれかからなる。各スライスは、i) 個々のスライスについてセットされ得るパラメータを備えるスライスヘッダと、i i) スライスデータとを有する。いくつかのパラメータは、ピクチャ中のすべてのスライスについて同じであるように制限される。C V S中の各スライスは、別個のV C L N A Lユニット中で搬送される。

10

【 0 0 3 8 】

V V Cドラフト仕様の前のバージョンでは、スライスは、タイルグループと呼ばれた。

【 0 0 3 9 】

V V Cの現在のバージョンでは、スライスの2つのモード、すなわち、ラスト走査スライスモードおよび矩形スライスモードがサポートされる。ラスト走査スライスモードでは、スライスは、ピクチャのタイルラスト走査におけるタイルのシーケンスを含んでいる。矩形スライスモードでは、スライスは、ピクチャの矩形領域をまとめて形成する、ピクチャのいくつかのブリックを含んでいる。矩形スライス内のブリックは、スライスのブリックラスト走査の順序おけるものである。

20

【 0 0 4 0 】

V V Cドラフト仕様の現在のバージョンでは、スライスヘッダ中で与えられる `s l i c e _ a d d r e s s` (表4参照) は、ピクチャ中のスライスについての空間位置を導出するために使用される。

30

40

50

表4 - VVC仕様ドラフトの現在のバージョンにおけるスライスヘッダ中のスライスアドレスシンタックス

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag == NumBricksInPic > 1)	
slice_address	u(v)
...	
}	

セマンティクス:

slice_addressはスライスのスライスアドレスを指定する。存在しないとき、slice_addressの値は0に等しいと推論される。

rect_slice_flagが0に等しい場合、以下が適用される。

- スライスアドレスは、式(7-59)によって指定されるブリックIDである。
- slice_addressの長さは、Ceil(Log2(NumBricksInPic))ビットである。
- slice_addressの値は、両端値を含む0~NumBricksInPic-1の範囲内にあるものとする。

他の場合(rect_slice_flagが1に等しい)、以下が適用される。

- スライスアドレスは、スライスのスライスIDである。
- slice_addressの長さは、signalled_slice_id_length_minus1+1ビットである。
- signalled_slice_id_flagが0に等しい場合、slice_addressの値は、両端値を含む0~num_slices_in_pic_minus1の範囲内にあるものとする。他の場合、slice_addressの値は、両端値を含む0~2^(signalled_slice_id_length_minus1)-1の範囲内にあるものとする。

以下の制約が適用されることが、ビットストリーム適合性の要件である。

- slice_addressの値は、同じコード化ピクチャの他のコード化スライスNALユニットのslice_addressの値に等しくないものとする。
- rect_slice_flagが0に等しいとき、ピクチャのスライスは、それらのslice_address値の昇順のものとする。
- ピクチャのスライスの形状は、各ブリックが、復号されたとき、ピクチャ境界からなる、または(1つまたは複数の)前に復号されたブリックの境界からなる、そのブリックの左境界全体および上境界全体を有するものとするようなものであるものとする。

10

20

30

【0041】

9. サブピクチャ

【0042】

サブピクチャは、VVCの現在のバージョンにおいてサポートされる。サブピクチャは、ピクチャ内の1つまたは複数のスライスの矩形領域として規定される。これは、サブピクチャが、ピクチャの矩形領域をまとめてカバーする1つまたは複数のスライスを含んでいることを意味する。VVC仕様の現在のバージョンでは、サブピクチャロケーションおよびサイズは、SPS中でシグナリングされる。サブピクチャ領域の境界は、SPS中のサブピクチャごとのフラグ(per_subpicture_flag_subpic_treated_as_pic_flag[i])に条件付けられる(ループ内フィルタ処理演算を除外する)ピクチャ境界として扱われ得る。また、サブピクチャ境界上のループフィルタ処理は、SPS中のサブピクチャごとのフラグloop_filter_across_subpic_enabled_flag[i]に条件付けられる。表5は、VVCの現在のバージョンにおけるSPS中のサブピクチャシンタックスを示す。

40

50

表5 - VVC仕様ドラフトの現在のバージョンにおけるSPS中のサブピクチャシンタックス

seq_parameter_set_rbsp() {	記述子
...	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
}	
...	
}	
<p>セマンティクス:</p> <p>1に等しいsubpics_present_flagは、サブピクチャパラメータが、現在のところSPS RBSPシンタックス中に存在することを示す。0に等しいsubpics_present_flagは、サブピクチャパラメータが、現在のところSPS RBSPシンタックス中に存在しないことを示す。</p> <p>注2 - ビットストリームが、サブビットストリーム抽出プロセスの結果であり、サブビットストリーム抽出プロセスへの入力ビットストリームのサブピクチャのサブセットのみを含んでいるとき、SPSのRBSP中でsubpics_present_flagの値を1に等しくセットすることが必要とされ得る。</p> <p>max_subpics_minus1+1は、CVS中に存在し得るサブピクチャの最大数を指定する。max_subpics_minus1は0~254の範囲内にあるものとする。255の値は、ITU-T ISO/IECによる将来の使用のために予約される。</p> <p>subpic_grid_col_width_minus1+1は、4つのサンプルの単位でサブピクチャ識別子グリッドの各エレメントの幅を指定する。シンタックスエレメントの長さは、Ceil(Log2(pic_width_max_in_luma_samples/4))ビットである。</p> <p>変数NumSubPicGridColsは、以下のように導出される。</p> $\text{NumSubPicGridCols} = (\text{pic_width_max_in_luma_samples} + \text{subpic_grid_col_width_minus1} * 4 + 3) / (\text{subpic_grid_col_width_minus1} * 4 + 4)$ <p>subpic_grid_row_height_minus1+1は、4つのサンプルの単位でサブピクチャ識別子グリッドの各エレメントの高さを指定する。シンタックスエレメントの長さは、Ceil(Log2(pic_height_max_in_luma_samples/4))ビットである。</p> <p>変数NumSubPicGridRowsは、以下のように導出される。</p> $\text{NumSubPicGridRows} = (\text{pic_height_max_in_luma_samples} + \text{subpic_grid_row_height_minus1} * 4 + 3) / (\text{subpic_grid_row_height_minus1} * 4 + 4)$ <p>subpic_grid_idx[i][j]は、グリッド位置(i, j)のサブピクチャインデックスを指定する。シンタックスエレメントの長さは、Ceil(Log2(max_subpics_minus1+1))ビットである。</p> <p>変数SubPicTop[subpic_grid_idx[i][j]]、SubPicLeft[subpic_grid_idx[i][j]]、SubPicWidth[subpic_grid_idx[i][j]]、SubPicHeight[subpic_grid_idx[i][j]]、およびNumSubPicsは、以下のように導出される。</p> $\text{NumSubPics} = 0$ <p>for(i=0; i.<NumSubPicGridRows; i++) { for(j=0; j.<NumSubPicGridCols; j++) {</p>	

10

20

30

40

50

```

    if(i==0)
        SubPicTop[subpic_grid_idx[i][j]]=0
    else if(subpic_grid_idx[i][j]!=subpic_grid_idx[i-1][j]) {
        SubPicTop[subpic_grid_idx[i][j]]=i
        SubPicHeight[subpic_grid_idx[i-1][j]]=i-SubPicTop[subpic_grid_idx[i-1][
j]]
    }
    if(j==0)
        SubPicLeft[subpic_grid_idx[i][j]]=0
    else if(subpic_grid_idx[i][j]!=subpic_grid_idx[i][j-1]) {
        SubPicLeft[subpic_grid_idx[i][j]]=j
        SubPicWidth[subpic_grid_idx[i][j]]=j-SubPicLeft[subpic_grid_idx[i][j-1]
]
    }
    if(i==NumSubPicGridRows-1)
        SubPicHeight[subpic_grid_idx[i][j]]=i-SubPicTop[subpic_grid_idx[i-1][j]
]+1
    if(j==NumSubPicGridRows-1)
        SubPicWidth[subpic_grid_idx[i][j]]=j-SubPicLeft[subpic_grid_idx[i][j-1]
]+1
    if(subpic_grid_idx[i][j]>NumSubPics)
        NumSubPics=subpic_grid_idx[i][j]
}
}

```

10

20

1に等しいsubpic_treated_as_pic_flag[i]は、CVS中の各コード化ピクチャのi番目のサブピクチャがグループ内フィルタ処理演算を除く復号プロセスにおいてピクチャとして扱われることを指定する。0に等しいsubpic_treated_as_pic_flag[i]は、CVS中の各コード化ピクチャのi番目のサブピクチャがグループ内フィルタ処理演算を除く復号プロセスにおいてピクチャとして扱われないことを指定する。存在しないとき、subpic_treated_as_pic_flag[i]の値は0に等しいと推論される。

1に等しいloop_filter_across_subpic_enabled_flag[i]は、グループ内フィルタ処理演算がCVS中の各コード化ピクチャ中のi番目のサブピクチャの境界にわたって実施され得ることを指定する。0に等しいloop_filter_across_subpic_enabled_flag[i]は、グループ内フィルタ処理演算がCVS中の各コード化ピクチャ中のi番目のサブピクチャの境界にわたって実施されないことを指定する。存在しないとき、loop_filter_across_subpic_enabled_pic_flag[i]の値は1に等しいと推論される。

30

以下の制約が適用されることが、ビットストリーム適合性の要件である。

- 任意の2つのサブピクチャsubpicAおよびsubpicBについて、subpicAのインデックスがsubpicBのインデックスよりも小さいとき、subpicAの任意のコード化NALユニットが、復号順序においてsubpicBの任意のコード化NALユニットに続くものとする。

- サブピクチャの形状は、各サブピクチャが、復号されたとき、ピクチャ境界からなる、または前に復号されたサブピクチャの境界からなる、そのサブピクチャの左境界全体および上境界全体を有するものとするようなものであるものとする。

40

【発明の概要】

【0043】

いくつかの課題が存在する。たとえば、VVCドラフト仕様の現在のバージョンでは、スライスヘッダ中でシグナリングされるslice_address(表4参照)は、ピクチャ中のスライスの空間位置を導出するために使用されるu(v)コード化コードワードである。しかしながら、サブピクチャ区分が使用されるとき、サブピクチャ中のスライスの空間位置は、スライスヘッダ中のslice_addressコードワードから直接導出され得ず、このスライスがあるサブピクチャに属することは、スライスヘッダから導出され得ない。VVC仕様の現在のバージョンにおけるサブピクチャ中のスライスの空間位置を導出するために、第1に、ピクチャ中のスライスの空間位置が導出される必要があ

50

り、第2に、ピクチャ中のその空間位置があるサブピクチャに属することが導出される必要があり、次いで、そのことから、第3のステップにおいて、そのサブピクチャ中のスライスの空間位置が導出され得る。サブピクチャ中のスライスの空間位置を導出するためのこのマルチステッププロセスは、簡略化され得、これは、スライス復号プロセスとサブピクチャ中の復号されたピクセル値の配置とを容易にすることになる。

【0044】

さらに、(サブビットストリーム抽出およびマージの場合のように)サブピクチャが抽出またはマージされているとき、サブピクチャ中のスライスの空間位置は変化しない。これは、サブピクチャ位置に対するスライスの配置がサブビットストリーム抽出およびマージプロセスにおいて固定されることを意味する。この情報は、現在、VVC仕様の現在のバージョンにおいて活用されず、VVC仕様の現在のバージョンは、VVC仕様の現在のバージョンに記載のスライスアドレスのためのシグナリングが準最適であり、改善され得ることを示す。

10

【0045】

VVC仕様の現在のバージョンでは、サブピクチャ区分を使用する場合、このスライスがどのサブピクチャ中に空間的に位置するかは、スライスヘッダのみから導出され得ない。この関係も、サブピクチャが抽出またはマージされているとき、固定されるが、この情報は、VVC仕様の現在のバージョンにおいて活用されない。

【0046】

本開示は、VVC仕様の現在のバージョンの短所を克服することを目的とする。一実施形態では、短所は、i)スライスが属するサブピクチャと、ii)スライスが属するサブピクチャ位置に対するスライスの空間配置とを示す情報をスライスヘッダ中に含むことによって克服される。たとえば、1つの変形形態では、スライスヘッダは、スライスアドレスについての2つの値、すなわち、i)スライスが属するサブピクチャを示すサブピクチャIDについての1つの値と、ii)スライスが属するサブピクチャ位置に対するスライスの空間配置を示すスライスアドレスについての1つの値とを含む。スライスアドレスの2つの値を使用して、ピクチャ中のスライスについての空間位置は、次いで、たとえば、(スライスヘッダ中でシグナリングされる値のうちの1つとして)サブピクチャIDからピクチャ中のサブピクチャロケーションを導出することと、次いで、スライスヘッダ中でシグナリングされる他のスライスアドレス値からサブピクチャ中のスライスの位置を導出することとによって導出され得る。

20

30

【0047】

VVC仕様の現在のバージョンでは、ビットストリームがサブビットストリーム抽出の結果である場合、`slice_address`の値は、スライスの空間位置を導出するためにPPS中の`slice_id`の値にマッピングされる。この場合、サブビットストリーム中の含まれるスライスのサブセットが、「元の」ビットストリーム中のピクチャの左上コーナースライスを含まない場合、スライスIDの値と、スライスアドレスの値とは同じでなくなる。スライスIDは、サブビットストリーム抽出およびマージプロセスを可能にするための間接参照(`indirection`)機構を提供する。一実施形態では、スライスアドレスのための間接参照機構を使用する代わりに、間接参照機構とともにサブピクチャIDが使用され得る。この実施形態は、`sub_bitstream`抽出およびマージの場合に使用され得、ここで、サブピクチャに対するスライスアドレスは、プロセス中に同じままであり、サブピクチャIDは、初期サブピクチャIDを、新しいサブビットストリーム中のサブピクチャインデックスにマッピングするために、たとえば、SPSにおける間接参照機構を使用する。

40

【0048】

本開示のいくつかの態様およびそれらの実施形態は、上述の課題のソリューションを提供し得る。

【0049】

実施形態の第1の態様が、デコーダによって実施される方法を規定する。本方法は、コ

50

ード化ビデオストリーム (CVS) を受信することを含む。本方法は、CVS を処理することを含み、CVS は、セグメントアドレスの第 1 の部分を表す 1 つまたは複数の値の第 1 のセットを符号化する 1 つまたは複数のコードワードの第 1 のセットを含み、CVS は、セグメントアドレスの第 2 の部分を表す 1 つまたは複数の値の第 2 のセットを符号化する 1 つまたは複数のコードワードの第 2 のセットを含み、セグメントアドレスは、ピクチャ内のセグメントの空間ロケーションを指定する。

【0050】

実施形態の第 2 の態様が、エンコーダによって実施される方法を規定する。本方法は、コード化ビデオストリーム (CVS) を生成することを含み、CVS は、セグメントアドレスの第 1 の部分を表す 1 つまたは複数の値の第 1 のセットを符号化する 1 つまたは複数のコードワードの第 1 のセットを含み、CVS は、セグメントアドレスの第 2 の部分を表す 1 つまたは複数の値の第 2 のセットを符号化する 1 つまたは複数のコードワードの第 2 のセットを含み、セグメントアドレスは、ピクチャ内のセグメントの空間ロケーションを指定する。

10

【0051】

実施形態の第 3 の態様が、処理回路によって実行されたとき、処理回路に、実施形態の第 1 の態様または第 2 の態様による方法を実施させる命令を備えるコンピュータプログラムを規定する。

【0052】

実施形態の第 4 の態様が、第 3 の態様によるコンピュータプログラムを含んでいるキャリアを規定し、キャリアは、電子信号、光信号、無線信号、およびコンピュータ可読記憶媒体のうちの 1 つである。

20

【0053】

実施形態の第 5 の態様が、実施形態の第 1 の態様による方法を実施するように適応された復号装置を規定する。

【0054】

実施形態の第 6 の態様が、実施形態の第 2 の態様による方法を実施するように適応された符号化装置を規定する。

【0055】

利点

30

【0056】

実施形態の利点は、実施形態が、一実施形態において、スライスヘッダ中で 2 つの値をシグナリングすることによってサブピクチャ中のスライスの相対位置を導出するためのマルチステッププロセスを簡略化することであり、一方は、サブピクチャに対するスライスアドレスであり、他方の値は、スライスがどのサブピクチャに属するかに関する情報、たとえば、サブピクチャとのスライスの固定関係を活用するサブピクチャ ID、およびサブピクチャエリア中のスライスの空間位置を復号プロセスに提供して、サブビットストリーム抽出およびマージを簡略化する。サブピクチャ中に複数のスライスがある場合、現在の VVC 設計を使用するビットストリーム抽出は、各スライス中の `slice_address` 値についてアドレス間接参照を必要とし得る。提案される実施形態を使用して、アドレス間接参照は、代わりにサブピクチャごとに行われ、これは、アドレス間接参照が、サブピクチャ中のすべてのスライスについて 1 回のみ行われることを意味する。

40

【図面の簡単な説明】

【0057】

【図 1】一実施形態による、システムを示す図である。

【図 2】一実施形態による、ビデオエンコーダの概略ブロック図である。

【図 3】一実施形態による、ビデオデコーダの概略ブロック図である。

【図 4】一実施形態による、符号化ビデオビットストリームを示す図である。

【図 5】階層区分を示す図である。

【図 6】一実施形態による、復号プロセスを示すフローチャートである。

50

【図 7】一実施形態による、符号化プロセスを示すフローチャートである。

【図 8】一実施形態による、装置のブロック図である。

【図 9 A】区分の一例を示す図である。

【図 9 B】2 ステップタイル区分を示す図である。

【発明を実施するための形態】

【0058】

図 1 は、例示的な一実施形態による、システム 100 を示す。システム 200 は、ネットワーク 110（たとえば、インターネットまたは他のネットワーク）を介してデコーダ 204 と通信しているエンコーダ 202 を含む。デブロッキングは、エンコーダ 202 とデコーダ 204 の両方において実施され得る。本明細書で説明される実施形態は、ビデオエンコーダ 102 またはビデオデコーダ 104 において使用され得る。

10

【0059】

図 2 は、一実施形態による、ビデオエンコーダ 102 の概略ブロック図である。同じフレーム中のまたは前のフレーム中のピクセルのすでに提供されたブロックから、動き推定器 250 を使用して動き推定を実施することによって、ピクセルの現在ブロックが予測される。動き推定の結果は、インター予測の場合、参照ブロックに関連する動きベクトルまたは変位ベクトルである。動きベクトルは、ピクセルのブロックのインター予測を出力するために動き補償器 250 によって使用され得る。イントラ予測器 249 が、ピクセルの現在ブロックのイントラ予測を算出する。動き推定器 / 補償器 250 からの出力と、イントラ予測器 249 からの出力とは、ピクセルの現在ブロックについてイントラ予測または

インター予測のいずれかを選択するセレクタ 251 中に入力される。セレクタ 251 からの出力は、加算器 241 の形態の誤差計算器に入力され、加算器 241 は、ピクセルの現在ブロックのピクセル値をも受信する。加算器 241 は、ピクセルのブロックとそのブロックの予測との間のピクセル値の差として残差誤差を計算および出力する。誤差は、離散コサイン変換などによって、変換器 242 中で変換され、量子化器 243 によって量子化され、その後、エントロピーエンコーダなどによるエンコーダ 244 におけるコーディングが続く。インターコーディングでは、推定された動きベクトルも、ピクセルの現在ブロックのコード化表現を生成するためにエンコーダ 244 にもたらされる。また、ピクセルの現在ブロックのための変換および量子化された残差誤差は、元の残差誤差を取り出すために、逆量子化器 245 と逆方向変換器 246 とに提供される。この誤差は、ピクセルの

次のブロックの予測およびコーディングにおいて使用され得るピクセルの参照ブロックを作成するために、動き補償器 250 またはイントラ予測器 249 から出力されたブロック予測に加算器 247 によって加算される。この新しい参照ブロックは、デブロッキングフィルタ 200 によって最初に処理される。次いで、処理された新しい参照ブロックは、フレームバッファ 248 に一時的に記憶され、ここで、処理された新しい参照ブロックは、イントラ予測器 249 および動き推定器 / 補償器 250 にとって利用可能である。

20

30

【0060】

図 3 は、いくつかの実施形態による、ビデオデコーダ 104 のブロック図である。デコーダ 104 は、量子化および変換された残差誤差のセットを得るためにピクセルのブロックの符号化表現を復号するための、エントロピーデコーダなどのデコーダ 361 を含む。これらの残差誤差は、逆量子化器 362 によって量子化解除され、逆方向変換器 363 によって逆方向変換されて、残差誤差のセットを提供する。これらの残差誤差は、加算器 364 によってピクセルの参照ブロックのピクセル値に加算される。参照ブロックは、インター予測が実施されるのかイントラ予測が実施されるのかに応じて、動き推定器 / 補償器 367 またはイントラ予測器 366 によって決定される。それにより、セレクタ 368 が加算器 364 と動き推定器 / 補償器 367 とイントラ予測器 366 とに相互接続される。加算器 364 から出力されたピクセルの得られた復号されたブロックは、デブロッキングフィルタ 300 に入力される。ピクセルのフィルタ処理されたブロックは、デコーダ 104 から出力され、さらに、復号されるべきピクセルの後続のブロックのためのピクセルの参照ブロックとして使用されるようにフレームバッファ 365 に一時的に提供され得る。

40

50

フレームバッファ 365 は、それにより、動き推定器 / 補償器 367 に接続されて、ピクセルの記憶されたブロックを動き推定器 / 補償器 367 にとって利用可能にする。加算器 364 からの出力はまた、イントラ予測器 366 に入力されて、ピクセルのフィルタ処理されていない参照ブロックとして使用され得る。

【0061】

図 4 は、例示的なビデオビットストリーム 400 を示す。ビットストリーム 400 は C V S 401 を含み、C V S 401 は、パラメータセット (P S) 410 (たとえば、パラメータセットを含んでいる非 V C L N A L ユニット) と、いくつかのセグメント (たとえば、V V C スライスを含んでいるいくつかの V C L N A L ユニット) とを備える。セグメント 412 a および 412 b が示されている。セグメントは、セグメントデータ (S D) を備えるデータのユニットであり、S D はサンプルデータを備える。セグメントは、セグメントデータ (S D) に加えてセグメントヘッダ (S H) を有し得る。V V C スライスおよび H E V C スライスがセグメントの例である。セグメントはまた、ピクチャ、タイルグループ、あるいは完全なピクチャまたはピクチャの一部を備える何らかの他のエンティティであり得る。この例では、各セグメントは、セグメントデータに加えてセグメントヘッダを含む。

【0062】

階層区分の場合が図 5 に示されており、ここで、ピクチャ 502 が、太い線で示されている大粒区分ブロック (l a r g e g r a i n p a r t i t i o n b l o c k) (たとえば、V V C サブピクチャ) に区分され (たとえば、ブロック 511)、薄い点線が、大粒区分ブロック内の小粒区分ブロック (s m a l l g r a i n p a r t i t i o n b l o c k) (たとえば、V V C スライス) を示す (たとえば、ブロック 511 中に空間的に位置するブロック 512 参照)。いくつかの実施形態では、そのような階層区分の場合、小粒区分ブロック (たとえば、ブロック 512) のヘッダまたはパラメータセット中で、少なくとも 2 つの値、すなわち、i) 小粒区分がどの大粒区分ブロック中に空間的に位置するか (たとえば、ブロック 511) を指定する 1 つの値と、i i) 大粒区分ブロックの位置に対する小粒区分ブロックのアドレスを提供するための 1 つの値とがシグナリングされる。V V C スライスは、小粒区分ブロックの一例であり、V V C サブピクチャは、大粒区分ブロックの一例である。

【0063】

以下の実施形態が組み合わせられて、明示的に規定されないが、依然として本開示によってカバーされるソリューションを形成し得ることが、当業者によって理解されるべきである。また、以下で説明される実施形態は、スライス (たとえば、小粒区分ブロック) およびサブピクチャ (たとえば、大粒区分ブロック) に関して説明され得る。すなわち、スライスおよびサブピクチャという用語は、それぞれ、小粒区分ブロックおよび大粒区分ブロックと互換的に使用される。また、実施形態がスライスに関して説明されるが、本発明は、スライスに限定されず、他のセグメントをカバーするものとする。

【0064】

1. スライスヘッダ中でスライスアドレスについてシグナリングされる 2 つの値

【0065】

第 1 の実施形態では、2 つの値、すなわち、i) 小粒区分ブロックが空間的に中に位置する大粒区分ブロックを示す第 1 の値、たとえば、I D と、i i) 大粒区分ブロックの位置に対する小粒区分ブロックの配置を示す第 2 の値とが、スライスのヘッダまたはパラメータセット中でシグナリングされる。この実施形態のための一例として、スライスアドレスをともに形成する 2 つの値、すなわち、i) スライスが属するサブピクチャ (すなわち、スライスが中に位置するサブピクチャ) を示す、サブピクチャ I D についての第 1 の値と、i i) スライスが属するサブピクチャ位置に対するスライスの空間配置を示す、ローカルスライスアドレスについての 1 つの値とが、スライスヘッダ中でシグナリングされる。以下は、スライスヘッダについての例示的なシンタックスおよびセマンティクスである (すべての例示的なシンタックスおよびセマンティクスは、V V C ドラフト仕様の現在の

10

20

30

40

50

バージョンに加えてテキストとして与えられることに留意されたい)。

表6

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1){	
if (subpics_present_flag)	
subpic_id	u(v)
local_slice_address	u(v)
}	
...	
}	

10

【0066】

subpic_idコードワード(別名、シンタックスエレメント)は、スライスが属するサブピクチャのIDを指定する。subpic_idコードワードは、subpics_present_flagを条件として、表中にあり、subpics_present_flagは、ピクチャ中にサブピクチャがあるとき、真であり(1に等しく)、サブピクチャがないとき、偽である(0に等しい)。subpic_idが偽である場合、local_slice_addressコードワードは、サブピクチャではなくピクチャに対するスライスの空間配置を指定する。Subpic_idの存在に対する他の条件が可能であり、条件がないことがあり、これは、local_slice_addressが存在するとき、subpic_idが常に存在することを意味することに留意されたい。存在しないとき、subpic_idの値は0に等しいと推論される。シンタックスエレメントの長さは、Ceil(Log2(N))ビットである。VVCの現在のバージョンでは、範囲0~254内にあり得るmax_subpics_minus_1をシグナリングするために8ビットがSPS中で使用されることに留意されたい。その場合、Nは、たとえば254であり得る。

20

【0067】

local_slice_addressコードワードは、subpic_idによって識別されるサブピクチャ中のスライスのスライスアドレスを指定する。存在しないとき、local_slice_addressの値は0に等しいと推論される。シンタックスエレメントの長さは、Ceil(Log2(max_num_slices_in_picture_minus_1+1))ビットであり、ここで、max_num_slices_in_picture_minus_1+1は、使用中のプロファイル、ティア、またはレベル規定によって可能にされるスライスの最大数である。

30

【0068】

local_slice_addressについての代替セマンティクスは、以下のようになる。

【0069】

local_slice_addressコードワードは、スライスのアドレスを指定する。存在しないとき、local_slice_addressの値は0に等しいと推論される。サブピクチャが可能にされない(subpics_present_flagが0に等しい)場合、以下が適用される。1)スライスアドレスはブリックIDであり、2)slice_addressの長さは、Ceil(Log2(NumBricksInPic))ビットであり、3)slice_addressの値は、両端値を含む0~NumBricksInPic-1の範囲内にあるものとする。そうではなく、サブピクチャが可能にされる(subpics_present_flagが1に等しい)場合、以下が適用される。1)スライスアドレスは、subpic_idをもつサブピクチャ中のスライスのスライスアドレスであり、2)slice_addressの長さはsignalled_slice_id_length_minus_1+1ビットに等しい。

40

50

【 0 0 7 0 】

デコーダは、ビットストリームから1つまたは複数のピクチャを復号するために、この実施形態では、以下のステップを実施し得、ここで、ビットストリームは、少なくとも2つのスライスを含む。

【 0 0 7 1 】

1) ビットストリーム中の1つまたは複数のシンタックスエレメントから、区分構造が階層の2つ以上のレベルを有するかどうかを決定する。

【 0 0 7 2 】

2) 階層の2つ以上のレベルがある場合のスライスについて、以下を行う。2 a) スライスについてスライスヘッダ中のコードワードから第1の値を復号し、ここで、第1の値はアドレスの第1の部分を表し、2 b) スライスヘッダ中のコードワードから第2の値を復号し、ここで、第2の値はアドレスの第2の部分を表し、2 c) 第1の値と第2の値とから、ピクチャ内のスライスの位置を特定するスライスアドレスを導出し、2 d) スライスを復号するためにスライスアドレスを使用する。

【 0 0 7 3 】

別のバージョンでは、スライスのヘッダまたはパラメータセット中で値の2つのセットがシグナリングされ、ここで、各セットが1つまたは複数の値を含み得、セットのうちの1つ中の1つまたは複数の値は、サブピクチャの位置に対するスライスの配置をまとめて示し、別のセット中の1つまたは複数の値は、スライスがどのサブピクチャ中に空間的に位置するかをまとめて示す。このバージョンのための一例として、スライスアドレスについてスライスヘッダ中で、2つの値セットがシグナリングされ、1つの値セットは、スライスがどのサブピクチャに属するかを示すサブピクチャIDについての1つの値を含み、1つの値セットは、スライスが属するサブピクチャ位置に対するスライスの空間配置をまとめて示す、2つの値 X_s および Y_s を含む。

【 0 0 7 4 】

2 - 間接参照を使用すること

【 0 0 7 5 】

別の実施形態では、小粒区分ブロックのヘッダまたはパラメータセット中で、2つの値がシグナリングされ、i) 一方の値は、小粒区分ブロックが空間的に中に位置する大粒区分ブロックを示し、ii) 他方の値は、大粒区分ブロックの位置に対する小粒区分ブロックの配置を示し、2つの値のうち少なくとも1つは、ターゲットにされる値を指定するために、たとえば、ビットストリーム中のパラメータセット、たとえば、PPSまたはSPS中でシグナリングされ得るインデックスマッピングリストまたはアドレスマッピングリストを使用する間接参照機構を使用する。好ましくは、この実施形態では、大粒区分ブロックは、間接参照機構を使用するものである。

【 0 0 7 6 】

たとえば、ピクチャが、4つの空間象限 (*s p a t i a l q u a d r a n t*) にスプリットされると仮定し、ここで、各象限はサブピクチャである。4つのサブピクチャの各々が、各々1つのスライスのみからなるとさらに仮定する。この例では、すべての第2の値 (たとえば、*l o c a l _ s l i c e _ a d d r e s s* 値) は、スライスの位置がサブピクチャの位置に等しいことを示すために、0に等しくなり得る。第1のID値 (たとえば、*s u b p i c _ i d* 値) は、各スライスが属するサブピクチャを示すために、それぞれ、0、1、2、3に等しくなり得る。次に、サブピクチャ2および3がビットストリームから抽出され、それらの2つのサブピクチャからなる新しいビットストリームが作成されることを考える。そのような動作をサポートするために、たとえば、PPSは、ID値2および3が、それぞれ0および1にマッピングされる、間接参照またはインデックスマッピングを含んでいることがある。新しいビットストリームを復号するデコーダは、最初に、新しいビットストリーム中に2つのサブピクチャがあることを復号し、したがって、最終サブピクチャID 0および1をそれらのサブピクチャに割り振り得る。次いで、デコーダは、インデックスマッピングを作成するために、PPS中の情報を復号することにな

10

20

30

40

50

る。その後、デコーダは、2のID値をもつスライスを復号し得る。インデックスマッピングを使用して、0に等しい最終サブピクチャID値が導出される。同様に、3のID値をもつスライスについて、最終サブピクチャID値は、1に等しいものとして導出される。この間接参照またはインデックスマッピング機構によって、サブピクチャデータを抽出し、各スライス中のスライスID値を書き直すことなしに新しいビットストリームを形成するが、代わりにインデックスマッピングを1回のみ作成することが可能である。

【0077】

3 - 2つ以上のレベルの区分階層についてのアドレスをシグナリングすること

【0078】

別の実施形態では、3つ以上のレベルの区分階層、たとえば、小粒区分ブロックと中粒区分ブロック (medium grain partition block) と大粒区分ブロックとをもつ3レベル区分階層が存在し、小粒区分ブロックのヘッダまたはパラメータセット中で、少なくとも3つの値、すなわち、i) 小粒区分ブロックが空間的に中に位置する中粒区分ブロックを示す第1の値、たとえば、IDと、ii) 中粒区分ブロックの位置に対する小粒区分ブロックの配置を示す第2の値と、iii) 小粒区分ブロックが空間的に中に位置する大粒区分ブロックを示す第3の値とがシグナリングされる。いくつかの実施形態では、ヘッダは、大粒区分ブロックの位置に対する小粒区分ブロックの配置を示す第4の値をも含む。この実施形態では、大粒ブロック区分に対する中粒区分ブロックの空間ロケーションは、中粒区分ブロックと大粒区分ブロックとに対する小粒区分ブロックの空間位置の差から導出される。

【0079】

4 - サブピクチャ中のローカルスライスの数のシグナリング

【0080】

前の実施形態のいずれかに基づき得る別の実施形態では、現在サブピクチャ中のスライスの数は、スライスを復号するとき知られる。この情報は、各サブピクチャについてのスライスヘッダ中で直接またはパラメータセット中で直接、たとえば、num_slices_in_subpicまたはnum_slices_in_subpic_minus1コードワードを用いてシグナリングされ得る。以下の例は、VVCの現在のバージョンに加えて、スライスヘッダ中でnum_slices_in_subpic_minus1をシグナリングするためのシンタックスおよびセマンティクスを説明する。

表7

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(rect_slice_flag NumBricksInPic > 1){	
slice_address	u(v)
subpic_id	u(v)
num_slices_in_subpic_minus1	ue(v)
local_slice_address	u(v)
}	
...	
}	

【0081】

subpic_idコードワードは、スライスが属するサブピクチャのIDを指定する。存在しないとき、subpic_idの値は0に等しいと推論される。シンタックスエレメントの長さは、Ceil(Log2(N))ビットである。VVCの現在のバージョンでは、範囲0~254内にあり得るmax_subpics_minus1をシグナリングするために8ビットがSPS中で使用されることに留意されたい。その場合、Nは、たとえば254であり得る。

【0082】

10

20

30

40

50

num_slices_in_subpic_minus1コードワードは、現在サブピクチャ中に存在するスライスの数（すなわち、num_slices_in_subpic_minus1+1）を示す。存在しないとき、num_slices_in_subpic_minus1の値は0に等しいと推論される。

【0083】

local_slice_addressコードワードは、subpic_idをもつサブピクチャ中のスライスのスライスアドレスを指定する。存在しないとき、local_slice_addressの値は0に等しいと推論される。シンタックスエレメントの長さは、Ceil(Log2(num_slices_in_subpic_minus1+1))ビットである。

10

【0084】

以下の例は、VVCの現在のバージョンに加えて、SPS中の各サブピクチャについて、num_slices_in_subpic_minus1[i]をシグナリングするためのシンタックスおよびセマンティクスを説明する。

表8

seq_parameter_set_rbsp() {	記述子
...	
subpics_present_flag	u(1)
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
...	
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
num_slices_in_subpic_minus1[i]	u(v)
}	
}	
}	

20

【0085】

max_subpics_minus1+1の値は、CVS中に存在し得るサブピクチャの最大数を指定する。max_subpics_minus1は0~254の範囲内にあるものとする。255の値は、ITU-T | ISO / IECによる将来の使用のために予約される。

30

【0086】

num_slices_in_subpic_minus1[i]+1の値は、i番目のサブピクチャ中に存在するスライス数を指定する。存在しないとき、num_slices_in_subpic_minus1[i]の値は0に等しいと推論される。

【0087】

実施形態5 - subpic_idを導出するときにmax_subpic_minus1を使用すること

40

【0088】

第1の実施形態に基づき得る別の実施形態では、VVCの現在のバージョンにおいてSPS中でシグナリングされるmax_subpics_minus1コードワードは、subpic_idのために使用されるビット数を導出するために使用される。スライスヘッダ中のsubpic_idについてのセマンティクスは、その場合、subpic_idが、スライスが属するサブピクチャのIDを指定することであり得る。存在しないとき、subpic_idの値は0に等しいと推論される。シンタックスエレメントの長さは、Ceil(Log2(max_subpics_minus1+1))ビットである。

【0089】

50

6 - サブピクチャごとに1つのスライスをシグナリングすること

【0090】

一実施形態では、フラグ `single_slice_in_subpicture_flag` が、パラメータセット、好ましくはSPSまたはDPS中に存在する。このフラグがある値を有するとき、2つ以上のスライスからなるサブピクチャがないものとする。このフラグが別の値を有するとき、サブピクチャ中に複数のスライスがあり得る。

【0091】

`slice_address`コードワードの存在は、各サブピクチャ中に1つのスライスがあることをフラグが示すとき、`slice_address`コードワードがパースされないように、このフラグを条件とし得る。

10

表9

<code>seq_parameter_set_rbsp() {</code>	記述子
<code>...</code>	
<code>subpics_present_flag</code>	<code>u(1)</code>
<code>if(subpics_present_flag) {</code>	
<code> max_subpics_minus1</code>	<code>u(8)</code>
<code> single_slice_in_subpicture_flag</code>	<code>u(1)</code>
<code> ...</code>	
<code>}</code>	
<code>}</code>	

20

【0092】

`single_slice_in_subpicture_flag`の値が1に等しいとき、これは、SPSを参照するCVS中の各サブピクチャ中に1つのスライスのみがあることを指定する。`single_tile_in_pic_flag`の値が0に等しいとき、これは、SPSを参照するCVS中のサブピクチャ中に2つ以上のスライスがあり得ることを指定する。`single_slice_in_subpicture_flag`が存在しないとき、`single_slice_in_subpicture_flag`は0に等しいと推論される。

表10

<code>pic_parameter_set_rbsp() {</code>	記述子
<code>...</code>	
<code>if(rect_slice_flag subpics_present_flag) {</code>	
<code> signalled_slice_id_flag</code>	<code>u(1)</code>
<code> if(signalled_slice_id_flag) {</code>	
<code> signalled_slice_id_length_minus1</code>	<code>ue(v)</code>
<code> for(i = 0; i <= num_slices_in_pic_minus1; i++)</code>	
<code> slice_id[i]</code>	<code>u(v)</code>
<code> }</code>	
<code>}</code>	

30

40

【0093】

`signalled_slice_id_length_minus1 + 1`は、スライスヘッダ中で、存在するときにシンタックスエレメント `slice_id[i]`を表し、シンタックスエレメント `slice_address`を表すために使用されるビット数を指定する。`signalled_slice_id_length_minus1`の値は、両端値を含む0 ~ 15の範囲内にあるものとする。存在しないとき、`signalled_slice_id_length_minus1`の値は、 $Ceiling(Log_2(Max(2, num_slices_in_pic_minus1 + 1))) - 1$ に等しいと推論される。

50

表11

slice_header() {	記述子
slice_pic_parameter_set_id	ue(v)
if(subpics_present_flag)	
subpic_id	u(v)
if(rect_slice_flag NumBricksInPic > 1)	
if(!single_slice_in_subpicture_flag NumBricksInPic > 1)	
slice_address	u(v)
...	
}	

10

【0094】

subpic_idは、スライスが属するサブピクチャのIDを指定する。存在しないとき、subpic_idの値は0に等しいと推論される。シンタックスエレメントの長さは、 $\text{Ceil}(\text{Log}_2(\text{max_subpics_minus}1+1))$ ビットである。

【0095】

slice_addressはスライスのアドレスを指定する。存在しないとき、slice_addressの値は0に等しいと推論される。

【0096】

サブピクチャが可能にされない(subpics_present_flagが0に等しい)場合、以下が適用される。1)スライスアドレスはブリックIDであり、2)slice_addressの長さは、 $\text{Ceil}(\text{Log}_2(\text{NumBricksInPic}))$ ビットであり、3)slice_addressの値は、両端値を含む0~NumBricksInPic-1の範囲内にあるものとする。

20

【0097】

そうではなく、サブピクチャが可能にされる(subpics_present_flagが1に等しい)場合、以下が適用される。1)スライスアドレスは、subpic_idに等しいサブピクチャIDをもつサブピクチャ内のスライスのスライスアドレスであり、2)slice_addressの長さは、signalled_slice_id_length_minus1+1ビットである。

30

【0098】

代替的に、サブピクチャごとのスライスの最大数max_number_of_slices_per_subpic_minus1コードワードは、パラメータセット中でシグナリングされ得る。この場合、slice_addressコードワードは、デコーダによってパースされないが、max_number_of_slices_per_subpic_minus1が0に等しい場合、0に等しいと推論される。max_number_of_slices_per_subpic_minus1が0よりも大きい場合にslice_addressのために使用すべきビット数は、 $\text{Ceil}(\text{Log}_2(\text{max_number_of_slices_per_subpic_minus}1+1))$ ビットに等しくセットされ得る。

40

【0099】

図8は、ビデオエンコーダ102またはビデオデコーダ104を実装するための、いくつかの実施形態による、装置800のブロック図である。すなわち、装置800は、プロセス600および/またはプロセス700を実施するように動作可能である。装置800がビデオエンコーダ102を実装する実施形態では、装置800は「符号化装置800」と呼ばれることがあり、装置800がビデオデコーダ104を実装する実施形態では、装置800は「復号装置800」と呼ばれることがある。図8に示されているように、装置800は、1つまたは複数のプロセッサ(P)855(たとえば、汎用マイクロプロセッサ、および/または、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲ

50

ートアレイ（FPGA）など、1つまたは複数の他のプロセッサなど）を含み得る処理回路（PC）802であって、そのプロセッサが、単一のハウジングにおいてまたは単一のデータセンタにおいて共同サイト式であり得るかあるいは地理的に分散され得る（すなわち、装置800が分散コンピューティング装置であり得る）、処理回路（PC）802と、ネットワークインターフェース848であって、装置800が、ネットワークインターフェース848が（直接または間接的に）接続されるネットワーク110（たとえば、インターネットプロトコル（IP）ネットワーク）に接続された他のノードにデータを送信し、他のノードからデータを受信することを可能にするための送信機（Tx）845および受信機（Rx）847を備える（たとえば、ネットワークインターフェース848はネットワーク110に無線で接続され得、その場合、ネットワークインターフェース848はアンテナ構成に接続される）、ネットワークインターフェース848と、1つまたは複数の不揮発性記憶デバイスおよび/または1つまたは複数の揮発性記憶デバイスを含み得るローカル記憶ユニット（別名「データ記憶システム」）808とを備え得る。PC802がプログラマブルプロセッサを含む実施形態では、コンピュータプログラム製品（CPP）841が提供され得る。CPP841はコンピュータ可読媒体（CRM）842を含み、CRM842は、コンピュータ可読命令（CRI）844を備えるコンピュータプログラム（CP）843を記憶する。CRM842は、磁気媒体（たとえば、ハードディスク）、光媒体、メモリデバイス（たとえば、ランダムアクセスメモリ、フラッシュメモリ）など、非一時的コンピュータ可読媒体であり得る。いくつかの実施形態では、コンピュータプログラム843のCRI844は、PC802によって実行されたとき、CRIが、装置800に、本明細書で説明されるステップ（たとえば、フローチャートを参照しながら本明細書で説明されるステップ）を実施させるように設定される。他の実施形態では、装置800は、コードの必要なしに本明細書で説明されるステップを実施するように設定され得る。すなわち、たとえば、PC802は、単に1つまたは複数のASICからなり得る。したがって、本明細書で説明される実施形態の特徴は、ハードウェアおよび/またはソフトウェアで実装され得る。

【0100】

様々な実施形態が（添付の書類を含む）本明細書で説明されたが、それらの実施形態は、限定ではなく、例として提示されたにすぎないことを理解されたい。したがって、本開示の広さおよび範囲は、上記で説明された例示的な実施形態のいずれによっても限定されるべきでない。その上、本明細書で別段に指示されていない限り、またはコンテキストによって明確に否定されていない限り、上記で説明されたエレメントのそれらのすべての考えられる変形形態における任意の組合せが、本開示によって包含される。

【0101】

さらに、上記で説明され、図面に示されたプロセスは、ステップのシーケンスとして示されたが、これは、説明のためにのみ行われた。したがって、いくつかのステップが追加され得、いくつかのステップが省略され得、ステップの順序が並べ替えられ得、いくつかのステップが並行して実施され得ることが企図される。

【0102】

略語 説明

A T S C 新型テレビジョンシステム委員会
 A U アクセスユニット
 A U D アクセスユニットデリミタ
 A L F 適応ループフィルタ
 A P S 適応パラメータセット
 B L A 切断リンクアクセス
 C L V S コード化レイヤビデオシーケンス
 C R A クリーンランダムアクセス
 C V S コード化ビデオストリーム
 C V S S C V S 開始

10

20

30

40

50

C U	コーディングユニット	
D A S H	動的適応ストリーミングオーバーH T T P	
D P S	復号パラメータセット	
D V B	デジタルビデオブロードキャストイング	
D R A P	依存ランダムアクセスポイント	
G D R	漸進的復号リフレッシュ	
H E V C	高効率ビデオコーディング	
I D R	瞬時復号リフレッシュ	
I R A P	イントラランダムアクセスポイント	
I S O	国際標準化機構	10
I S O B M F F	I S Oベースメディアファイルフォーマット	
L M C S	ルーママッピングおよびクロマスケーリング	
M P E G	モーションピクチャエキスパートグループ	
M M T	M P E Gメディアトランスポート	
N A L	ネットワークアブストラクションレイヤ	
N A L U	N A Lユニット	
N U T	N A Lユニットタイプ	
P P S	ピクチャパラメータセット	
R A D L	ランダムアクセス復号可能リーディング	
R A P	ランダムアクセスポイント	20
R A S L	ランダムアクセススキップリーディング	
R B S P	ローバイトシーケンスペイロード	
R P L	参照ピクチャリスト	
S E I	補足エンハンスメントレイヤ	
S P S	シーケンスパラメータセット	
S T S A	ステップワイズ時間レイヤアクセス	
V C L	ビデオコーディングレイヤ	
V P S	ビデオパラメータセット	
V V C	多用途ビデオコーディング	
【 0 1 0 3 】		30
付録		
	以下のテキストは、V V Cの現在のバージョンの変更を提案する寄与文書からのものである。	
【 0 1 0 4 】		
	テキスト始まり	
要約		
	この寄与文書は、サブピクチャの場合のスライスアドレスシグナリングに関係するV V C仕様の以下の変更を提案する。	
	- 第1に、C V S中のサブピクチャの存在を条件として、現在スライスがどのサブピクチャに属するかを指定するためにスライスヘッダ中でサブピクチャI Dをシグナリングすることが提案される。	40
	- 第2に、スライスヘッダ中で、サブピクチャ位置に対するスライスアドレスをシグナリングすることが提案される。	
	- 第3に、サブピクチャI Dについてのアドレス間接参照を使用し、スライスアドレスについてのアドレス間接参照を削除することが提案される。各サブピクチャについて、スライスアドレスは、サブピクチャに対して固定され、サブビットストリーム抽出およびマージプロセス中に再使用され得る。	
1 . 序論		
	J V E T - O 2 0 0 1 - v Eにおける現在のV V C仕様ドラフトでは、サブピクチャがサポートされ、サブビットストリーム抽出およびマージプロセスを簡略化することをターゲット	50

ットにする。しかしながら、ピクチャおよびスライスなど、他の規定された階層区分に関する、サブピクチャのためのアドレスシグナリング機構は、改善を必要とし得る。

現在のVVC仕様ドラフトでは、スライスアドレスは、スライスヘッダ中でシグナリングされ、ピクチャ中のスライスの空間位置を導出するために使用される。しかしながら、サブピクチャが使用されるとき、およびサブピクチャ中に2つ以上のスライスがあるとき、現在のスライスアドレスシグナリング方式に関する数個の問題がある。

1 - サブピクチャ中のスライスの空間位置は、スライスヘッダ中の `slice_address` シンタックスから直接導出され得ず、その空間位置は、マルチステッププロセスを必要とする。

○ ピクチャ中のスライスの空間位置が、最初に導出される必要がある。

○ 次いで、第2のステップにおいて、ピクチャ中のその空間位置がどのサブピクチャに属するかが導出される必要がある。

○ 次いで、第3のステップにおいて、そのサブピクチャ中のスライスの空間位置が導出され得る。

2 - スライスヘッダから、このスライスがどのサブピクチャに属するかが導出され得ない。この情報は、サブビットストリームマージおよび抽出プロセスにとって有用であることになる。

3 - サブピクチャ中のスライスの固定相対空間位置は、(サブビットストリーム抽出およびマージの場合のように)サブピクチャが抽出またはマージされているとき、活用されない。

4 - `slice_address` を `slice_id` にマッピングするために使用される間接参照機構は、サブビットストリーム抽出およびマージプロセスにとって準最適であり得、なぜなら、サブピクチャ中の複数のスライスの場合、現在のVVC設計を使用するサブビットストリーム抽出が、いくつかのアドレス間接参照、すなわち、各スライス中の `slice_address` 値について1つの間接参照を必要とし得るからである。

2. 提案

この寄与文書は、上記の問題を解決し、サブピクチャ中のスライスの相対位置を導出するためのマルチステッププロセスを簡略化するための、ソリューションを提案する。この寄与文書は、サブピクチャの場合のスライスアドレスシグナリングに関係する以下の変更を提案する。

- 第1に、CVS中のサブピクチャの存在に条件付けられる、現在スライスがどのサブピクチャに属するかを指定するためにスライスヘッダ中でサブピクチャIDをシグナリングすることが提案される。

- 第2に、スライスヘッダ中で、サブピクチャ位置に対するスライスアドレスをシグナリングすることが提案される。

- 第3に、サブピクチャIDについてのアドレス間接参照を使用し、スライスアドレスについてのアドレス間接参照を削除することが提案される。各サブピクチャについて、スライスアドレスは、サブピクチャに対して固定され、サブビットストリーム抽出およびマージプロセス中に再使用され得る。

-

この提案では、4つの前述の問題が、以下のようにして解決される。

1 - サブピクチャ中のスライスの空間位置は、スライスヘッダから直接導出される。

2 - スライスが属するサブピクチャのIDは、スライスヘッダ中でシグナリングされる。

3 - サブピクチャ中のスライスの相対空間位置は、スライスヘッダ中でシグナリングされる。

4 - 間接参照プロセスは、サブピクチャの抽出およびマージにおいて(スライスごとではなく)サブピクチャごとに行われる。

以下は、JVET-O2001-vEに加える、スライスヘッダの提案されるシンタックスおよびセマンティクス変更である。

10

20

30

40

50

	記述子
seq_parameter_set_rbsp() {	
...	
if(subpics_present_flag) {	
max_subpics_minus1	u(8)
subpic_grid_col_width_minus1	u(v)
subpic_grid_row_height_minus1	u(v)
for(i = 0; i < NumSubPicGridRows; i++)	
for(j = 0; j < NumSubPicGridCols; j++)	
subpic_grid_idx[i][j]	u(v)
for(i = 0; i <= NumSubPics; i++) {	
subpic_treated_as_pic_flag[i]	u(1)
loop_filter_across_subpic_enabled_flag[i]	u(1)
}	
signalled_subpic_id_flag	u(1)
if(signalled_subpic_id_flag) {	
signalled_subpic_id_length_minus1	ue(v)
for(i = 0; i < NumSubPics; i++)	
subpic_id[i]	u(v)
}	
}	
...	
}	

10

20

$max_subpics_minus1 + 1$ は、CVS中に存在し得るサブピクチャの最大数を指定する。 $max_subpics_minus1$ は0～254の範囲内にあるものとする。255の値は、ITU-T | ISO/IECによる将来の使用のために予約される。

$subpic_grid_col_width_minus1 + 1$ は、4つのサンプルの単位でサブピクチャ識別子グリッドの各エレメントの幅を指定する。シンタックスエレメントの長さは、 $Ceiling(\log_2(pic_width_max_in_luma_samples / 4))$ ビットである。

30

変数NumSubPicGridColsは、以下のように導出される。

$$NumSubPicGridCols = (pic_width_max_in_luma_samples + subpic_grid_col_width_minus1 * 4 + 3) / (subpic_grid_col_width_minus1 * 4 + 4) \quad (7-5)$$

$subpic_grid_row_height_minus1 + 1$ は、4つのサンプルの単位でサブピクチャ識別子グリッドの各エレメントの高さを指定する。シンタックスエレメントの長さは、 $Ceiling(\log_2(pic_height_max_in_luma_samples / 4))$ ビットである。

40

変数NumSubPicGridRowsは、以下のように導出される。

$$NumSubPicGridRows = (pic_height_max_in_luma_samples + subpic_grid_row_height_minus1 * 4 + 3) / (subpic_grid_row_height_minus1 * 4 + 4) \quad (7-6)$$

$subpic_grid_idx[i][j]$ は、グリッド位置(i , j)のサブピクチャインデックスを指定する。シンタックスエレメントの長さは、 $Ceiling(\log_2(max_subpics_minus1 + 1))$ ビットである。

変数SubPicTop[subpic_grid_idx[i][j]]、SubPic

50

`cLeft[subpic_grid_idx[i][j]]`、`SubPicWidth[subpic_grid_idx[i][j]]`、`SubPicHeight[subpic_grid_idx[i][j]]`、および`NumSubPics`は、以下のように導出される。

```

NumSubPics = 0
for( i = 0; i < NumSubPicGridRows; i++ ) {
    for( j = 0; j < NumSubPicGridCols; j++ ) {
        if( i == 0 )
            SubPicTop[ subpic_grid_idx[ i ][ j ] ] = 0
        else if( subpic_grid_idx[ i ][ j ] != subpic_grid_idx[ i - 1 ][ j ] ) {
            SubPicTop[ subpic_grid_idx[ i ][ j ] ] = i
            SubPicHeight[ subpic_grid_idx[ i - 1 ][ j ] ] = i - SubPicTop[ subpic_grid_idx[ i - 1 ][ j ] ]
        }
        if( j == 0 )
            SubPicLeft[ subpic_grid_idx[ i ][ j ] ] = 0
        else if( subpic_grid_idx[ i ][ j ] != subpic_grid_idx[ i ][ j - 1 ] ) {
            SubPicLeft[ subpic_grid_idx[ i ][ j ] ] = j
            SubPicWidth[ subpic_grid_idx[ i ][ j ] ] = j - SubPicLeft[ subpic_grid_idx[ i ][ j - 1 ] ]
        }
        if( i == NumSubPicGridRows - 1 )
            SubPicHeight[ subpic_grid_idx[ i ][ j ] ] = i - SubPicTop[ subpic_grid_idx[ i - 1 ][ j ] ] + 1
        if( j == NumSubPicGridRows - 1 )
            SubPicWidth[ subpic_grid_idx[ i ][ j ] ] = j - SubPicLeft[ subpic_grid_idx[ i ][ j - 1 ] ] + 1
        if( subpic_grid_idx[ i ][ j ] > NumSubPics )
            NumSubPics = subpic_grid_idx[ i ][ j ]
    }
}

```

1に等しい`subpic_treated_as_pic_flag[i]`は、CVS中の各コード化ピクチャの*i*番目のサブピクチャがループ内フィルタ処理演算を除く復号プロセスにおいてピクチャとして扱われることを指定する。0に等しい`subpic_treated_as_pic_flag[i]`は、CVS中の各コード化ピクチャの*i*番目のサブピクチャがループ内フィルタ処理演算を除く復号プロセスにおいてピクチャとして扱われないことを指定する。存在しないとき、`subpic_treated_as_pic_flag[i]`の値は0に等しいと推論される。

1に等しい`loop_filter_across_subpic_enabled_flag[i]`は、ループ内フィルタ処理演算がCVS中の各コード化ピクチャ中の*i*番目のサブピクチャの境界にわたって実施され得ることを指定する。0に等しい`loop_filter_across_subpic_enabled_flag[i]`は、ループ内フィルタ処理演算がCVS中の各コード化ピクチャ中の*i*番目のサブピクチャの境界にわたって実施されないことを指定する。存在しないとき、`loop_filter_across_subpic_enabled_pic_flag[i]`の値は1に等しいと推論される。

以下の制約が適用されることが、ビットストリーム適合性の要件である。

- 任意の2つのサブピクチャ`subpicA`および`subpicB`について、`subpicA`のインデックスが`subpicB`のインデックスよりも小さいとき、`subpicA`の任意のコード化NALユニットが、復号順序において`subpicB`の任意のコー

ド化NALユニットに続くものとする。

- サブピクチャの形状は、各サブピクチャが、復号されたとき、ピクチャ境界からなる、または前に復号されたサブピクチャの境界からなる、そのサブピクチャの左境界全体および上境界全体を有するものとするようなものであるものとする。

1に等しい `signalled_subpic_id_flag` は、各サブピクチャについてのサブピクチャIDがシグナリングされることを指定する。0に等しい `signalled_subpic_id_flag` は、サブピクチャIDがシグナリングされないことを指定する。`subpics_present_flag` が0に等しいとき、`signalled_subpic_id_flag` の値は0に等しいと推論される。

10

`signalled_subpic_id_length_minus_1+1` は、スライスヘッダ中で、存在するときにシンタックスエレメント `subpic_id[i]` を表し、シンタックスエレメント `slice_subpic_id` を表すために使用されるビット数を指定する。`signalled_subpic_id_length_minus_1` の値は、両端値を含む0~15の範囲内にあるものとする。存在しないとき、`signalled_subpic_id_length_minus_1` の値は、 $Ceil(\log_2(\text{Max}(2, \text{max_subpics_minus_1}+1))) - 1$ に等しいと推論される。

`subpic_id[i]` は、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]` シンタックスエレメントの長さは、`signalled_subpic_id_length_minus_1+1` ビットである。存在しないとき、`subpic_id[i]` の値は、両端値を含む0~`NumSubPics-1` の範囲内の各iについて、iに等しいと推論される。

20

以下は、JVET-02001-vEに加える、スライスヘッダの提案されるシンタックスおよびセマンティクス変更である。

<code>slice_header() {</code>	記述子
<code> slice_pic_parameter_set_id</code>	ue(v)
<code> if(rect_slice_flag NumBricksInPic > 1){</code>	
<code> if(subpics_present_flag)</code>	
<code> slice_subpic_id</code>	u(v)
<code> slice_address</code>	u(v)
<code> }</code>	
<code> ...</code>	
<code>}</code>	

30

`slice_pic_parameter_set_id` は、使用中のPPSについての `pps_pic_parameter_set_id` の値を指定する。`slice_pic_parameter_set_id` の値は、両端値を含む0~63の範囲内にあるものとする。

40

現在ピクチャのTemporalIdの値が、`slice_pic_parameter_set_id` に等しい `pps_pic_parameter_set_id` を有するPPSのTemporalIdの値よりも大きいかまたはそれに等しいものとするのが、ビットストリーム適合性の要件である。

`slice_subpic_id`は、スライスが空間的に中に位置するサブピクチャについての`subpic_id`の値を指定する。存在しないとき、`slice_subpic_id`の値は0に等しいと推論される。シンタックスエレメントの長さは、 $\text{Ceil}(\text{Log}_2(\text{max_subpics_minus}_1))$ ビットである。

`slice_address`はスライスのスライスアドレスを指定する。存在しないとき、`slice_address`の値は0に等しいと推論される。`subpics_present_flag`が0に等しい場合、`slice_address`は、ピクチャに対するスライスのスライスアドレスを表し、そうでなければ、`subpics_present_flag`が1に等しい場合、`slice_address`は、`slice_subpic_id`に等しいサブピクチャIDをもつサブピクチャに対するスライスのスライスアドレスを表す。

10

`rect_slice_flag`が0に等しい場合、以下が適用される。

- スライスアドレスは、式(7-59)によって指定されるブリックIDである。
- `slice_address`の長さは、 $\text{Ceil}(\text{Log}_2(\text{NumBricksInPic}))$ ビットである。
- `slice_address`の値は、両端値を含む0 ~ $\text{NumBricksInPic} - 1$ の範囲内にあるものとする。

他の場合(`rect_slice_flag`が1に等しい)、以下が適用される。

~~— スライスアドレスは、スライスのスライスIDである。~~

- `slice_address`の長さは、 ~~$\text{signalled_slice_id_length_minus}_1 + 1$~~ $\text{Ceil}(\text{Log}_2(\text{NumBricksInPic} - \text{NumSubpics}))$ ビットである。

~~— `signalled_slice_id_flag`が0に等しい場合、`slice_address`の値は、両端値を含む0 ~ $\text{num_slices_in_pic_minus}_1$ の範囲内にあるものとする。他の場合、`slice_address`の値は、両端値を含む0 ~ $2^{-(\text{signalled_slice_id_length_minus}_1 + 1)} - 1$ の範囲内にあるものとする。~~

20

30

以下の制約が適用されることが、ビットストリーム適合性の要件である。

~~— `slice_address`の値は、同じコード化ピクチャの他のコード化スライスNALユニットの`slice_address`の値に等しくないものとする。~~

- `rect_slice_flag`が0に等しいとき、ピクチャのスライスは、それらの`slice_address`値の昇順のものとする。

- ピクチャのスライスの形状は、各ブリックが、復号されたとき、ピクチャ境界からなる、または(1つまたは複数の)前に復号されたブリックの境界からなる、そのブリックの左境界全体および上境界全体を有するものとするようなものであるものとする。

40

`num_bricks_in_slice_minus_1`は、存在するとき、スライス中のブリックの数 - 1を指定する。`num_bricks_in_slice_minus_1`の値は、両端値を含む0 ~ $\text{NumBricksInPic} - 1$ の範囲内にあるものとする。`rect_slice_flag`が0に等しく、`single_brick_per_slice_flag`が1に等しいとき、`num_bricks_in_slice_minus_1`の値は0に等しいと推論される。`single_brick_per_slice_flag`が1に等しいとき、`num_bricks_in_slice_minus_1`の値は0に等しいと推論される。

現在スライス中のブリックの数を指定する変数`NumBricksInCurrSlice`

50

e と、現在スライス中の i 番目のブリックのブリックインデックスを指定する変数 $SliceBrickIdx[i]$ とは、以下のように導出される。

```

if( rect_slice_flag ) {
  sliceIdx = 0
  while( slice_address != slice_id[ sliceIdx ] )
    sliceIdx++
  subpicIdx = 0
  while( slice_subpic_id != subpic_id[ subpicIdx ] )
    subpicIdx++
  sliceIdx = subpic_id[ subpicIdx ] + slice_address
  NumBricksInCurrSlice = NumBricksInSlice[ sliceIdx ]
  brickIdx = TopLeftBrickIdx[ sliceIdx ]
  for( bIdx = 0; brickIdx <= BottomRightBrickIdx[ sliceIdx ]; brickIdx++ )
    if( BricksToSliceMap[ brickIdx ] == sliceIdx )
      SliceBrickIdx[ bIdx++ ] = brickIdx
} else {
  NumBricksInCurrSlice = num_bricks_in_slice_minus1 + 1
  SliceBrickIdx[ 0 ] = slice_address
  for( i = 1; i < NumBricksInCurrSlice; i++ )
    SliceBrickIdx[ i ] = SliceBrickIdx[ i - 1 ] + 1
}

```

(7-92)

10

テキスト終わり

20

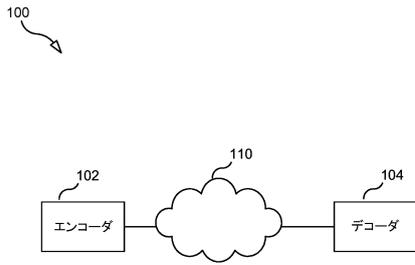
30

40

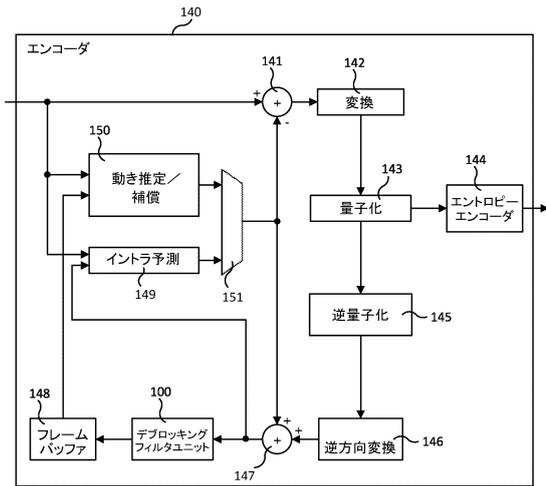
50

【図面】

【図 1】

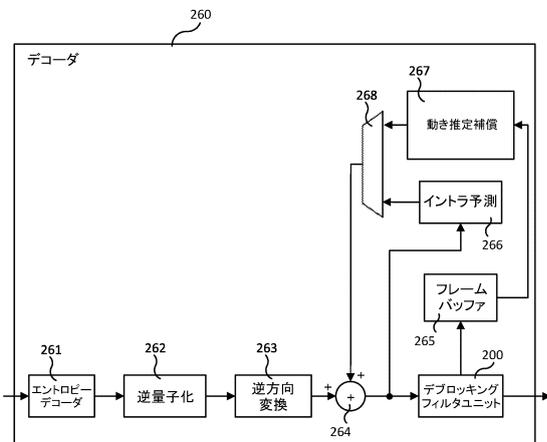


【図 2】



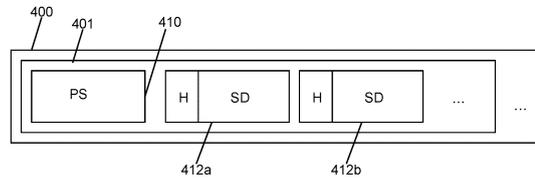
10

【図 3】



30

【図 4】

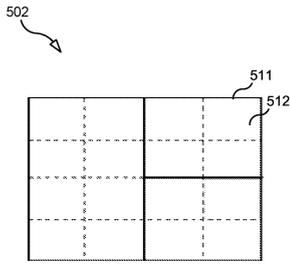


20

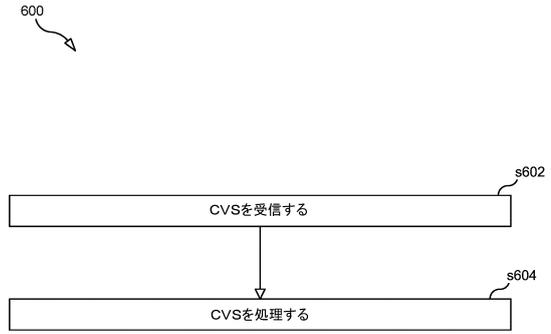
40

50

【図 5】

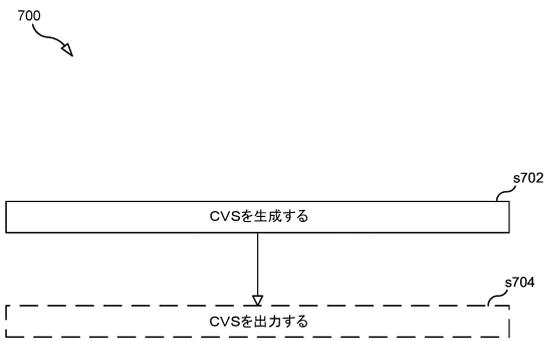


【図 6】

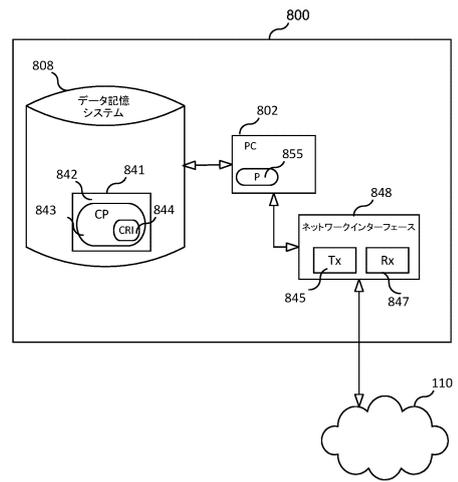


10

【図 7】

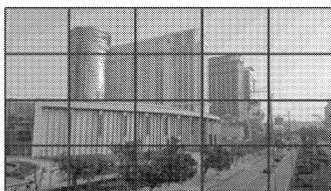


【図 8】

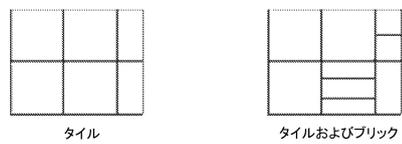


20

【図 9 A】



【図 9 B】



30

40

50

フロントページの続き

- ズ - ブロー , マルコールヴェーゲン 3
(72)発明者 ペッテション , マルティン
スウェーデン国 エスエー - 1 8 6 3 4 ヴァレンチューナ , ステンヴァッラヴェーゲン 1 0
(72)発明者 ショバーリ , リカルド
スウェーデン国 1 1 3 6 2 ストックホルム , サンクト イェリクスガータン 8 4
審査官 久保 光宏
(56)参考文献 特表 2 0 2 2 - 5 4 8 2 2 0 (J P , A)
Ye-Kui Wang, et al. , "Core Experiment Description of Sub-Picture Coding" , Document VCE
G-O57, [online] , ITU - Telecommunications Standardization Sector STUDY GROUP 16 Ques
tion 6 Video Coding Experts Group (VCEG) , 2002年01月16日 , Pages 1-10 , [令和5年5月
25日検索] , インターネット , URL: https://www.itu.int/wftp3/av-arch/video-site/0112_Pat/VCEG-O57.doc .
(58)調査した分野 (Int.Cl. , D B 名)
H 0 4 N 1 9 / 0 0 - 1 9 / 9 8
C S D B (日本国特許庁)
学術文献等データベース (日本国特許庁)
I E E E X p l o r e (I E E E)