

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2019-41402
(P2019-41402A)

(43) 公開日 平成31年3月14日(2019.3.14)

(51) Int.Cl.
H04N 19/70 (2014.01)

F I
H04N 19/70

テーマコード(参考)
5C159

審査請求 有 請求項の数 13 O L (全 68 頁)

(21) 出願番号 特願2018-207464 (P2018-207464)
(22) 出願日 平成30年11月2日(2018.11.2)
(62) 分割の表示 特願2017-19898 (P2017-19898)
の分割
原出願日 平成25年7月1日(2013.7.1)
(31) 優先権主張番号 61/666, 185
(32) 優先日 平成24年6月29日(2012.6.29)
(33) 優先権主張国 米国 (US)

(71) 出願人 515089080
ジーイー ビデオ コンプレッション エ
ルエルシー
アメリカ合衆国 ニューヨーク州 122
11 オールバニー サウスウッズ プー
ルバード 8
(74) 代理人 100079577
弁理士 岡田 全啓
(72) 発明者 シーアル トーマス
ドイツ連邦共和国 10437 ベルリン
ドゥンカーシュトラッセ 72
(72) 発明者 ジョージ ヴァレーリ
ドイツ連邦共和国 10365 ベルリン
ヨーン-ジーク-シュトラッセ 24

最終頁に続く

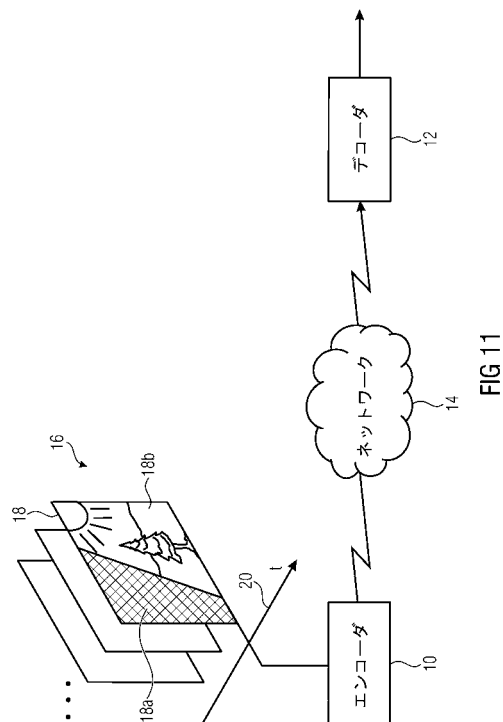
(54) 【発明の名称】 ビデオ・データストリーム・コンセプト

(57) 【要約】 (修正有)

【課題】低い端末間遅延を可能にする際により効率的で
および/または関心領域に、または、特定のタイルにデ
ータストリームの部分の識別がより容易になるビデオ・
データストリーム符号化コンセプトを提供する。

【解決手段】デコーダ検索タイミング情報、ROI情報
およびタイル識別情報は、例えばMANEまたはデコー
ダのようなネットワークエンティティによって簡単なア
クセスを可能にするレベルでビデオ・データストリー
ムの中で伝達される。このようなレベルに達するために、
そのようなタイプの情報がビデオ・データストリームの
アクセスユニットの packets に分散する packets を経
由してビデオ・データストリームの中で伝達される。分散
型の packets は、除去可能な packet ・タイプの中に
あり、すなわち、これらの分散型の packets の除去は、完
全に、ビデオ・データストリームを介して伝送されるビ
デオ内容を回復するデコーダの能力を維持する。

【選択図】 図 11



【特許請求の範囲】**【請求項 1】**

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にしてその中で符号化されるビデオ内容(16)を有し、各サブポーション(24)はビデオ・データストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)にそれぞれ符号化され、パケットのシーケンス(34)はアクセスユニット(30)のシーケンスに分割され、それにより各アクセスユニット(30)がビデオ内容(16)のそれぞれの画像(18)に関するペイロード・パケット(32)を集めるビデオ・データストリームであって、パケットのシーケンス(34)はその中にタイミング制御パケット(36)を分散し、それによりタイミング制御パケット(36)がアクセスユニット(30)を復号化ユニット(38)に再分割し、それにより少なくともいくつかのアクセスユニット(30)が複数の復号化ユニット(38)に再分割され、各タイミング制御パケット(38)が復号化ユニット(38)のためのデコーダバッファ検索時間に信号を送り、そのペイロード・パケット(32)がパケットのシーケンス(34)におけるそれぞれのタイミング制御パケット(38)に続く、ビデオ・データストリーム。

10

【請求項 2】

サブポーション(24)はスライスであり、各ペイロード・パケット(32)は1つ以上のスライスを含む、請求項1に記載のビデオ・データストリーム。

【請求項 3】

スライスは、独立して復号化可能なスライスと、WPP処理のために、スライス境界を越えたエントロピーおよび予測復号化を用いた復号化を可能にする従属性スライスとを含む、請求項2に記載のビデオ・データストリーム。

20

【請求項 4】

パケットのシーケンス(34)の各パケットが異なるパケットタイプであるペイロード・パケット(32)およびタイミング制御パケット(36)を有する複数のパケット・タイプから正確に1つのパケット・タイプに割り当てられ、パケットのシーケンス(34)における複数のパケットタイプのパケットの発生は、各アクセスユニット(30)の範囲内でパケットにより従わされることになっているパケットタイプの間の順序を規定するいくらかの限定に依存し、それにより、アクセスユニット境界はインスタンスを検出することにより限定を用いて検出可能であり、たとえいかなる除去可能なパケットタイプがビデオ・データストリームから除去されても、限定は反対され、パケットのシーケンスの範囲内の同じ位置に残り、ペイロード・パケット(32)は除去不可能なパケットタイプであり、タイミング制御パケット(36)は除去可能なパケットタイプである、請求項1ないし請求項3のいずれかに記載のビデオ・データストリーム。

30

【請求項 5】

各パケットは、構文要素部を示しているパケット・タイプを含む、請求項1ないし請求項4のいずれかに記載のビデオ・データストリーム。

【請求項 6】

構文要素部を示しているパケット・タイプは、それぞれのパケットのパケット・ヘッダのパケット・タイプフィールド、ペイロード・パケットおよびタイミング制御パケットとの間の異なる内容、およびタイミング制御パケットのために、一方ではタイミング制御パケットと他方では異なるタイプのSEIパケットとの間を区別しているSEIパケット・タイプフィールドを含む請求項5に記載のビデオ・データストリーム。

40

【請求項 7】

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にするビデオ内容(16)をビデオ・データストリーム(22)に符号化するためのエンコーダであって、それぞれ各サブポーション(24)をビデオ・データストリーム(22)のシーケンス(34)の1つ以上のペイロード・パケット(32)に符号化し、それにより、パケットのシーケンス(34)がアクセスユニット(30)のシーケンスに分割され、各アクセスユニット(30)がビデオ内容(16)のそれぞれの画像(18)に関するペイロード・パケッ

50

ト(32)を集め、エンコーダはパケットのシーケンス(34)にタイミング制御パケット(36)を分散し、それにより、タイミング制御パケット(36)はアクセスユニット(30)を復号化ユニット(38)に再分割し、それにより、少なくともいくつかのアクセスユニット(30)は複数の復号化ユニット(38)に再分割され、各タイミング制御パケット(36)は復号化ユニット(38)のためのデコーダバッファ検索時間の信号を送り、そのペイロード・パケット(32)はパケットのシーケンス(34)におけるそれぞれのタイミング制御パケット(36)に続くように構成された、エンコーダ。

【請求項8】

エンコーダは、ビデオ内容の現在の画像を符号化する間、現在の画像(18)の現在のサブポーション(24)を現在の復号化ユニット(38)の現在のペイロード・パケット(32)に符号化し、

10

第1の瞬間に、データストリームの中で、現在のタイミング制御パケット(36)によって信号を送られるデコーダバッファ検索時間を設定することで現在のタイミング制御パケット(36)によって前に置かれる現在の復号化ユニットを伝達し、

第1の瞬間より後の第2の瞬間に現在の画像の更なるサブポーションを符号化するように構成される、請求項7に記載のエンコーダ。

【請求項9】

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にするビデオ内容(16)をビデオ・データストリーム(22)に符号化のための方法であって、それぞれ各サブポーション(24)をビデオデータストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)に符号化し、それにより、パケットのシーケンス(34)がアクセスユニット(30)のシーケンスに分割され、各アクセスユニット(30)がビデオ内容(16)のそれぞれの画像(18)に関連するペイロード・パケット(32)を集め、この方法はパケットのシーケンス(34)にタイミング制御パケット(36)を分散し、それにより、タイミング制御パケット(36)はアクセスユニット(30)を復号化ユニット(38)に再分割し、それにより、少なくともいくつかのアクセスユニット(30)は複数の復号化ユニット(38)に再分割され、各タイミング制御パケット(36)は復号化ユニット(38)のためのデコーダバッファ検索時間の信号を送り、そのペイロード・パケット(32)はパケットのシーケンス(34)におけるそれぞれのタイミング制御パケット(36)に続くように構成された、方法。

20

30

【請求項10】

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にしてその中で符号化されるビデオ内容(16)を有するビデオ・データストリーム(22)を復号化するためのデコーダであって、それぞれ各サブポーションをビデオデータストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)に符号化し、パケットのシーケンス(34)がアクセスユニット(30)のシーケンスに分割され、それにより、各アクセスユニット(30)がビデオ内容(16)のそれぞれの画像(18)に関連するペイロード・パケット(32)を集め、デコーダはビデオ・データストリームをバッファリングするためのバッファまたはビデオ・データストリームの復号化によりそこから得られるビデオ内容の再現を含み、パケットのシーケンスに分散されたタイミング制御パケット(36)を探し、アクセスユニット(30)をタイミング制御パケット(36)で復号化ユニット(38)に再分割し、それにより、少なくともいくつかのアクセスユニットが複数の復号化ユニットに再分割され、復号化ユニットを単位にするバッファを空にするように構成された、デコーダ。

40

【請求項11】

デコーダは、タイミング制御パケット(36)を探す際に、各パケットにおいて、構文要素部を示すパケットタイプを検査し、構文要素部を示すパケットタイプの値が所定の値に等しい場合それぞれのパケットをタイミング制御パケット(36)として認めるように構成された、請求項10に記載のデコーダ。

【請求項12】

50

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にしてその中で符号化されるビデオ内容(16)を有するビデオ・データストリーム(22)を復号化するための方法であって、それぞれ各サブポーションをビデオ・データストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)に符号化し、パケットのシーケンス(34)がアクセスユニット(30)のシーケンスに分割され、それにより、各アクセスユニット(30)がビデオ内容(16)のそれぞれの画像(18)に関連するペイロード・パケット(32)を集め、この方法は、ビデオ・データストリームをバッファリングするためのバッファまたはビデオ・データストリームの復号化によりそこから得られるビデオ内容の再現を使用し、パケットのシーケンスに分散されたタイミング制御パケット(36)を探し、アクセスユニット(30)をタイミング制御パケット(36)で復号化ユニット(38)に再分割し、それにより、少なくともいくつかのアクセスユニットが複数の復号化ユニットに再分割され、復号化装置を単位にするバッファを空にする、方法。

10

【請求項13】

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にしてその中で符号化されるビデオ内容(16)を有するビデオ・データストリーム(22)を送信するためのネットワークエンティティであって、各サブポーションはビデオ・データストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)にそれぞれ符号化され、パケットのシーケンス(34)はアクセスユニット(30)のシーケンスに分割され、それにより、各アクセス装置(30)がビデオ内容(16)のそれぞれの画像(18)に関連するペイロード・パケット(32)を集め、デコーダはパケットのシーケンスに分散するタイミング制御パケット(36)を探すように構成され、アクセスユニットをタイミング制御パケット(36)で復号化ユニットに再分割し、それにより、少なくともいくつかのアクセスユニット(30)は複数の復号化ユニット(38)に再分割され、各タイミング制御パケット(36)から復号化ユニット(38)のためのデコーダバッファ検索時間を引き出し、そのペイロード・パケット(32)はパケットのシーケンス(34)のそれぞれのタイミング制御パケット(36)に続き、復号化ユニット(38)のためのデコーダバッファ検索時間に従属しているビデオ・データストリームの伝送を実行する、ネットワークエンティティ。

20

【請求項14】

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にしてその中で符号化されるビデオ内容(16)を有するビデオ・データストリーム(22)を伝送するための方法であって、各サブポーションはビデオ・データストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)にそれぞれ符号化され、パケットのシーケンス(34)はアクセスユニット(30)のシーケンスに分割され、それにより、各アクセスユニット(30)がビデオ内容(16)のそれぞれの画像(18)に関連するペイロード・パケット(32)を集め、この方法は、パケットのシーケンスに分散するタイミング制御パケット(36)を探すように構成され、アクセスユニットをタイミング制御パケット(36)で復号化ユニットに再分割し、それにより、少なくともいくつかのアクセスユニット(30)は複数の復号化ユニット(38)に再分割され、各タイミング制御パケット(36)から復号化ユニット(38)のためのデコーダバッファ検索時間を引き出し、そのペイロード・パケット(32)はパケットのシーケンス(34)のそれぞれのタイミング制御パケット(36)に続き、復号化ユニット(38)のためのデコーダバッファ検索時間に従属しているビデオ・データストリームの伝送を実行する、方法。

30

40

【請求項15】

スライス(24)の中の符号化順序を用いて、ビデオ内容(16)の画像(18)が空間的に再分割されるスライス(24)を単位にして、予測およびエントロピー符号化を用いて、そこに符号化されるビデオ内容(16)を有するビデオ・データストリームであって、ビデオ内容の画像が空間的に再分割されるタイルの内部に予測符号化の予測および/

50

またはエントロピー符号化の予測を制限し、そこにおいて、スライス(24)のシーケンスは符号化順序でビデオ・データストリームのパケットのシーケンスのペイロード・パケット(32)にパケット化され、パケットのシーケンス(34)はアクセスユニット(30)のシーケンスに分割され、それにより、各アクセスユニットがビデオ内容(16)のそれぞれの画像(18)に関連するスライス(24)をそこにパケット化し、そこにおいて、パケットのシーケンス(34)が1つのアクセスユニットのペイロード・パケットとの間でそこに分散するタイル識別パケット(72)を有し、パケットのシーケンス(34)のそれぞれのタイル識別パケット(72)に直ちに続く1つ以上のペイロード・パケット(32)にパケット化されたいくつかのスライス(24)により1つ以上のタイル(70)を確認する、ビデオ・データストリーム。

10

【請求項16】

タイル識別パケット(72)は、正確に直ちに続くペイロード・パケットにパケット化されたいくつかのスライス(24)によって覆われる1つ以上のタイル(70)を確認する、請求項15に記載のビデオ・データストリーム。

【請求項17】

タイル識別パケット(72)は、現在のアクセスユニット(30)の端部の前までパケットのシーケンス(34)のそれぞれのタイル識別パケット(72)に直ちに続く1つ以上のペイロード・パケット(32)にパケット化されたいくつかのスライス(24)によって覆われる1つ以上のタイル(70)を確認し、次のタイル識別パケット(72)は、それぞれ、パケットのシーケンス(34)にある、請求項15に記載のビデオ・データストリーム。

20

【請求項18】

請求項15ないし請求項16のいずれかに記載のビデオ・データストリームを受信し、タイル識別パケット(72)、パケットのシーケンスのそれぞれのタイル識別パケット(72)に直ちに続く1つ以上のペイロード・パケット(72)にパケット化されたスライス(24)によって覆われるタイル(70)に基づいて確認するように構成される、ネットワークエンティティ。

【請求項19】

ネットワークエンティティは、ビデオ・データストリームに関連する伝送タスクを決定するために、識別の結果を使用するように構成される、請求項18に記載のネットワークエンティティ。

30

【請求項20】

伝送タスクが不良パケットに関連する再伝送を含む、請求項18または請求項19に記載のネットワークエンティティ。

【請求項21】

ネットワークエンティティは、タイル識別パケット(72)およびそれぞれのタイル識別パケットによって確認されたタイル(70)を覆うパケット化されたスライス(24)を有するそれぞれのタイル識別パケット(72)に直ちに続くペイロード・パケットに高い優先度を割り当てることにより異なる優先度を有する異なるタイル(70)を取り扱い、そのため優先度は高く、そして、それぞれのタイル識別パケットによって確認されたタイル(70)を覆うパケット化されたスライス(24)を有するそれぞれのタイル識別パケット(72)に直ちに続くペイロード・パケットに対して、そのために優先度が低い、請求項18または請求項19に記載のネットワークエンティティ。

40

【請求項22】

ネットワークエンティティは、割り当てられた低い優先度を有するペイロード・パケットのいかなる再伝送の要求の前に、まず割り当てられた高い優先度を有するペイロード・パケットの再伝送を要求するように構成された、請求項21に記載のネットワークエンティティ。

【請求項23】

請求項15ないし請求項16のいずれかに記載のビデオ・データストリームを受信し、

50

タイル識別パケット(72)、パケットのシーケンスのそれぞれのタイル識別パケット(72)に直ちに続く1つ以上のペイロード・パケット(72)にパケット化されたスライス(24)によって覆われるタイル(70)に基づいて確認するように構成される、方法。

【請求項24】

ビデオ内容(16)の画像(18)のサブポーション(24)を単位にしてその中で符号化されるビデオ内容(16)を有するビデオ・データストリームであって、各サブポーション(24)は、それぞれビデオ・データストリーム(22)のパケットのシーケンス(34)の1つ以上のペイロード・パケット(32)に符号化され、パケットのシーケンス(34)はアクセスユニット(30)に分割され、それにより、各アクセスユニット(30)はビデオ内容(16)のそれぞれの画像(18)に関連するペイロード・パケット(32)を集め、少なくともいくつかのアクセスユニットはパケットのシーケンス(34)がその中にROIパケット(64)を分散したことを有し、それにより、タイミング制御パケット(64)は復号化ユニット(38)にアクセスユニット(64)を再分割し、それにより、少なくともいくつかのアクセスユニット(30)はそれぞれのアクセスユニットの画像に関連するペイロード・パケットとの間で分散されたROIパケット(64)を有し、各ROIパケットはパケットのシーケンス(34)の1つ以上の次のペイロード・パケットに関連し、それぞれのROIパケットに続き、それぞれのROIパケットが関連する1つ以上のペイロード・パケットに符号化されるサブポーション(24)がビデオ内容の関心領域を覆うかどうかについて確認する、ビデオ・データストリーム。

10

20

【請求項25】

サブポーションはスライスであり、ビデオ内容はビデオ内容の画像が分割されたタイルの内部への予測および/またはエントロピー符号化に制限された予測およびエントロピー符号化を用いてビデオ・データストリームに符号化され、各ROIパケットが更にそれぞれのROIパケットが関連する1つ以上のペイロード・パケットのいくつかに符号化されたサブポーション(24)が関心領域を覆うタイルを確認する、請求項24に記載のビデオ・データストリーム。

【請求項26】

各ROIパケットは直ちに続くペイロード・パケットだけに関連する、請求項24または請求項25に記載のビデオ・データストリーム。

30

【請求項27】

各ROIパケットは、それぞれのROIパケット、および次のROIパケットが配置されたアクセスユニットの端部の始まりまでパケットのシーケンスのそれぞれのROIパケットが直ちに続く全てのROIパケットに関連する、請求項24または請求項25に記載のビデオ・データストリーム。

【請求項28】

請求項24ないし請求項27のいずれかに記載のビデオ・データストリームを受信し、ROIパケットに基づいてビデオ内容のROIパケットを確認するように構成された、ネットワークエンティティ。

40

【請求項29】

ネットワークエンティティは、ビデオ・データストリームに関連する伝送タスクを決定するために確認の結果を使用するように構成された、請求項27に記載のネットワークエンティティ。

【請求項30】

伝送タスクが不良パケットに関する再伝送要求を含む、請求項28または請求項29に記載のネットワークエンティティ。

【請求項31】

ネットワークエンティティは、ROIパケットおよびそれぞれのROIパケットが関連し、ROIパケット信号が重ならないそれぞれのROIパケット(72)に続く1つ以上

50

のペイロード・パケットよりも、それぞれのROIパケットが関連し、ROIパケットはそれぞれのROIパケットが関連する1つ以上のペイロード・パケットのいくつかに符号化されたサブポーション(24)により関心領域の重なり信号を送る、ROIパケット(72)およびそれぞれのROIパケット(72)に続く1つ以上のペイロード・パケットに高い優先度を割り当てることにより増加された優先度を有する関心領域(70)を取り扱うように構成される、ネットワークエンティティ。

【請求項32】

ネットワークエンティティは、そこに割り当てられた低い優先度を有するペイロード・パケットのいかなる再伝送の要求より前に、まずそこに割り当てられた高い優先度を有するペイロード・パケットの再伝送を要求するように構成される、請求項31に記載のネットワークエンティティ。

10

【請求項33】

請求項23ないし請求項26のいずれかに記載のビデオ・データストリームを受信し、ROIパケットに基づいて、ビデオ内容のROIを確認する、方法。

【請求項34】

コンピュータ上で動作するとききに、請求項9、12、14、23または33に記載の方法を実行するためのプログラムコードを有する、コンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、特に、低い遅延アプリケーションと関連して有利であるビデオ・データストリーム・コンセプトに関する。

20

【背景技術】

【0002】

HEVC[2]は、アプリケーション層へのHigh Level Syntaxシグナリングの異なる手段を可能にする。この種的手段は、NALユニットヘッダ、Parameter SetsおよびSupplemental Enhancement Information(SEI)Messagesである。後者は、復号化プロセスで使用されない。High Level Syntaxシグナリングの他の手段は、例えばMPEG2 Transport Protocol[3]またはRealtime Transport Protocol[4]等のそれぞれのトランスポート・プロトコル、およびそのペイロードに特有の仕様、例えばH.264/AVC[5]のための提案、拡張性のあるビデオ符号化(SVC)[6]またはHEVC[7]等が起源である。このようなトランスポート・プロトコルは、例えばHEVC[2]のような、それぞれのアプリケーションレイヤーコーデック仕様のHigh Levelシグナリングに類似の構造および機構を使用するHigh Levelシグナリングを導入することができる。このようなシグナリングの1つの例は、補助情報をトランスポートレイヤーに提供する[6]にて説明されているようなPayload Content Scalability Information(PACSI)NALユニットである。

30

【0003】

パラメータ・セットのために、HEVCは1つのおよび中心の位置でアプリケーションレイヤーにより使用される最も重要なストリーム情報に従うVideo Parameter Set(VPS)を含む。初期の方法において、この情報は、複数のParameter SetsおよびNALユニットヘッダから集められることが必要とされていた。

40

【0004】

本出願に先立って、Hypothetical Reference Decoder(HRD)のCoded Picture Buffer(CPB)動作に関する標準の状態、およびPicture Parameter Set(PPS)と同様にDependent Slicesのサブピクチャおよび構文を表す復号化ユニットの定義とともにSequence Parameter Set(PPS)/Video Usab

50

ility Information (VUI)、Picture Timing SEI、Buffering Period SEIに存在するすべての関連した構文は、以下の通りであった。

【0005】

サブピクチャ・レベルにおける低遅延CPB動作を可能にするために、サブピクチャCPB動作が提案され、HEVCドラフト標準7 JCTVC I1003 [2]に組み込まれた。ここでは特に、復号化ユニットは、以下のように〔2〕のセクション3において定められた：

復号化装置：アクセスユニットまたはアクセスユニットのサブセット。SubPicCpbFlagが0に等しいものである場合、復号化ユニットはアクセスユニットである。さもなければ、復号化ユニットは、アクセスユニットおよび関連する非VCLNALユニットにおける1つ以上のVCLNALユニットから成る。アクセスユニットにおける第1のVCLNALユニットのために、関連する非VCLNALユニットはあり、そして、フィルアNALユニットは、もしあれば、第1のVCLNALユニットにすぐに続き、第1のVCLNALユニットに先行するアクセスユニットにおけるすべての非VCLNALユニットに続く。アクセスユニットにおける第1のVCLNALユニットでないVCLNALユニットのために、関連する非VCLNALユニットは、もしあれば、VCLNALユニットに直ちに続くフィルアデータNALユニットである。

【0006】

その時間まで定められる基準において、「復号化ユニットの除去および復号化ユニットの復号化のタイミング」が記載されていて、Annex C「Hypothetical reference decoder」に加えられた。sub-picture timingを送るために、VUIにおけるHRDパラメータとともに、buffering period SEIおよびpicture timing SEIメッセージは、サブピクチャユニットとして、復号化ユニットをサポートするまで拡張されている。

【0007】

〔2〕のBuffering period SEIメッセージ構文は、図1に示される。

【0008】

NalHrdBpPresentFlagまたはVclHrdBpPresentFlagが1に等しいときに、buffering period SEIメッセージはビットストリームのいかなるアクセスユニットとも関連していることがありえる、そして、buffering period SEIメッセージは、各RAPアクセス装置と、および、recovery point SEIメッセージと関連する各アクセスユニットと関連する。

【0009】

いくつかの適用において、buffering period SEIメッセージのしばしば起こる存在は好ましい。

【0010】

buffering periodは、復号化順序におけるbuffering SEIメッセージの2つのインスタンス間のアクセスユニットのセットとして規定された。

【0011】

意味論は、以下の通りだった：

【0012】

seq_parameter_set_idは、シーケンスHRD属性を含むシーケンス・パラメータ・セットを規定する。seq_parameter_set_idの値は、バッファリング期間SEIメッセージと関連した主要な符号化画像によって参照される画像パラメータ・セットにおけるseq_parameter_set_idの値に等しい。seq_parameter_set_idの値は範囲0から31の範囲内に含まれる。

10

20

30

40

50

【0013】

1に等しい`rap_cpb_params_present_flag`は、`initial_alt_cpb_removal_delay[SchedSelIdx]`および`initial_alt_cpb_removal_delay_offset[SchedSelIdx]`構文要素の存在を規定する。存在しない場合、`rap_cpb_params_present_flag`の値は、0に等しいものと推測される。関連する画像がCRA画像でもBLA画像でもない場合、`rap_cpb_params_present_flag`の値は0に等しい。

【0014】

`initial_cpb_removal_delay[SchedSelIdx]`および`initial_alt_cpb_removal_delay[SchedSelIdx]`は、`SchedSelIdx-th-CPB`のための最初のCPB除去遅延を規定する。構文要素は、`initial_cpb_removal_delay_length_minus1+1`によって与えられるビットの長さを有し、90kHzクロックの単位にある。構文要素の値は、0に等しいものではなくて、90kHzクロックタイムにおけるCPBサイズの時間等価である $90000 * (CpbSize[SchedSelIdx] \div BitRate[SchedSelIdx])$ を超えるものではない。

10

【0015】

`initial_cpb_removal_delay_offset[SchedSelIdx]`および`initial_alt_cpb_removal_delay_offset[SchedSelIdx]`は、`SchedSelIdx-th-CPB`がCPBに符号化データユニットの最初の搬送時間を規定するために使用される。構文要素は、`initial_cpb_removal_delay_length_minus1+1`によって与えられるビットの長さを有し、90kHzクロックの単位にある。これらの構文要素はデコーダによって使用されず、搬送スケジューラ(HSS)のためにだけ必要とされる。

20

【0016】

全ての符号化ビデオ・シーケンスの上に、`initial_cpb_removal_delay[SchedSelIdx]`および`initial_cpb_removal_delay_offset[SchedSelIdx]`の合計は`SchedSelIdx`の値ごとに一定であり、`initial_alt_cpb_removal_delay[SchedSelIdx]`および`initial_alt_cpb_removal_delay_offset[SchedSelIdx]`の合計は`SchedSelIdx`の値ごとに一定である。

30

【0017】

〔2〕の画像タイミングSEIメッセージ構文は、図2に示される。

【0018】

画像タイミングSEIメッセージの構文は、画像タイミングSEIメッセージに関連する符号化画像のために機能しているシーケンス・パラメータ・セットの内容に依存していた。しかしながら、IDRまたはBLAアクセスユニットの画像タイミングSEIメッセージに先立ってアクセスユニットの範囲内でバッファリング期間SEIメッセージがない限り、関連するパラメータ・セット(そして、ビットストリームにおける第1の画像でないIDRまたはBLA画像のために、符号化画像がIDR画像またはBLA画像であることの決定)は、符号化画像の第1の符号化スライスNALユニットを復号化するまで発生しない。符号化画像の符号化スライスNALユニットがNALユニット順序における画像タイミングSEIメッセージに続くので、デコーダが符号化画像のために機能するシーケンス・パラメータのパラメータを決定するまで画像タイミングSEIメッセージを含んでいるRBSPPを格納することが必要であるケースがあり、そうすると、画像タイミングSEIメッセージの構文解析を実行する。

40

【0019】

50

ビットストリームにおける画像タイミングSEIメッセージの存在は、以下のように規定された。

- `CpbDpbDelaysPresentFlag` が 1 に等しい場合、1つの画像タイミングSEIメッセージは、符号化ビデオ・シーケンスのあらゆるアクセスユニットに存在する。

- さもなければ (`CpbDpbDelaysPresentFlag` が 0 に等しい)、画像タイミングSEIメッセージは、符号化ビデオ・シーケンスのいかなるアクセスユニットにも存在しない。

【0020】

意味論は、以下の通りに定められた：

10

【0021】

`cpb_removal_delay` は、バッファから画像タイミングSEIメッセージに関連するアクセスユニットデータを除去する前の先行するアクセスユニットにおけるごく最近のバッファリング期間SEIメッセージに関連するアクセスユニットのCPBからの除去の後、どれくらいのクロック・ティックを待つかについて規定する。この値は、HSSのためのCPBにアクセスユニットデータの出現の最も早く可能な時間を計算するためにも用いられる。構文要素は、ビットの長さが `cpb_removal_delay_length_minus_1 + 1` によって与えられる固定長コードである。`cpb_removal_delay` は、法としての剰余 $2^{(cpb_removal_delay_length_minus1 + 1)}$ カウンタである。

20

【0022】

構文要素 `cpb_removal_delay` の (ビットにおける) 長さを決定する `cpb_removal_delay_length_minus_1` の値は、画像タイミングSEIメッセージに関連する主要な符号化画像のために機能するシーケンス・パラメータ・セットにおいて符号化される `cpb_removal_delay_length_minus_1` の値であるが、但し、`cpb_removal_delay` はバッファリング期間SEIメッセージを含んでいる先行するアクセスユニットの除去時間に関連するクロック・ティックの数を規定し、それは異なる符号化ビデオ・シーケンスのアクセスユニットであってもよい。

【0023】

`dpb_output_delay` は、画像のDPB出力時間を計算するために使用される。それは、CPBからアクセス装置の最後の復号化装置の除去の後、復号化画像がDPBから出力される前に、どれくらいのクロック・ティックを待たせるかについて規定する。

30

【0024】

画像は、それがいまだに「短期参照のために使用する」または「長期参照のために使用する」ものとしてマークされているその出力時間に、DPBから取り除かれない。

【0025】

1つの `dpb_output_delay` だけは、復号化画像のために規定される。

【0026】

構文要素 `dpb_output_delay` の長さは、`dpb_output_delay_length_minus_1 + 1` によってビットにおいて与えられる。`sps_max_dec_pic_buffering_max_temporal_layers_minus_1` が 0 に等しいときに、`dpb_output_delay` は 0 に等しくなるであろう。

40

【0027】

出力タイミング同調デコーダから出力されるいくつかの画像の `dpb_output_delay` から得られる出力時間は、復号化順序においていくつかの次の符号化ビデオ・シーケンスにおけるすべての画像の `dpb_output_delay` から得られる出力時間に先行する。

50

【0028】

この構文要素の値によって規定される画像アウトプット順序は、PicOrderCntValの値によって規定されるのと同じ順序である。

【0029】

復号化順序において、no_output_of_prior_pics_flagが1に等しいか、または1に等しいと推測されるIDRまたはBLAに先行するため、「パング」プロセスによって出力されない画像のために、dpb_output_delayから得られる出力時間は、同じ符号化ビデオ・シーケンスの中ですべての画像と関連するPicOrderCntValの値の増加と共に増加している。

【0030】

num_decoding_units_minus1 plus 1は、画像タイミングSEIメッセージが関連しているアクセスユニットにおける復号化ユニットの数を規定する。num_decoding_units_minus1の値は、0からPicWidthInCtbs * PicHeightInCtbs - 1の範囲内において含まれている。

10

【0031】

num_nalus_in_du_minus1[i] plus 1は、画像タイミングSEIメッセージが関連しているアクセスユニットのi番目の復号化ユニットにおけるNALユニットの数を規定する。num_nalus_in_du_minus1[i]の値は、0からPicWidthInCtbs * PicHeightInCtbs - 1の範囲内に含まれている。

20

【0032】

アクセスユニットの第1の復号化ユニットは、アクセスユニットにおける復号化順序において第1のnum_nalus_in_du_minus1[0]の連続的なNALユニットから成る。アクセスユニットのi番目(iは0より大)の復号化ユニットは、復号化順序において、アクセスユニットの前の復号化ユニットにおける最後のNALユニットに直ちに続くnum_nalus_in_du_minus1[i] + 1の連続的なNALユニットから成る。各復号化ユニットには、少なくとも1つのVCL NAL装置がある。VCL NALユニットに関連するすべての非VCL NALユニットが、同じ復号化ユニットに含まれる。

30

【0033】

du_cpb_removal_delay[i]は、画像タイミングSEIメッセージに関連するアクセスユニットにおけるi番目の復号化ユニットをCPBから除去する前であって前のアクセスユニットにおいてごく最近のバッファリング期間SEIメッセージに関連するアクセスユニットにおける第1の復号化ユニットのCPBからの除去の後、どれくらいのサブピクチャ・クロック・ティックを待つかについて規定する。この値は、HSSのためのCPBに復号化ユニットデータの到着の最も早い可能な時間を計算するためにも用いられる。構文要素は、ビットの長さがcpb_removal_delay_length_minus1 + 1によって与えられる固定長コードである。du_cpb_removal_delay[i]は、2(cpb_removal_delay_length_minus1 + 1)カウンタを法とする剰余である。

40

【0034】

構文要素du_cpb_removal_delay[i]の(ビットにおける)長さを決定するcpb_removal_delay_length_minus1の値は、画像タイミングSEIメッセージに関連する符号化画像のために機能するシーケンス・パラメータ・セットにおいて符号化されるcpb_removal_delay_length_minus1の値であるが、du_cpb_removal_delay[i]はバッファリング期間SEIメッセージを含む先行するアクセスユニットにおける第1の復号化ユニットの除去時間と関連してサブピクチャ・クロック・ティックの数を特定し、それは異なる符号化ビデオ・シーケンスのアクセスユニットでもよい。

50

【 0 0 3 5 】

いくつかの情報、〔 2 〕の VUI 構文に含まれた。〔 2 〕の VUI パラメータ構文は、図 3 に示される。〔 2 〕の HRD パラメータ構文は、図 4 に示される。意味論は、以下の通りに定められた：

【 0 0 3 6 】

1 に等しい `sub_pic_cpb_params_present_flag` は、サブピクチャ・レベルの CPB 除去遅延パラメータがあり、CPB がアクセスユニットレベルまたはサブピクチャ・レベルで作動することができることを示している。0 に等しい `sub_pic_cpb_params_present_flag` は、サブピクチャ・レベルの CPB 除去遅延パラメータがなく、CPB はアクセスユニットレベルで作動することを示している。`sub_pic_cpb_params_present_flag` が
10

【 0 0 3 7 】

`num_units_in_sub_tick` は、サブピクチャ・クロック・ティックカウンタの（サブピクチャ・クロック・ティックと呼ばれる）1 つの増加に対応する周波数 `time_scale` Hz で作動するクロックのタイムユニットの数である。`num_units_in_sub_tick` は、0 より大きい。サブピクチャ・クロック・ティックは、`sub_pic_cpb_params_present_flag` が 1 に等しいときに、符号化データにおいて表されることができる時間の最小限の間隔である。

【 0 0 3 8 】

1 に等しい `tiles_fixed_structure_flag` は、示される場合、符号化ビデオ・シーケンスにおいて機能している各画像パラメータ・セットが、構文要素 `num_tile_columns_minus1`、`num_tile_rows_minus1`、`uniform_spacing_flag`、`column_width[i]`、`row_height[i]` および `loop_filter_across_tiles_enabled_flag` と同じ値を有することを示す。0 に等しい `tiles_fixed_structure_flag` は、異なる画像パラメータ・セットのタイル構文要素が同じ値を有することができるかまたは有することができないかを示す。`tiles_fixed_structure_flag` 構文要素がないときに、それは 0 に等しいものであると推測される。
20

【 0 0 3 9 】

1 に等しい `tiles_fixed_structure_flag` の信号伝達は、符号化ビデオ・シーケンスにおける各画像がマルチスレッドの復号化の場合作業負荷配分に役立つかもしれない同一方法で分配されるタイルと同じ数を有するというデコーダへの保証である。

【 0 0 4 0 】

〔 2 〕の充填材データは、図 5 に示されるフィルタデータ RBS P 構文を使用して信号伝達された。

【 0 0 4 1 】

ビットストリームおよびデコーダの適合をチェックするのに用いられる〔 2 〕の仮想参照デコーダは以下の通りに定められた：

【 0 0 4 2 】

2 種類のビットストリームは、このリコメンデーション | 国際基準についてチェックしている HRD の適合に従属する。タイプ I ビットストリームと呼ばれている第 1 のこの種のタイプのビットストリームは、ビットストリームのすべてのアクセスユニットのための VCL NAL ユニットおよびフィルタデータ NAL ユニットだけを含んでいる NAL ユニットストリームである。タイプ II ビットストリームと呼ばれている第 2 のタイプのビットストリームは、ビットストリームのすべてのアクセスユニットのための VCL NAL ユニットおよびフィルタデータ NAL ユニットに加えて、以下のうちの少なくとも 1 つを含む：
40

10

20

30

40

50

- フィラーデータNALユニット以外のさらなる非VCL NALユニット、
- NALユニットストリームからバイトストリームを形成するすべての `leading__zero__8bits`、`zero__byte`、`start__code__prefix__one__3bytes` および `trailing__zero__8bits` 構文要素。

【0043】

図6は、〔2〕のHRDによってチェックされるビットストリーム適合ポイントのタイプを示す。

【0044】

2種類のHRDパラメータ・セット(NAL HRDパラメータおよびVCL HRDパラメータ)が用いられる。HRDパラメータ・セットはビデオ有用性情報によって信号が伝達され、それはシーケンス・パラメータ構文構造の一部である。

10

【0045】

VCL NALユニットに付託されるすべてのシーケンス・パラメータ・セットおよび画像パラメータ・セット、および対応するバッファリング期間および画像タイミングSEIメッセージは、HRDに、タイムリーな方法で、ビットストリームにおいて、または、他の手段によって伝達される。

【0046】

それらのNALユニット(またはちょうどそれらのいくつか)がこのリコメンデーション|国際基準によって規定されない他の手段によってデコーダ(またはHRDに)に伝達されるときに、非VCL NALユニットの「存在」のための仕様も満たされる。ビットを計数するために、ビットストリームに実際に存在する適当なビットだけは計数される。

20

【0047】

一例として、ビットストリームにおける存在以外の方法で伝達され、ビットストリームの中に存在するNALユニットを有する、非VCL NALユニットの同期は、ビットストリームの中の2つのポイントを示すことによって達成され、それらの間においてビットストリーム中に非VCL NALユニットが存在し、ビットストリーム中でそれを伝達することを決定されるエンコーダを有していた。

【0048】

非VCL NALユニットの内容がビットストリームの範囲内における存在以外のいくつかの手段によってアプリケーションのために伝達されるときに、非VCL NALユニットの内容の表現はこの添付において特定される同じ構文を使用することを必要としない。

30

【0049】

HRD情報がビットストリーム内に含まれるときに、単にビットストリームに含まれる情報に基づいてのみビットストリームとこの従属の要求との一致を確認することができる点に注意すべきである。HRD情報がビットストリームに存在しないときに、すべての「独立型」Type Iビットストリームの場合のように、HRDデータがこのRecommendation|International Standardにおいて特定されないいくつかの他の手段によって提供されるときに、一致は確認されることができるだけである。

40

【0050】

HRDは、図7に示すように、符号化画像バッファ(CPB)、瞬間的な復号化プロセス、復号化画像バッファ(DPB)および出力クロッピングを含む。

【0051】

CPBサイズ(ビットの数)は、`CpbSize[SchedSelIdx]` である。時間的レイヤーXのためのDPBサイズ(画像格納バッファの数)は、0から`sps__max__temporal__layers__minus__1`を含む範囲のXごとに`sps__max__dec__pic__buffering[X]` である。

【0052】

可変`SubPicCpbPreferredFlag`は外部手段によって規定されてい

50

るか、または、外部手段によって規定されないときは、0に規定される。

【0053】

可変SubPicCpbFlagは、以下の通りに引き出される：

SubPicCpbFlag=SubPicCpbPreferredFlag&&sub__pic__cpb__params__present__flag

【0054】

SubPicCpbFlagが0に等しい場合、CPBはアクセスユニットレベルで作動し、各復号化ユニットはアクセスユニットである。さもなければ、CPBはサブピクチャ・レベルで作動し、各復号化ユニットはアクセスユニットのサブセットである。

【0055】

HRDは、次のように作動する。指定された到着スケジュールに従ってCPBに流れ込む復号化ユニットに関連するデータは、HSSによって分配される。各復号化ユニットに関連したデータは削除されて、CPB除去時間に瞬間的な復号化プロセスによって即座に復号化される。各復号化画像は、DPBに置かれる。復号化画像は、後のDPB出力時またはインター・プレディクション・リファレンスのためにもはや必要でなくなった時に、DPBから取り除かれる。

10

【0056】

HRDは、バッファ期間SEIによって特定されるように初期化される。CPBからの復号化ユニットの除去タイミングおよびDPBからの復号化画像の出力タイミングは、画像タイミングSEIメッセージにおいて規定される。特定の復号化ユニットに関するすべてのタイミング情報は、復号化ユニットのCPB除去時間の前に到着する。

20

【0057】

HRDは、ビットストリームおよびデコーダの一致をチェックするために用いられる。

【0058】

一致が、ビットストリームを生成するために用いられるすべてのフレームレートおよびクロックが正確にビットストリームにおいて信号を送られる値に合致するという仮定の下で保証される一方、本当のシステムにおいて、これらの各々は信号を送られたか指定された値から変化することができる。

【0059】

すべての計算は本当の値によって行われ、その結果、丸め誤差は伝搬することができない。例えば、ちょうど復号化ユニットの除去より前の、または、後のCPBのビットの数が、必ずしも整数であるというわけではない。

30

【0060】

可変tcは、以下の通りに引き出されて、クロック・ティックと呼ばれている：

tc=num__units__in__tick time__scale

可変tc_subは、以下の通りに引き出されて、サブピクチャ・クロック・ティックと呼ばれている：

tc_sub=num__units__in__sub__tick time__scale

【0061】

以下は、制約を表すために規定される：

- アクセスユニット0である第1のアクセスユニットを有する復号化順序においてアクセスユニットnをn番目のアクセスユニットとする。
- 画像nをアクセスユニットnの符号化画像または復号化画像であるとする。
- 復号化ユニット0である第1の復号化ユニットを有する復号化順序において復号化ユニットmをm番目の復号化ユニットとする。

40

【0062】

{2}において、スライス・ヘッダ構文は、いわゆる従属スライスを可能にする。

【0063】

図8は、{2}のスライス・ヘッダ構文を示す。

【0064】

スライス・ヘッダ意味論は、以下の通りに定められた：

50

【0065】

1に等しい`dependent__slice__flag`は、示されていない各スライス・ヘッダ構文要素の値が、符号化ツリー・ブロック・アドレスが`SliceCtbAddrRS`である符号化ツリー・ブロックを含んでいる前のスライスにおける対応するスライス・ヘッダの値に等しいと推測されることを示している。示されていないとき、`dependent__slice__flag`の値は0に等しいと推測される。`dependent__slice__flag`の値は、`SliceCtbAddrRS`が0に等しいときに、0に等しくなる。

【0066】

`slice__address`は、スライスが始まるスライス粒状解像度のアドレスを規定する。`slice__address`構文要素の長さは、 $(CEil(\log_2(\text{PicWidthInCtbs} * \text{PicHeightInCtbs})) + \text{SliceGranularity})$ ビットである。

10

【0067】

可変`SliceCtbAddrRS`は、符号化ツリー・ブロック・ラスタ・スキャン順序においてスライスがスタートする符号化木ブロックを規定して、以下の通りに引き出される。

$$\text{SliceCtbAddrRS} = (\text{slice_address} \gg \text{SliceGranularity})$$

【0068】

可変`SliceCbAddrZS`は、z - スキャン順序の最小限の符号化ブロック粒度のスライスの第1の符号化ブロックのアドレスを規定して、以下の通りに引き出される。

20

$$\text{SliceCbAddrZS} = \text{slice_address}$$

$$\ll ((\log_2_diff_max_min_coding_block_size = \text{SliceGranularity}) \ll 1)$$

【0069】

スライス復号化は、スライス開始座標の可能な最大の符号化ユニットで始まる。

【0070】

`first__slice__in__pic__flag`は、スライスが画像の第1のスライスであるかどうかを示す。`first__slice__in__pic__flag`が1に等しい場合、変数`SliceCbAddrZS`および`SliceCtbAddrRS`は両方とも0に設定され、画像の第1の符号化ツリー・ブロックから復号化が始まる。

30

【0071】

`pic__parameter__set__id`は、組み込まれた画像パラメータを規定する。`pic__parameter__set__id`の値は、0から255の範囲を含んでいる。

【0072】

`num__entry__point__offsets`は、スライス・ヘッダの`entry__point__offset[i]`構文要素の数を規定する。`tiles__or__entropy__coding__sync__idc`が1に等しいとき、`num__entry__point__offsets`の値は、0から $(\text{num_tile_columns_minus}1 + 1) * (\text{num_tile_rows_minus}1 + 1)$ の範囲を含んでいる。`tiles__or__entropy__coding__sync__idc`が2に等しいとき、`num__entry__point__offsets`の値は、0から`PicHeightInCtbs = 1`の範囲を含んでいる。示されていないとき、`num__entry__point__offsets`の値は、0に等しいと推測される。

40

【0073】

`offset__len__minus1_plus1`は、ビットにおいて、`entry__point__offset[i]`構文要素の長さを規定する。

【0074】

`entry__point__offset[i]`は、i番目のエントリポイントのオフセットをバイトで規定し、`offset__len__minus1_plus1`ビットによ

50

て表される。スライス・ヘッダの後の符号化スライス・データは、0から`num__entry__point__offsets`の範囲を含む、`num__entry__point__offsets + 1`サブセットから成る。サブセット0は、符号化スライス・データ`k`の0から`entry__point__offset[0] - 1`を含むバイトからなり、`k`は1から`num__entry__point__offset - 1`の範囲を含み、符号化スライス・データの`entry__point__offset[k - 1]`から`entry__point__offset[k] + entry__point__offset[k - 1] - 1`を含むバイトからなり、(サブセット・インデックスが`num__entry__point__offsets`に等しい)最後のサブセットは、符号化スライス・データの残りのバイトから成る。

10

【0075】

`tiles__or__entropy__coding__sync__idc`が1に等しく、`num__entry__point__offsets`が0より大きいとき、各サブセットは正確に1つのタイルのすべての符号化ビットを含み、サブセット(すなわち、`num__entry__point__offsets + 1`の値)の数はスライスのタイルの数以下である。

`tiles__or__entropy__coding__sync__idc`が1に等しいとき、各スライスは(エントリポイントのケース・シグナリングは、不必要である)1つのタイルのサブセットが完全なタイルの整数値を含まなければならない。

20

【0076】

`tiles__or__entropy__coding__sync__idc`が2に等しく、`num__entry__point__offsets`が0より大きいとき、0から`num__entry__point__offsets - 1`の範囲を含む`k`を有する各サブセット`k`は、確実に符号化ツリー・ブロックの1つの行の全ての符号化ビットを含み、(`num__entry__point__offsets`に等しいサブセット・インデックスを有する)最後のサブセットは、スライスに含まれる残りの符号化ブロックの全ての符号化ビットを含み、残りの符号化ブロックは符号化ツリー・ブロックの正確に1つの行か符号化ツリー・ブロックの1つの行のサブセットのいずれかから成り、サブセットの数(すなわち、`num__entry__point__offsets + 1`の値)はスライスの符号化ツリー・ブロックの行の数に等しく、スライスの符号化ツリー・ブロックの1つの行のサブセットも

30

【0077】

`tiles__or__entropy__coding__sync__idc`が2に等しいとき、スライスは、符号化ツリー・ブロックの多くの行および符号化ツリー・ブロックの行のサブセットを含む。例えば、スライスが符号化ツリー・ブロックの2つ半分の行を含む場合、サブセットの数(すなわち、`num__entry__point__offsets + 1`の値)は3に等しいであろう。

【0078】

図9は、[2]の画像パラメータ・セットRBS P構文、および[2]の画像パラメータ・セットRBS P意味論を示し、以下のように定義される：

40

【0079】

1に等しい`dependent__slice__enabled__flag`は、画像パラメータ・セットを参照して符号化画像のためのスライス・ヘッダの構文要素`dependent__slice__flag`の存在を規定する。0に等しい`dependent__slice__enabled__flag`は画像パラメータ・セットを参照して符号化画像のためのスライス・ヘッダの構文要素`dependent__slice__flag`の欠如を特定する。`tiles__or__entropy__coding__sync__idc`が3に等しいとき、`dependent__slice__enabled__flag`の値は1に等しいであろう。

【0080】

50

0に等しい `tiles_or_entropy_coding_sync_idc` は、画像パラメータ・セットを参照して各画像に1つのタイルだけがあることを示し、画像パラメータ・セットの各画像における符号化ツリー・ブロックの行の第1の符号化ツリー・ブロックを復号化する前に引き起こされる文脈の変化のための特定の同期プロセスを有し、画像パラメータ・セットを参照した符号化画像のための `cabac_independent_flag` および `dependent_slice_flag` の値は両方とも1に等しくないであろう。

【0081】

`cabac_independent_flag` および `dependent_slice_flag` は両方ともスライスのための1に等しいとき、スライスはエントロピー・スライスである。]

10

【0082】

1に等しい `tiles_or_entropy_coding_sync_idc` は、画像パラメータ・セットを参照して、各画像に1つ以上のタイルがあることが示されており、画像パラメータ・セットを参照して各画像における符号化ツリー・ブロックの行の第1の符号化ツリー・ブロックを復号化する前に引き出される文脈変化のための特定の同期プロセスはないであろうし、画像パラメータ・セットを参照した符号化画像のための `cabac_independent_flag` および `dependent_slice_flag` の値は、両方とも1に等しくはないであろう。

【0083】

20

2に等しい `tiles_or_entropy_coding_sync_idc` は、画像パラメータ・セットを参照して各画像に1つのタイルだけがあることを示しており、画像パラメータ・セットを参照して各画像における符号化ツリー・ブロックの行の第1の符号化ツリー・ブロックを復号化する前に文脈変化のための特定の同期プロセスが引き出され、画像パラメータ・セットを参照して各画像における符号化ツリー・ブロックの行の2つの符号化ツリー・ブロックを復号化した後に文脈変化のための特定の記憶プロセスが引き出され、画像パラメータを参照した符号化画像のための `cabac_independent_flag` および `dependent_slice_flag` の値は両方とも1にならないであろう。

【0084】

30

3に等しい `tiles_or_entropy_coding_sync_idc` は、画像パラメータ・セットを参照して各画像に1つだけのタイルがあることを明確にし、画像パラメータ・セットを参照して各画像における符号化ツリー・ブロックの行の第1の符号化ツリー・ブロックを復号化する前に引き出される文脈変化のための特定の同期プロセスがないであろうし、画像パラメータ・セットを参照した符号化画像のための `cabac_independent_flag` および `dependent_slice_flag` の値は、両方とも1に等しいであろう。

【0085】

`dependent_slice_enabled_flag` が0に等しいとき、`tiles_or_entropy_coding_sync_idc` は3に等しくないであろう。

40

【0086】

`tiles_or_entropy_coding_sync_idc` の値が符号化ビデオ・シーケンスの中で起動するすべての画像パラメータ・セットに共通することは、ビットストリーム適合の要件である。

【0087】

画像パラメータ・セットを参照して各スライスのために、`tiles_or_entropy_coding_sync_idc` が2に等しく、スライスの第1の符号化ブロックが符号化ツリー・ブロックの行の第1の符号化ツリー・ブロックでない場合、スライスの最後の符号化ブロックは、スライスの第1の符号化ブロックと同じ符号化ツリー・ブ

50

ックの行に属する。

【0088】

`num__tile__columns__minus 1` プラス 1 は、画像を分割しているタイルの列の数を特定する。

【0089】

`num__tile__rows__minus 1` プラス 1 は、画像を分割しているタイルの行の数を特定する。`num__tile__columns__minus 1` が 0 に等しいとき、`num__tile__rows__minus 1` は 0 に等しくないであろう。

【0090】

1 に等しい `uniform__spacing__flag` はその列境界を規定し、同様に、行境界は画像全体に一様に分布される。0 に等しい `uniform__spacing__flag` はその列境界を規定し、同様に、行境界は画像全体に一様に分布されないが、構文要素 `column__width [i]` および `row__height [i]` を使用して明確に信号を送った。

【0091】

`column__width [i]` は、符号化ツリー・ブロックを単位とする i 番目のタイルの列の幅を規定する。

【0092】

`row__height [i]` は、符号化ツリー・ブロックを単位にする i 番目のタイルの行の高さを特定する。

【0093】

ベクトル `colWidth [i]` は 0 から `num__tile__columns__minus 1` にわたっている列 i を有する CTB を単位にする i 番目のタイルの列の幅を規定し、含んでいる。

【0094】

ベクトル `CtbAddrRStoTS [ctbAddrRS]` は、0 から $(picHeightInCtbs * picWidthInCtbs) - 1$ にわたっているインデックス `ctbAddrRS` で、ラスタ・スキャン順序の CTB アドレスからタイルスキャン順序の CTB アドレスまでの会話を規定し、含んでいる

ベクトル `CtbAddrTStoRS [ctbAddrTS]` は、0 から $(picHeightInCtbs * picWidthInCtbs) - 1$ にわたっているインデックス `ctbAddrTS` で、タイルスキャン順序の CTB アドレスからラスタ・スキャン順序の CTB アドレスまでの会話を規定し、含んでいる

【0095】

ベクトル `TileId [ctbAddrTS]` は、0 から $(picHeightInCtbs * picWidthInCtbs) - 1$ にわたっている `ctbAddrTS` で、タイルスキャン順序の CTB アドレスからタイル `id` への会話を規定し、含んでいる

【0096】

`colWidth`、`CtbAddrRStoTS`、`CtbAddrTStoRS` および `TileId` の値は、入力として、サブクローズ 6.5.1 において規定されたように、CTB ラスタおよびタイルスキニングを呼び出すことによって引き出され、出力は、`colWidth`、`CtbAddrRStoTS` および `TileId` に割り当てられる。

【0097】

`luma` サンプルを単位にする i 番目のタイルの列の幅を規定する、`ColumnWidthInLumaSamples [i]` の値は、 $colWidth [i] \ll \log_2 CtbSize$ に等しく設定される。

【0098】

0 から $picWidthInMinCbs - 1$ にわたっている x 、および 0 から $picHeightInMinCbs - 1$ にわたっている y で、 z - スキャン順序における最小限の C Bs を単位にするロケーション (x, y) から最小限の CB アドレスへの会話を規

10

20

30

40

50

定するアレイ `MinCbAddrZS[x][y]` は、入力および出力が `MinCbAddrZS` に割り当てられるように、`Log2MinCbSize`、`Log2CtbSize`、`PicHeightInCtbs`、`PicWidthInCtbs`、およびベクトル `CtbAddrRstOTS` で、サブローズ 6.5.2 に規定されるような Z スキャン順序アレイ初期化プロセスを呼び出すことによって得られる。

【0099】

1 に等しい `loop_filter_across_tiles_enabled_flag` は、インループフィルタリング動作がタイル境界全体に実行されることを示している。0 に等しい `loop_filter_across_tiles_enabled_flag` は、インループフィルタリング動作がタイル境界全体に実行されないことを示している。インループ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および適応ループ・フィルタ動作を含む。ないときには、`loop_filter_across_tiles_enabled_flag` の値は 1 に等しいものと推測される。

10

【0100】

1 に等しい `cabac_independent_flag` は、スライスの符号化ブロックの CABAC 復号化が前に復号化されたスライスのいかなる様相からも独立していることを示している。0 に等しい `cabac_independent_flag` は、スライスの符号化ブロックの CABAC 復号化が前に復号化されたスライスの様相に依存していることを示している。ないときには、`cabac_independent_flag` の値は 0 に等しいものであると推測される。

20

【0101】

最小限の符号化ブロック・アドレスを有する符号化ブロックの入手可能性のための派生プロセスは、以下の通りに説明されていた：

このプロセスへの入力は、次のようなものである。

- z - スキャン順序の最小限の符号化ブロック・アドレス `minCbAddrZS`

- z - スキャン順序の現在の最小限の符号化ブロック・アドレス `currMinCbAddrZS`

【0102】

このプロセスの出力は、z - スキャン順序 `cbAvailable` における最小限の符号化ブロック・アドレス `cbAddrZS` を有する符号化ブロックの入手可能性である。

30

【0103】

このプロセスが呼び出されるときに、有効性の意味が決定されることに注意されたい。

【0104】

そのサイズに関係なく、いかなる符号化ブロックも、z - スキャン順序の最小限の符号化ブロック・サイズを有する符号化ブロックのアドレスである最小限の符号化ブロック・アドレスと関係していることに注意されたい。

【0105】

- 以下の状況の 1 つ以上が真である場合、`cbAvailable` は FALSE にセットされる。

40

- `minCbAddrZS` は、0 未満である

- `minCbAddrZS` は、`currMinCbAddrZS` より大きい

- 最小限の符号化ブロック・アドレス `minCbAddrZS` を有する符号化ブロックは、現在の最小限の符号化ブロック・アドレス `currMinCbAddrZS` を有する符号化ブロックに属し、現在の最小限の符号化ブロック・アドレス `currMinCbAddrZS` を有する符号化ブロックを含んでいるスライスの `dependent_slice_flag` と異なるスライスは 0 に等しい。

- 最小限の符号化ブロック・アドレス `minCbAddrZS` を有する符号化ブロックは、現在の最小限の符号化ブロック・アドレス `currMinCbAddrZS` を有する符号化ブロックとは異なるタイルに含まれる。

50

- さもなければ、`cbAvailable`は、`TRUE`にセットされる。

【0106】

〔2〕のスライス・データのためのCABAC構文解析プロセスは、以下の通りだった：

記述子`ae(v)`を有する構文要素を解析するときに、このプロセスは呼び出される。

【0107】

このプロセスへの入力は、構文要素の値および前に解析された構文要素の値の要求である。

【0108】

このプロセスの出力は、構文要素の値である。

10

【0109】

スライスのスライス・データの構文解析を始めるときに、CABAC構文解析プロセスの初期化プロセスは呼び出される。

【0110】

空間隣接ブロック T (図10a)を含んでいる符号化ツリー・ブロックの最小限の符号化ブロック・アドレス、`ctbMinCbAddrT`は、以下のように現在の符号化ツリー・ブロックの左上`luma`サンプルのロケーション (x_0, y_0) を使用して引き出される。

$x = x_0 + 2 \ll \text{Log2CtbSize} - 1$

$y = y_0 - 1$

$\text{ctbMinCbAddrT} = \text{MinCbAddrZS}[x \gg \text{Log2MinCbSize}][y \gg \text{Log2MinCbSize}]$

20

【0111】

可変`availableFlagT`は、入力として`ctbMinCbAddrT`を有する符号化ブロック有効性派生プロセスを呼び出すことによって得られる。

【0112】

符号化ツリーの構文解析を始めるときに、以下の手順が適用される。

1. 算術復号化エンジンは、以下のように初期化される。

- `CtbAddrRS`が`slice_address`に等しい場合、`dependent_slice_flag`は1に等しく、`entropy_coding_reset_flag`は0に等しく、以下が適用される。

30

- CABAC構文解析プロセスの同期化プロセスは、入力としての`TableStateIdxDS`および`TableMPSValDS`によって呼び出される。

- 終了の前のバイナリの決定のための復号化プロセスは呼び出され、算術復号化のための初期化プロセスが続く。

- さもなければ、`tiles_or_entropy_coding_sync_idc`が2に等しく、`CtbAddrRS % PicWidthInCtbs`が0に等しい場合、以下が適用される。

- `availableFlagT`が1に等しいとき、CABAC構文解析プロセスの同期化プロセスは、入力としての`TableStateIdxWPP`および`TableMPSValWPP`によって呼び出される。

40

- 終了の前のバイナリの決定のための復号プロセスは呼び出され、算術復号化エンジンのための初期化プロセスが続く。

【0113】

2. `cabac_independent_flag`が0に等しく、`dependent_slice_flag`が1に等しいとき、または、`tiles_or_entropy_coding_sync_idc`が2に等しいとき、記憶プロセスは以下のように適用される。

- `tiles_or_entropy_coding_sync_idc`が2に等しく、`CtbAddrRS % PicWidthInCtbs`が2に等しいとき、CABAC構文解析プロセスの記憶プロセスは出力として`TableStateIdxWPP`および`T`

50

ableMPSValWPPによって呼び出される。

- cabac_independent_flagが0に等しく、dependent_slice_flagが1に等しく、end_of_slice_flagが1に等しいとき、CABC構文分析プロセスの記憶プロセスは出力としてTableStateIdxDSおよびTableMPSValDSによって呼び出される。

【0114】

構文要素の構文解析は、以下の通りに進行する：

【0115】

構文要素の要求された値ごとに、2値化が得られる。

【0116】

構文要素のための2値化および構文解析されたピン(bin)のシーケンスは、復号化プロセスフローを決定する。

【0117】

可変binIdxによってインデックスを付けられる構文要素の2値化のピンごとに、文脈インデックスctxIdxが得られる。

【0118】

ctxIdxごとに、算術復号化プロセスが起動する。

【0119】

構文解析されたピンの結果として生じるシーケンス(b0...bbinIdx)は、各ピンの復号化の後、2値化プロセスによって与えられるピンスtringのセットと比較される。与えられたセットにおいてシーケンスがピンに一致するときに、対応する値は構文要素に割り当てられる。

【0120】

構文要素の値の要求が構文要素pcm_flagおよびpcm_flagの復号化値が1に等しい場合、復号化エンジンはいかなるpcm_alignment_zero_bitも、num_subsequent_pcmおよびすべてのpcm_sample_lumaおよびpcm_sample_chromaデータを復号化した後に初期化される。

【0121】

今までに記載されている設計フレームワークにおいて、以下の課題が発生した。

【0122】

復号化ユニットのタイミングは、低遅延シナリオでデータを符号化し送信する前に知られている必要があり、NALユニットはすでにエンコーダによって送られ、その一方で、エンコーダは画像、すなわち、他のサブピクチャ復号化ユニットをまだ符号化している。これは、アクセスユニットにおけるNALユニット順序が、アクセスユニットにおけるVCL(ビデオ符号化NALユニット)に先行するSEIメッセージを許しているだけであって、そのような低遅延シナリオにおいて、エンコーダが復号化ユニットの符号化を開始している場合、非VCL NALユニットはすでに通信、すなわち送信されている必要があるためである。図10bは、〔2〕に記載のアクセスユニットの構造を例示する。〔2〕はまだシーケンスまたはストリームの端部を特定しておらず、そのため、アクセスユニットのそれらの存在は暫定的であった。

【0123】

また、サブピクチャに関連するNALユニットの数が低遅延シナリオにおいて予め知られていなければならない、それは、画像タイミングSEIがこの情報を含み、従わなければならない、エンコーダが実際の画像を符号化し始める前に送信しなければならないためである。アプリケーションデザイナーは、NALユニットナンバーに適合するフィルタデータが潜在的に存在しないフィルタデータNALユニットをしつこく挿入するが、それは、画像タイミングSEIの復号化ユニットごとに送信され、サブピクチャ・レベル上のこの情報を送信する手段を必要とするからである。それは、タイミングSEIメッセージにおいて与えられるパラメータによってアクセスユニットの存在で現在確定されるサブピクチャ

10

20

30

40

50

・タイミングのために保持する。

【 0 1 2 4 】

さらに、仕様書案 [2] の更なる欠点は、例えば R O I シグナリングまたはタイル次元シグナリングのように、特殊利用のために必要とされるサブピクチャ・レベルの多数のシグナリングを含む。

【 先行技術文献 】

【 非特許文献 】

【 0 1 2 5 】

【 非特許文献 1 】 Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Syst. Video Technol., vol. 13, N7, July 2003. 10

【 非特許文献 2 】 JCT-VC, "High-Efficiency Video Coding (HEVC) text specification Working Draft 7", JCTVC-I1003, May 2012.

【 非特許文献 3 】 ISO/IEC 13818-1: MPEG-2 Systems specification.

【 非特許文献 4 】 IETF RFC 3550 - Real-time Transport Protocol.

【 非特許文献 5 】 Y.-K. Wang et al., "RTP Payload Format for H.264 Video", IETF RFC 6184, <http://tools.ietf.org/html/>

【 非特許文献 6 】 S. Wenger et al., "RTP Payload Format for Scalable Video Coding", IETF RFC6190, <http://tools.ietf.org/html/rfc6190>

【 非特許文献 7 】 T. Schierl et al., "RTP Payload Format for High Efficiency Video Coding", IETF internet draft, <http://datatracker.ietf.org/doc/draft-schierl-payload-rtp-h265/> 20

【 発明の概要 】

【 発明が解決しようとする課題 】

【 0 1 2 6 】

上記の概説された課題は、HEVC基準に特有のものではない。むしろ、この課題は、同様に他のビデオ・コーデックと関連して発生する。さらに一般的に言えば、図 1 1 は、短時間の端末間遅延でエンコーダ 1 0 からデコーダ 1 2 にビデオ 1 6 を送信するためにネットワーク 1 4 を介して一対のエンコーダ 1 0 およびデコーダ 1 2 が接続されているビデオ伝送風景を示す。すでに上で概説される課題は次のようなものである。エンコーダ 1 0 は、必然的にではなく実質的に、フレーム 1 8 の再現順序 2 0 に続き、各フレーム 1 8 内で、例えばフレーム 1 8 のタイル・セクショニング方法を有するあるいは有していないラスト・スキャンのようなくつかの特定の 방법으로フレーム 1 8 のフレーム領域を通して伝わる特定の復号化順序に従って、ビデオ 1 6 のフレーム 1 8 のシーケンスを符号化する。復号化順序は、例えば予測および/またはエントロピー符号化、すなわち、予測または文脈選択の基礎として役立つ利用可能なビデオ 1 6 の空間的および/または時間的に隣接する部分に関する情報の有効性のよう、エンコーダ 1 0 によって用いられる符号化技術のための情報の有効性を制御する。エンコーダ 1 0 がビデオ 1 6 のフレーム 1 8 を符号化するために並列処理を使用することが可能かもしれない場合であっても、エンコーダ 1 0 は必然的に、現行フレームのような特定のフレーム 1 8 を符号化する若干の時間を必要とする。図 1 1 は、例えば、エンコーダ 1 0 がすでに現行フレーム 1 8 の部分 1 8 a を符号化し終わった瞬間を示し、その一方で、現行フレーム 1 8 の他の部分 1 8 b はまだ符号化されなかったことを示す。エンコーダ 1 0 が部分 1 8 b をまだ符号化しなかったため、エンコーダ 1 0 は現行フレーム 1 8 を符号化するための利用可能なビットレートがどのように、例えば、率/歪曲認識力に関して最適条件を成し遂げるために現行フレーム 1 8 を通じて空間的に分配されるべきかについて予測することができない。したがって、エンコーダ 1 0 は、単に 2 つの選択を有するだけである：エンコーダ 1 0 は、現在のフレーム 1 8 が前もって空間的にさらに細かく分割されるスライス上に現在のフレーム 1 8 のための利用可能なビットレートの最適に近い分配を推測し、したがって推測は間違っていると認めるか、または、エンコーダ 1 0 は、エンコーダ 1 0 からデコーダ 1 2 にスライスを含むパケ 40 50

ットを送信する前に現在のフレーム 18 の符号化を終了させているかのどちらかである。

いずれにせよ、その符号化の終了の前に現在の符号化フレーム 18 のスライス・パケットのいかなる伝送も利用することが可能であるために、ネットワーク 14 は、符号化ピクチャ・バッファ検索時間の形でこの種の各スライス・パケットと関連したビットレートを知らせるべきである。しかしながら、上述のように、エンコーダ 10 は、HEVC の現在バージョンに従って、個々にサブピクチャ領域のためのデコーダバッファ検索時間を定めることを用いてフレーム 18 上に分配されるビットレートを変化させることが可能であるにもかかわらず、エンコーダ 10 は、現在のフレーム 18 に関する全てのデータを集める各アクセスユニットの初めにおいてネットワーク 14 を介してデコーダ 12 にそのような情報を送信または送る必要があり、それにより、エンコーダ 10 に概説した選択肢の中から選択することを促し、一方は低い遅延であるが悪い率 / 歪曲をリードし、他方は最適な率 / 歪曲であるが増加した端末間遅延をリードする。

10

【0127】

このように、これまで、エンコーダが現在のフレームの残りの部分 18 b を符号化するのに先立って現在のフレームの部分 18 a に関するパケットの送信の開始を可能にし、デコーダがエンコーダ 12 からデコーダ 14 に送るビデオ・データストリーム内で伝達される復号化バッファ検索タイミングに従うネットワーク 16 による予備部分 18 a に関してパケットのこの中間伝達を利用するような低遅延の成績を可能にしたビデオ・コーデックはなかった。例えば、見本としてこの種の低い遅延を利用するアプリケーションは、例えばオートメーションまたは点検目的等のためのワークピースまたは製作監視のような産業アプリケーションを含む。これまで、ネットワーク 16 の中の中間のネットワーク存在物がパケット、すなわちスライス構文の内部を深く検査する必要のないデータストリームからこの種の情報を集めることを可能にされるように、現在のフレームが構成され、領域（関心領域）に関心を持たせるタイルへのパケットの関連上の復号化側に知らせるための十分な解決もない。

20

【0128】

したがって、この発明の目的は、低い端末間遅延を可能にする際により効率的でおよび / または関心領域に、または、特定のタイルにデータストリームの部分の識別がより容易になるビデオ・データストリーム符号化コンセプトを提供することである。

【課題を解決するための手段】

30

【0129】

この目的は、特許請求の範囲の独立請求項の主題によって達成される。

【0130】

本出願が基礎とする 1 つの考えは、デコーダ検索タイミング情報、ROI 情報およびタイル識別情報が MANES またはデコーダのようなネットワークエンティティによって簡単なアクセスを可能にするレベルでビデオ・データストリームの中で伝達されなければならないということであり、この種のレベルに達するために、この種のタイプの情報がビデオ・データストリームのアクセスユニットのパケットに分散するパケットを経由してビデオ・データストリームの中で伝達されなければならないということである。実施例によれば、分散型のパケットは、除去可能なパケット・タイプの中にある、すなわち、これらの分散型のパケットの除去は、完全に、ビデオ・データストリームを介して運搬されるビデオ内容を回復するデコーダの能力を維持する。

40

【0131】

本出願の態様によれば、低い端末間遅延の達成は、現在のアクセスユニットの範囲内でビデオ・データストリームのそれぞれのタイミング制御パケットに続くペイロード・パケットによって形成される復号化ユニットのためのデコーダバッファ検索時間に情報を伝達するために、分散型のパケットを用いることによってより効果的になる。この計測によって、エンコーダは、現在のフレームを符号化する間、その場でデコーダバッファ検索時間を決定することを可能にされ、現在のフレームを符号化する間、実はすでにペイロード・パケットに符号化され、送信された現在のフレームの部分に費やされるか、または、一方

50

で、タイミング制御パケットによって前に置かれ、したがって、まだ符号化されていなかった現在のフレームの残りの部分上に現在のフレームに利用できる残りのビットレートの分配を適応させるビットレートを連続的に決定することができるようになる。この計測によって、利用可能なビットレートは効果的に利用され、そして、エンコーダが完全に現在のフレームを符号化し終わるために待つ必要がないため、遅延はそれにもかかわらずより短く保たれる。

【0132】

本出願の更なる態様によれば、アクセスユニットのペイロード・パケットに分散するパケットは関心領域に情報伝達するために利用され、それによって、上述したように、中間のペイロード・パケットを検査する必要がないため、ネットワークエンティティによってこの情報の簡単なアクセスが可能になる。さらに、エンコーダは、前もってサブポーションおよびそれぞれのペイロード・パケットに現在のフレームの細別を決定する必要がなく、現在のフレームを符号化する間、臨機応変にROIに属しているパケットを決定することがまだできる。さらに、分散型のパケットが移動可能なパケット・タイプである実施例によれば、ROI情報は、ROI情報に関心がないか、それを処理することができないビデオ・データストリームの受容者によって無視されることができる。

10

【0133】

類似の考えは、分散型のパケットがアクセスユニットの範囲内の特定のパケットがどのタイルに属するかについて情報伝達する他の態様に従う本出願において利用される。

20

【0134】

本発明の有利な実施態様は、従属クレームの主題である。

本出願の好ましい実施例は、以下の図を参照して更に詳細に記載されており、図1ないし図10bはHEVCの現在の状態を示している。

【図面の簡単な説明】

【0135】

【図1】図1は、バッファ期間SEIメッセージ構文を示す。

【図2】図2は、画像タイミングSEIメッセージ構文を示す。

【図3a】図3aは、VUIパラメータ構文を示す。

【図3b】図3bは、VUIパラメータ構文を示す。

【図4】図4は、HRDパラメータ構文を示す。

30

【図5】図5は、フィルタデータRBS P構文を示す。

【図6】図6は、HRD一致チェックのためのバイトストリームおよびNALユニットストリームの構造を示す。

【図7】図7は、HRDバッファ・モデルを示す。

【図8】図8は、スライス・ヘッダ構文を示す。

【図9】図9は、画像パラメータ・セットRBS P構文を示す。

【図10a】図10aは、場合により現在の符号化ツリー・ブロックに関連する符号化ツリー・ブロック利用誘導プロセスを起動させるために用いられる空間的に隣接する符号化ツリー・ブロックTの図解図を示す。

【図10b】図10bは、アクセスユニットの構造の定義を示す。

40

【図11】図11は、図式的に、ビデオ・データストリーム伝送で起こっている課題を例示するためのネットワークを介して接続される一対のエンコーダおよびデコーダを示す。

【図12】図12は、タイミング制御パケットを使用している実施例に従うエンコーダの概略ブロック図を示す。

【図13】図13は、実施例に従う図12のエンコーダの作動モードを例示しているフロー図を示す。

【図14】図14は、図12に従ってエンコーダによって発生するビデオ・データストリームと関連してその機能を説明するためにデコーダの実施例のブロック図を示す。

【図15】図15は、ROIパケットを使用している更なる実施例に従ってエンコーダ、ネットワーク実体およびビデオ・データストリームを例示している概略ブロック図を示す

50

。

【図 16】図 16 は、タイル識別パケットを使用している更なる実施例に従ってエンコーダ、ネットワークエンティティおよびビデオ・データストリームを例示している概略ブロック図を示す。

【図 17】図 17 は、実施例に従ってアクセスユニットの構造を示す。点線は、任意のスライス接頭辞 NAL ユニットのケースを反映する。

【図 18】図 18 は、関心シグナリングの領域のタイルの使用を示す。

【図 19】図 19 は、第 1 の単純な構文 / バージョン 1 を示す。

【図 20】図 20 は、SEI メッセージコンセプトとは異なる `tile_id signaling`、復号化ユニットスタート識別子、スライス接頭辞 ID およびスライス・ヘッダデータを含む拡張した構文 / バージョン 2 を示す。

【図 21】図 21 は、NAL ユニットのタイプ・コードおよび NAL ユニットのタイプクラスを示す。

【図 22】図 22 は、スライス・ヘッダのための可能な構文を示し、現在のバージョンに従うスライス・ヘッダに存在する特定の構文要素は、`slice_header_data` と呼ばれる下部の階層構文要素へ移される。

【図 23 a】図 23 a は、スライス・ヘッダから離れたすべての構文要素が構文要素スライス・ヘッダデータによって信号を送られるテーブルを示す。

【図 23 b】図 23 b は、スライス・ヘッダから離れたすべての構文要素が構文要素スライス・ヘッダデータによって信号を送られるテーブルを示す。

【図 23 c】図 23 c は、スライス・ヘッダから離れたすべての構文要素が構文要素スライス・ヘッダデータによって信号を送られるテーブルを示す。

【図 24】図 24 は、補足的な強化情報メッセージ構文を示す。

【図 25 a】図 25 a は、新しいスライスまたはサブピクチャ SEI メッセージタイプを導入するために、構成される SEI ペイロード構文を示す。

【図 25 b】図 25 b は、新しいスライスまたはサブピクチャ SEI メッセージタイプを導入するために、構成される SEI ペイロード構文を示す。

【図 26】図 26 は、サブピクチャ・バッファ SEI メッセージのための実施例を示す。

【図 27】図 27 は、サブピクチャ・タイミング SEI メッセージのための実施例を示す

。

【図 28】図 28 は、サブピクチャスライス情報 SEI がどのように見えるかを示す。

【図 29】図 29 は、サブピクチャ・タイル情報 SEI メッセージのための実施例を示す

。

【図 30】図 30 は、サブピクチャ・タイル範囲情報 SEI メッセージのための構文の例を示す。

【図 31】図 31 は、各 ROI が個々の SEI メッセージにおいて信号を送られる関心領域 SEI メッセージのための構文の例の第 1 の変異型を示す。

【図 32】図 32 は、すべての ROI が単一の SEI メッセージにおいて信号を送られる関心領域 SEI メッセージのための構文の例の第 2 の変異型を示す。

【図 33】図 33 は、更なる実施例に従うタイミング制御パケットのための可能な構文を示す。

【図 34】図 34 は、実施例に従うタイル識別パケットのための可能な構文を示す。

【図 35】図 35 は、実施例に従う異なる細別設定に従う画像の可能な細別を示す。

【図 36】図 36 は、実施例に従う異なる細別設定に従う画像の可能な細別を示す。

【図 37】図 37 は、実施例に従う異なる細別設定に従う画像の可能な細別を示す。

【図 38】図 38 は、実施例に従う異なる細別設定に従う画像の可能な細別を示す。

【図 39】図 39 は、アクセスユニットのペイロード・パケットとの間に分散しているタイミング制御パケットを使用している実施例に従うビデオ・データストリームからの部分の例を示す。

【発明を実施するための形態】

10

20

30

40

50

【0136】

図12に関して、本出願およびその作動モードの実施例に従うエンコーダ10が記載されている。エンコーダ10は、ビデオ内容16をビデオ・データストリーム22に符号化するように構成される。エンコーダは、ビデオ内容16のフレーム/画像18のサブポーションを単位にしてこれをするように構成され、サブポーションは、例えば、画像18が分割されるスライス24、または例えばタイル26またはWPPサブストリーム28のようないくつかの他の空間部分でもよく、その全ては、エンコーダ10が、例えば、タイルまたはWPP並列処理をサポートすることが可能なことを必要とする、または、サブポーションがスライスであることを必要とすることを示唆しているというよりはむしろ、図示する目的だけのために単に図12において例示される。

10

【0137】

サブポーション24を単位にするビデオ内容16を符号化することにおいて、エンコーダ10は、復号化順序に沿ってそのようなブロックの連続運転を示すサブポーションをもって、画像18がラスタ・スキャン順序に従って分配される、例えば、画像18の間で規定される再現順序20と必ずしも一致せず、および各画像18ブロック内で妨害される、例えば、画像復号化順序に従ってビデオ16の画像18を妨害するサブポーション24内で定められる復号化順序 - あるいは符号化順序 - に従うことができる。特に、エンコーダ10は、例えば、予測および/またはエントロピー文脈を決定するように、予測符号化および/またはエントロピー符号化におけるそのような隣接部分を示す特性を用いるために符号化させるべき現在の部分に空間的および/または時間的に隣接する部分の有用性を決定するこの復号化順序に従うように構成される：単にビデオの前に訪れる（符号化/復号化）だけだった部分は利用可能である。さもなければ、ちょうど言及された特性はデフォルト値に設定され、または、いくつかの他の代替手段がとられる。

20

【0138】

一方では、エンコーダ10は、復号化順序に沿って連続的にサブポーション24を符号化する必要はない。むしろ、エンコーダ10は符号化プロセスのスピードアップを図るため並行処理を行ったり、リアルタイムでより複合的な符号化を実行することが可能である。同様に、エンコーダ10は、復号化順序に沿ってサブポーションを符号化しているデータを伝えるかまたは送るように構成されてもよいし、構成されなくてもよい。たとえば、エンコーダ10は、サブポーションの符号化が、並列処理により、例えば、ちょうど言及される復号化順序から逸脱することができるエンコーダ10によって終了する順序に従うような、いくつかの他の順序で符号化されたデータを出力/送信することができる。

30

【0139】

サブポーション24の符号化されたバージョンをネットワーク上の伝送に適するようにするために、エンコーダ10は、ビデオストリーム22のパケットのシーケンスの1つ以上のペイロード・パケットに各サブポーション24を符号化する。サブポーション24がスライスである場合、エンコーダ10は、例えば、各スライス・データ、例えば、各符号化データをNALユニットなどの1つのペイロード・パケットに置き換えるように構成される。このパケット化は、ネットワークを介してビデオ・データストリーム22が伝送に適するようになるのに役立つことができる。したがって、パケットは、ビデオ・データストリーム22が起こる、すなわち受取人に対するネットワークを経て伝送用のエンコーダ10によって個々に送られる最も小型のユニットを表す。

40

【0140】

さらに、それらの間に分散するペイロード・パケットおよびタイミング制御パケットや、以下で述べる他のパケット、例えば、まれに変化する構文要素またはEOF（ファイルの終わり）またはAUE（アクセスユニット端）パケット等の伝達するためのフィルデータ・パケット、画像またはシーケンス・パラメータのような他のタイプのパケットが存在する。

【0141】

パケットのシーケンスがアクセスユニット30のシーケンスに分けられ、各アクセスユ

50

ビットがビデオ内容 16 の 1 つの画像 18 に関するペイロード・パケット 32 を集めるように、エンコーダはペイロード・パケットへの符号化を実行する。すなわち、ビデオ・データストリーム 22 を形成しているパケットのシーケンス 34 は、それぞれが画像 18 のそれぞれの 1 つに関連している、アクセスユニット 30 と呼ばれる重なり合わない部分に再分割される。アクセスユニット 30 のシーケンスは、アクセスユニット 30 が関連する画像 18 の復号化順序に従う。図 12 は、例えば、図示されたデータストリーム 22 の部分の中央に配置されるアクセスユニット 30 が、画像 18 が再分割されるサブポーション 24 につき 1 つのペイロード・パケット 32 を含む。すなわち、各ペイロード・パケット 32 は、対応するサブポーション 24 を担持する。エンコーダ 10 は、パケットタイミング制御パケット 36 のシーケンス 34 に分散され、それによりタイミング制御パケットがアクセスユニット 30 を復号化ユニット 38 に細分割し、それにより図 12 の中央部に示されているように、少なくともいくつかのアクセスユニット 30 は 2 つ以上の復号化ユニット 38 に細分割され、各タイミングコントロールパケットは、そのペイロード・パケット 32 がパケットのシーケンス 34 におけるそれぞれのタイミング制御パケットに続く復号化ユニット 38 のためのデコーダバッファ検索時間に信号を送る。換言すれば、エンコーダ 10 は、それぞれのタイミング制御パケット 36 によって前に置かれ、復号化ユニット 38、デコーダバッファ検索時間を形成するペイロード・パケットのそれぞれのサブシーケンスのために信号を送るそれぞれのタイミング制御パケット 36 で 1 つのアクセスユニット 30 内のペイロード・パケット 32 のシーケンスのサブシーケンスの前に置く。図 12 は、例えば、あらゆる第 2 のパケット 32 がアクセスユニット 30 の復号化ユニット 38 の第 1 のペイロード・パケットを表すケースを例示する。図 12 に示すように、データの量または復号化ユニット 38 ごとに費やされるビットレートは変化し、そして、復号化ユニット 38 のデコーダバッファ検索時間がこの直ちに前の復号化ユニット 38 に費やされるビットレートに対応する時間間隔に加えて直ちに前の復号化ユニット 38 のタイミング制御パケット 36 によって信号を送られるデコーダバッファ検索時間に続くことができるという点で、デコーダバッファ検索時間は復号化ユニット 38 の中のこのビットレートのバリエーションと関連することができる。

【0142】

すなわち、エンコーダ 10 は、図 13 に示すように作動することができる。特に、上述したように、エンコーダ 10 は、ステップ 40 において、現在の画像 18 の現在のサブポーション 24 を符号化させることができる。すでに述べたように、エンコーダ 10 は、矢印 42 で示すように、上述の復号化順序のサブポーション 24 によって順次循環することができ、または、エンコーダ 10 は並行していくつかの「現在のサブポーション」24 を同時に符号化するために WPP および / またはタイル処理のようないくらかの並列処理を使用することができる。並列処理の使用の有無にかかわらず、エンコーダ 10 は、ステップ 40 およびそれに続くステップ 44 において符号化される 1 つまたはいくつかのサブポーションから復号化ユニットを形成し、エンコーダ 10 は、デコーダバッファ検索時間をこの復号化ユニットに設定して伝送し、前に置かれたこの復号化ユニットは時間制御パケットで復号化ユニットのために設定された復号化バッファ検索時間の信号を送る。たとえば、エンコーダ 10 は、もしあれば、この復号化ユニット内の全ての更なる中間パケット、すなわち「接頭辞パケット」を含む現在の復号化ユニットを形成するペイロード・パケットに符号化されたサブポーションを符号化するのに費やされるビットレートに基づいてステップ 44 におけるデコーダバッファ検索時間を決定する。

【0143】

それから、ステップ 46 において、エンコーダ 10 は、ちょうど今ステップ 44 において送信されていた復号化装置に費やされるビットレートに基づいて、利用できるビットレートを適応させることができる。例えば、ステップ 44 においてちょうど送信される復号化ユニット内の画像内容が圧縮率に関して非常に複雑である場合、エンコーダ 10 は、例えば、ビデオ・データストリーム 22 を送信しているネットワークと関連して直面される現在の帯域幅状況に基づいて、決定されたいくつかの外部的に設定された目標ビット

レートに従うように、次の復号化ユニットのために利用できるビットレートを減らすことができる。ステップ 40 ~ 46 は、それから繰り返される。この計測によって、画像 18 は符号化されて送信される、すなわち、各々対応するタイミング制御パケットによって前に置かれた、復号化ユニットを単位にして送信される。

【0144】

換言すれば、エンコーダ 10 は、ビデオ内容 16 の現在の画像 18 を符号化する間、現在の画像 18 の現在のサブポジション 24 を現在の復号化ユニット 28 の現在のペイロード・パケット 32 に符号化 40 し、最初の瞬間、データストリームの中で、現在のタイミング制御パケット (36) によって信号を送られるデコーダバッファ検索時間で現在のタイミング制御パケット 36 で前に置かれる現在の復号化ユニット 38 をデータストリーム内で伝送 44 し、ステップ 46 から 40 に回帰することにより、第 1 の時間的瞬間 - 第 1 の時間訪問ステップ 44 より遅い第 2 の時間的瞬間 - 第 2 の時間訪問ステップに現在の画像 18 の更なるサブポジション 24 を符号化 44 する。

【0145】

エンコーダはこの復号化ユニットが属する現在の画像の残りの符号化に先立って復号化ユニットを伝送することができるので、エンコーダ 10 は端子間遅延を少なくすることができる。一方では、エンコーダ 10 が現在の画像の内容および複雑さのその空間分布の特定の性質に反応することが可能であるため、エンコーダ 10 は利用できるビットレートを浪費する必要がない。

【0146】

一方において、更にエンコーダからデコーダにビデオ・データストリームの伝送に対して重要な中間ネットワークエンティティは、エンコーダ 10 による復号化ユニット的な符号化および伝送の利益を得るように、ビデオ・データストリーム 22 を受信するいくつかのデコーダが時間内に復号化ユニットを受信することを保証するように、タイミング制御パケット 36 を使用することができる。たとえば、ビデオ・データストリーム 22 を復号化するためのデコーダを示す図 14 を参照されたい。デコーダ 12 は、エンコーダ 10 がデコーダ 12 にビデオ・データストリーム 22 を伝送したネットワークを經由して符号化画像バッファ CPB 48 でビデオ・データストリーム 22 を受信する。特に、ネットワーク 14 は低遅延アプリケーションをサポートすることができるものと見なされるため、ネットワーク 10 はデコーダ 12 の符号化画像バッファ 48 にビデオ・データストリーム 22 のパケットのシーケンス 34 を送るようにデコーダバッファ検索時間を検査し、それにより、各復号化ユニットはそれぞれの復号化ユニットの前におかれるタイミング制御パケットによって信号を送られるデコーダバッファ検索時間に先立って符号化画像バッファ 48 内に存在する。この計測によって、デコーダは、行き詰まることなく、すなわち、符号化画像バッファ 48 内で利用できるペイロード・パケットが尽きることなく、完全なアクセスユニットよりむしろ復号化ユニットを単位にするデコーダの符号化ピクチャ・バッファ 48 を空にするために、タイミング制御パケットのデコーダバッファ検索時間を使用する。図 14 は、例えば、符号化画像バッファ 48 の出力に接続され、その入力ビデオ・データストリーム 22 を受信するような処理ユニット 50 を例示的に示している。エンコーダ 10 と同様に、デコーダ 12 は、例えばタイル並列処理 / 復号化および / または W P 並列処理 / 復号化を用いて並列処理を実行することができる。

【0147】

以下において詳述するように、デコーダバッファ検索時間は、デコーダ 12 の符号化ピクチャ・バッファ 48 に関する検索時間に必ずしも関連するというわけではない。むしろ、タイミング制御パケットは、加えて、あるいは、デコーダ 12 の対応する復号化ピクチャ・バッファのすでに復号化された画像データの検索を進めることができる。図 14 は、例えば、処理ユニット 50 によりビデオ・データストリーム 22 を復号化することによって得られるビデオ内容の復号化バージョンがバッファリングされる、したがって、復号化ユニットの復号化バージョンを単位にして格納され、出力されるデコーダピクチャバッファを含むデコーダ 12 を示す。デコーダの復号化ピクチャ・バッファ 22 は、このように

デコーダ 12 の出力および処理ユニット 50 の出力の間に接続される。復号化ピクチャ・バッファ 52 からユニットを復号化する復号化バージョンを出力するための検索時間を設定する能力を有することによって、エンコーダ 10 は、臨機応変に、すなわち、現在の画像を符号化する間、再現を制御し、または端末間で、画像率またはフレーム率より小さい粒度でさえ復号化側のビデオ内容の再現の遅延を得る機会を与えられる。明らかに、符号化側で各画像 18 を大量のサブポーション 24 に過度に分割することはビデオ・データストリーム 22 を送信するためのビットレートに負の影響を及ぼすが、その一方で、このような復号化ユニットを符号化し、送信し、復号化し、出力するために必要な時間が最小化されるため、端末間の遅延は最小化されることができる。一方では、サブポーション 24 のサイズを増加させることは、端末間の遅延を増加させる。したがって、妥協は見つけられなければならない。復号化ユニットを単位とするサブポーション 24 の復号化バージョンの出力タイミングを誘導するように言及したデコーダバッファ検索時間を使用することは、エンコーダ 10 または符号化側のいくつかの他のユニットに現在の画像の内容を空間的に超えてこの妥協を適用することを可能にする。この手段によって、それが現在の画像内容全体で空間的に変化させるそのような方法における端末間遅延を制御することは可能である。

10

20

30

40

50

【0148】

上述のような実施例を実行する際に、タイミング制御パケットとして、除去可能なパケット・タイプのパケットを使用することが可能である。除去可能なパケット・タイプのパケットは、復号側でビデオ内容を回復するために必要でない。以下に、この種のパケットは、SEIパケットと呼ばれている。さらに、除去可能なパケット・タイプのパケット、すなわち、ストリームにおいて伝送される場合に、冗長パケットのような別のタイプの除去可能なパケットが同様に存在する。他の方法として、タイミング制御パケットは特定の除去可能なパケット・タイプのパケットでもよいが、加えて、特定のSEIパケット・タイプフィールドを担持する。たとえば、タイミング制御パケットは、1つまたはいくつかのSEIメッセージを担持している各SEIパケットを有するSEIパケットであり、特定のタイプのSEIメッセージを含むこれらのSEIパケットだけが前述のタイミング制御パケットを形成する。

【0149】

このように、図 12 ~ 14 に関して今までに記載されている実施例は、更なる実施例に従って、HEVC基準に適用され、それによって、より低い端末間遅延においてより効果的なHEVCを提供するための可能なコンセプトを形成する。この際、上述したパケットはNALユニットによって形成され、そして、上述したペイロード・パケットは上述したサブポーションを形成するスライスを含むNALユニットストリームのVCL NALユニットである。

【0150】

より詳細な実施例の説明の前に、分散されたパケットが効率的な方法でビデオ・データストリームを示している情報を伝送するために用いられるという点で上述の概説された実施例と一致する更なる実施例が示されるが、情報の分類が、タイミング制御パケットがデコーダバッファ検索タイミング情報に伝送された上述の実施例と異なる。さらに以下に記載されている実施例において、アクセスユニットに属しているペイロード・パケットに分散する分散パケットを介して移される情報の種類は、関心領域 (ROI) 情報および/またはタイル識別情報に関する。さらに以下に記載される実施例は、図 12 ~ 14 に関して記載されている実施例と結合されることができるとは結合されることができない。

【0151】

図 15 は、図 15 のエンコーダ 10 に対してオプションとなる、図 13 に関して上述されたタイミング制御パケットおよび機能の分散を除いて、図 12 に関して上で説明したものに類似の動作をするエンコーダ 10 を示す。しかしながら、図 15 のエンコーダ 10 は、図 11 に関連して上で説明したように、ビデオ内容 16 の画像 18 のサブポーション 24 を単位とするビデオ・データストリーム 22 にビデオ内容 16 を符号化するように構成

される。ビデオ内容 16 を符号化する際に、エンコーダ 10 は、ビデオ・データストリーム 22 とともに、復号化側に関心領域 ROI 60 に関する情報を伝達することに興味を持っている。ROI 60 は、デコーダが、例えば、特別に注意を払わなければならない、現在の画像 18 の空間サブエリアである。ROI 60 の空間位置は、現在の画像 18 の符号化の間、ユーザーの入力などのような、点線 62 で示されるような外部からエンコーダ 10 に入力することができ、または、エンコーダ 10 または臨機応変にいくつかの他の存在により自動的に決定されることができる。いずれにせよ、エンコーダ 10 は、以下の課題に直面する：ROI 60 の位置の表示は、原則としてエンコーダ 10 にとって問題ではない。そのために、エンコーダ 10 は、データストリーム 22 の中で ROI 60 の位置を容易に示すことができる。しかしながら、容易にアクセスできるこの情報を示すために、図 15 のエンコーダ 10 はアクセスユニットのペイロード・パケットの間に ROI パケットの分散を使用し、それにより、エンコーダ 10 は、サブポーション 24 および / またはサブポーション 24 がパケット化され、ROI 60 の空間的に外側および空間的に内側にあるペイロード・パケットの数に自由に、継続的に、現在の画像 18 の区分を選択する。分散型の ROI パケットを用いて、いかなるネットワークエンティティも、ROI に属するペイロード・パケットを容易に確認することができる。一方では、これらの ROI パケットのための除去可能なパケット・タイプを使用する場合、それはいかなるネットワークエンティティによっても容易に無視されることができる。

10

20

30

40

50

【0152】

図 15 は、アクセスユニット 30 のペイロード・パケット 32 の間に ROI パケット 64 を分散させるための実施例を示す。ROI パケット 64 は、ビデオ・データストリーム 22 内で ROI 60 に関連する、すなわち符号化する符号化データが含まれるところを示す。ROI パケット 64 が ROI 60 の位置を示す方法は、多種多様な方法で行うことができる。たとえば、ROI パケット 64 の純粋な存在 / 発生は、シーケンス 34 の順次的な順序において続く、すなわち前に置かれたペイロード・パケットに属する、以下のペイロード・パケット 32 の 1 つ以上の範囲内で ROI 60 に関する符号化データの編入を示す。あるいは、ROI パケット 64 内部の構文要素は、1 つ以上の次のペイロード・パケット 32 が ROI 60 に関連するかどうか、すなわち少なくとも部分的に符号化するかどうかを示す。変化の多さも、それぞれの ROI パケット 64 の「スコープ」に関して可能なバリエーション、すなわち 1 つの ROI パケット 64 によって前に置かれる前に置かれたペイロード・パケットの数から生じる。たとえば、1 つの ROI パケット内で ROI 60 に関するいくらか符号化データの編入または非編入の示唆は、次の ROI パケット 64 の発生までシーケンス 34 の順次的な順序に続き、または、単にすぐに続くペイロード・パケット 32、すなわちシーケンス 34 の順次的な順序におけるそれぞれの ROI パケット 64 に続くペイロード・パケット 32 に関連する。図 15 において、グラフ 66 は見本として、次の ROI パケット 64 の発生までそれぞれの ROI パケット 64 のダウンストリームを発生させる全てのペイロード・パケット 32、またはパケットのシーケンス 34 に沿って早く発生するものは何でも現在のアクセスユニット 30 の終わりに関連して、ROI パケット 64 が ROI の関連、すなわち ROI 60 に関するいかなる符号化されたデータの編入、または ROI の非関連、すなわち ROI 60 に関連するいかなる符号化データの欠如を示すケースを例示する。特に、図 15 は ROI パケット 64 が構文要素を中で備えているケースを例示し、それはいずれにせよパケット・シーケンス 34 の順次的な順序において続いているペイロード・パケット 32 が中の ROI 60 に関する符号化されたデータを有するか否かを示す。このような実施例も、以下に記載する。しかしながら、他の可能性は、今述べたように、各 ROI パケット 64 が、それぞれの ROI パケット 64 の「スコープ」に属するペイロード・パケット 32 がデータ内部に関連する ROI 60、すなわち ROI 60 に関連するデータを有しているパケット・シーケンス 34 に存在することにより示すことである。更に詳細に以下に記載する実施例によれば、ROI パケット 64 は、その「スコープ」に属しているペイロード・パケット 32 に符号化される ROI 60 の部分の位置さえ示す。

【0153】

たとえば、パケットのシーケンス34のROI関連部分をパケット・シーケンス34の他の部分より高い優先度で扱うためにROIパケット64を用いて理解されるように、ビデオ・データストリーム22を受信しているいかなるネットワークエンティティ68もROI関連の指し示すことを利用することができる。あるいは、ネットワークエンティティ68は、例えば、ビデオ・データストリーム22の伝送に関する他の作業を遂行するために、ROI関連情報を使用することができる。ネットワークエンティティ68は、例えば、ビデオ・データストリーム22、28を介して伝送されるようなビデオ内容60の復号化および再生のためのMANEまたはデコーダである。換言すれば、ネットワークの素材68は、ビデオ・データストリームに関連している伝送作業を決定するために、ROIパケットの識別の結果を使用することができる。伝送作業は、欠陥パケットに関して再伝送要請を含むことができる。ネットワークエンティティ68は、増加した優先度を有する関心領域70を扱って、より高い優先権をROIパケット72およびそれらの付随するペイロード・パケット、すなわちそれによって前に置かれるものに割り当てるように構成されることができ、それは、ROIを覆わないように信号を送られるROIパケットおよびそれらの付随するペイロード・パケットと比較して、関心領域を覆うように信号を送られる。それに対して割り当てられる下の優先度を有するペイロード・パケットのいくつかの転送を要請する前に、ネットワークエンティティ68は、それに対して割り当てられるより高い優先度を有するペイロード・パケットの転送を最初に要請することができる。

10

【0154】

図15の実施例は、図12～14に関して以前に記載されている実施例と、容易に結合されることができる。たとえば、上述のROIパケット64は、そこにおいて、含まれる特定のタイプのSEIメッセージ、すなわちROI SEIメッセージを有するSEIパケットであってもよい。すなわち、SEIパケットは、例えば、タイミング制御パケットおよび同時にROIパケットであってもよく、すなわち、それぞれのSEIパケットが両方のタイミングから成る場合、ROI表示情報と同様に情報を制御する。あるいは、SEIパケットは、他の1つであるよりもむしろ、タイミング制御パケットおよびROIパケットのうちの一つでもよく、ROIパケットかタイミング制御パケットでなくてもよい。

20

【0155】

図16において図示した実施例に従って、アクセスユニットのペイロード・パケット間へのパケットの分散は、現在のアクセスユニット30が関連している、現在の画像18のタイルである、ビデオ・データストリーム22を取り扱うネットワークエンティティ68に容易にアクセスできる方法で示すように用いられ、それぞれのパケットが接頭辞として役立つペイロード・パケット32のいくつかに符号化されるいくつかのサブポジションによって覆われる。図16において、例えば、見本としてここでは現在の画像18の4つの四半部によって形成されて、現在の画像18は4つのタイル70に再分割されることを示す。また、例えば、パケットのシーケンス34に分散されるVPSまたはSPSパケットで、タイル70への現在の画像18の細別は、例えば、画像のシーケンスから成るユニットのビデオ・データストリームの中で信号を送られることができる。以下において更に詳細に説明するように、現在の画像18のタイル再分割はタイルの列および行における画像18の規則的な再分割でもよい。タイルの列の幅および行の高さと同様に列の数および行の数は変化していてもよい。特に、タイルの列/行の幅と高さは、異なる行および異なる列でそれぞれ異なってもよい。図16は、加えて、サブポジション24が画像18のスライスである実施例を示す。スライス24は、画像18を再分割する。以下において更に詳細に概説されるように、スライス24への画像の18の再分割は、各スライス24が1つのタイル70の範囲内に完全に含まれることができるか、または完全に2つ以上のタイル70をカバーすることができるという制約に従属される。図16は、画像18が5つのスライス24に再分割されるケースを例示する。上述した復号化順序におけるこれらの最初の4つのスライス24は最初の2つのタイル70をカバーするが、その一方で、5番目のスライスは完全に第3および第4のタイル70をカバーする。さらに、図16は、各

30

40

50

スライス 24 がそれぞれのペイロード・パケット 32 に個々に符号化されるケースを例示する。さらに、図 16 は、見本として、各ペイロード・パケット 32 が先行するタイル識別パケット 72 によって前に置かれるケースを例示する。各タイル識別パケット 72 は、次に、このペイロード・パケット 32 に符号化されるサブポーション 24 がタイル 70 のいずれを覆うかについてその直ちに続くペイロード・パケット 32 のために指示する。したがって、現在の画像 18 に関するアクセスユニット 30 の範囲内の最初の 2 つのタイル識別パケット 72 が第 1 のタイルを示すと共に、第 3 および第 4 のタイル識別パケット 72 は画像 18 の第 2 のタイル 70 を示し、第 5 のタイル識別パケット 72 は第 3 および第 4 のタイル 70 を示す。図 16 の実施例に関して、例えば、同じバリエーションは、図 15 に関して上述したように可能である。すなわち、タイル識別パケット 72 の「スコープ」は、例えば、単に次のタイル識別パケットの発生まで、第 1 の直ちに続くペイロード・パケット 32 または直ちに続くペイロード・パケット 32 を含むことができるだけである。

10

【0156】

タイルに関して、タイル境界全体に、空間予測または前後関係選択が起こらないように、エンコーダ 10 は各タイル 70 を符号化するように構成されることができる。エンコーダ 10 は、例えば、並行してタイル 70 を符号化することができる。同様に、ネットワークエンティティ 68 のようないかなるデコーダも、並行してタイル 70 を復号化することができる。

20

【0157】

ネットワークエンティティ 68 はエンコーダ 10 およびデコーダとの間における M A N E またはデコーダまたはいくつかの他の装置でもよく、特定の伝送作業を決定するためにタイル識別パケット 72 によって伝達される情報を使用するように構成することができる。たとえば、ネットワークエンティティ 68 は、より高い優先度でビデオ 16 の現在の画像 18 の特定のタイルを扱うことができ、すなわち以前にこのようなタイルに関するかまたはより安全な F E C 保護等を使用するとして示されるそれぞれのペイロード・パケットを進めることができる。換言すれば、ネットワークエンティティ 68 は、ビデオ・データストリームに関係している伝送作業を決定するために、識別の結果を使用することができる。伝送作業は欠陥状態において - すなわち、もしあれば、ビデオのいかなる F E C 保護も上回ることで、受け取られるパケットに関して再伝送要請を含むことができる。ネットワークエンティティは、例えば、異なる優先度で異なるタイル 70 を扱うことができる。この目的で、ネットワークエンティティは識別パケット 72 よびそれらのペイロード・パケット 72 により高い優先度を割り当てることができる、すなわち、低い優先度に関連するタイル識別パケット 72 およびそれらのペイロード・パケットよりも、前に置かれそれにより高い優先度に関連する。それに対して割り当てられる低い優先度を有するペイロード・パケットのいかなる転送を要請する前に、ネットワークエンティティ 68 は、例えば、それに対して割り当てられるより高い優先度を有するペイロード・パケットの転送を最初に要請することができる。

30

【0158】

今までに記載されている実施例は、以下に説明するように、本出願の前置き部分にて説明したような H E V C フレームワークに組み込まれることができる。

40

【0159】

特に、S E I メッセージは、サブピクチャ C P B / H R D ケースの復号化ユニットのスライスに割り当てられることができる。すなわち、バッファリング期間およびタイミング S E I メッセージは、復号化ユニットのスライスを含んでいる N A L ユニットに割り当てられることができる。これは、直接復号化装置の 1 つ以上のスライス / V C L N A L ユニットに先行することができる非 V C L N A L ユニットである新しい N A L ユニットタイプによって達成されることができる。この新規な N A L ユニットは、スライス接頭辞 N A L ユニットと呼ばれることができる。図 17 は、シーケンスおよびストリームの端部のためのいかなる一時的な N A L ユニットも省略したアクセスユニットの構造を例示する。

50

【0160】

図17によれば、アクセスユニット30は、以下の通りに構成される：パケットのシーケンス34のパケットの順次的な順序で、アクセスユニット30は、特別なタイプのパケット、すなわち、アクセスユニットデリミタ80の発生から始めることができる。それから、全てのアクセスユニットに関するSEIパケット・タイプの1つ以上のSEIパケット82は、アクセスユニット30内で続くことができる。両方のパケット・タイプ80および82は、任意である。すなわち、このタイプのパケットは、アクセスユニット30内で発生することができない。それから、復号化ユニット38のシーケンスは続く。各復号化ユニット38は、例えばタイミング制御情報を含み、または図15または16の実施例によれば、ROI情報またはタイル情報または、より一般的には、それぞれのサブピクチャSEIメッセージ86を含む、スライス接頭辞NALユニット84で任意に始まる。それから、それぞれのペイロード・パケットまたはVCL NALユニットの実際のスライス・データ88は、88に示すように続く。このように、各復号化ユニット38は、それぞれのスライス・データNALユニット88が続くスライス接頭辞NALユニット84のシーケンスを含む。スライス接頭辞NALユニットを回避する図17のバイパス矢印90は、現在のアクセスユニット30の非復号化ユニット再分割の場合には、スライス接頭辞NALユニット84がないことを示す。

10

【0161】

すでに上述したように、スライス接頭辞で信号を送ったすべての情報および付随するサブピクチャSEIメッセージは、スライス接頭辞NALユニットにおいて伝えられるフラグに応じて、アクセスユニットの、または、第2の接頭辞NALユニットの発生までのすべてのVCL NALユニットのために、または、復号化順序において以下のVCL - NALユニットのために効果的であってもよい。

20

【0162】

スライス接頭辞で信号を送られる情報が有効であるスライスVCL NALユニットは、以下において接頭辞スライスと呼ばれる。前に置かれる単一のスライスと関連する接頭辞スライスは、完全な復号化ユニットを必ずしも構成する必要はないが、その一部であり得る。しかしながら、単一のスライス接頭辞は並列の復号化ユニット（サブピクチャ）に有効であることができず、復号化ユニットの始まりはスライス接頭辞で信号を送られる。信号を送るための手段がスライス接頭辞構文（以下に示す「単純な構文」/バージョンのような）によって与えられない場合、スライス接頭辞NALユニットの発生は復号化ユニットの始まりの信号を送る。特定のSEIメッセージ（以下の構文説明のペイロードタイプを介して確認される）だけはスライス接頭辞NALユニットの範囲内でサブピクチャ・レベルで送られることができ、その一方で、いくらかのSEIメッセージはサブピクチャ・レベルでスライス接頭辞NALユニットにおいて、または、アクセスユニットレベルで通常のSEIメッセージとして送られることができる。

30

【0163】

図16に関して上述したように、さらに、または、あるいは、タイルID SEIメッセージ/タイルIDシグナリングは、高水準構文において理解されることができる。HEVCの以前のデザインにおいて、スライス・ヘッダ/スライス・データは、それぞれのスライスに含まれるタイルのための識別子を含んだ。例えば、スライス・データ意味論は示す：

40

【0164】

$tile_idx_minus_1$ は、ラスト・スキャン順序のTile IDを規定する。画像の第1のタイルは、0のTile IDを有する。 $tile_idx_minus_1$ の値は、0から $(num_tile_columns_minus_1 + 1) * (num_tile_rows_minus_1 + 1) - 1$ の範囲内にある。

【0165】

$tiles_or_entropy_coding_sync_idc$ が1に等しい場合、このIDがスライスアドレスから、および画像パラメータ・セットにおいて信号を送

50

られるようなスライスディメンジョンから容易に引き出されるため、このパラメータは役立つとは認められない。

【0166】

タイルIDが暗に復号化プロセスの中で引き出されることができるにもかかわらず、異なるタイルがプレイバックに対して異なる優先度を有する（これらのタイルが典型的には会話形式を用いる話者を含む関心領域を形成する）ビデオ会議シナリオが別のタイルより高い優先度をもっているように、アプリケーションレイヤーにおけるこのパラメータの知識は異なる使用の場合に重要である。複数のタイルの伝送のネットワーク・パケットを失う場合には、関心領域を表しているタイルを含んでいるそれらのネットワーク・パケットは、いかなる優先度順序もない再送信タイルの場合よりも高いレシーバ端末に経験品質を保つために、より高い優先度で再送信されることができる。寸法およびそれらの位置が公知である場合、例えばテレビ会議シナリオのように、異なるスクリーンに、他の使用ケースはタイルを割り当てることができる。

10

【0167】

このようなアプリケーションレイヤーが伝送シナリオにおいて特定の優先度を有するタイルを扱うことができるために、`tile_id`は、サブピクチャまたはスライスに特有のSEIメッセージとして、または、タイルの1つ以上のNALユニットの前の特別なNALユニットにおいて、または、タイルに属しているNALユニットの特別なヘッダ部分において設けられることができる。

20

【0168】

図15に関して上述したように、関心SEIメッセージの領域は、さらに、または、代わりに設けられていることもできる。このようなSEIメッセージは、関心領域(ROI)の信号を送ることを可能にし、特に、特定の`tile_id`/タイルが属するROIのシグナリングにおいて感心領域(ROI)のシグナリングを可能にすることができる。メッセージは、関心領域IDに加えて関心領域の優先度を与えることを可能にすることができる。

【0169】

図18は、関心領域シグナリングのタイルの使用を例示する。

【0170】

上で記載されていたことに加えて、スライス・ヘッダ・シグナリングが実行されることができる。スライス接頭辞NALユニットは、以下の従属するスライス、すなわち、それぞれのスライス接頭辞によって前に置かれるスライスのためのスライス・ヘッダを含むこともできる。スライス・ヘッダがスライス接頭辞NALユニットにおいて提供されるだけである場合、それぞれの従属するスライスを含んでいるNALユニットのNALユニットタイプによって、または次のスライス・データがランダムアクセスポイントとして働くスライスタイプに属するかどうかの信号を送るスライス接頭辞におけるフラグによって、実際のスライスタイプが引き出される。

30

【0171】

さらに、スライス接頭辞NALユニットは、例えばサブピクチャ・タイミングまたはタイル識別子のような任意の情報を伝達するために、スライスまたはサブピクチャに特定のSEIメッセージを持たせることができる。任意のサブピクチャの特定のメッセージ発信は、本出願の紹介の部分に記載されているHEVCの明細書においてサポートされていないが、特定のアプリケーションのために重要である。

40

【0172】

次に、スライスの接頭辞付加の上述の概念を実行するための可能な構文が説明される。特に、基本として本出願の明細書の紹介部分で概説されるHEVC状況を使用するとき、どの変化がスライス・レベルに十分でありえるかが説明される。

【0173】

特に、以下において、可能なスライス接頭辞構文の2つのバージョンが与えられ、1つはSEIメッセージ発信だけの機能をもつものであり、1つは次のスライスのためにスラ

50

イス・ヘッダの一部の信号を送る拡張した機能をもつものである。第1の単純な構文/バージョン1は、図19に示される。

【0174】

予備的な注釈として、図19は、このように、図11～16に関して上記の実施例のいずれかを実行するためのありうる実行を示す。その中で示される分散型のバケットは図19に示すように解釈されることができ、以下にいて、これは特定の実施例によって更に詳細に記載されている。

【0175】

SEIメッセージ概念から離れた `tile_id` シグナリング、復号化ユニットスタート識別子、スライス接頭辞IDおよびスライス・ヘッダデータを含む拡張した構文/バージョン2は、図20の表において与えられる。

10

【0176】

意味論は、以下の通りに定められることができる：

1の値を有する `rap_flag` は、スライス接頭辞を含むアクセスユニットがRAP画像であることを示す。0の値を有する `rap_flag` はスライス接頭辞を含まないアクセスユニットがRAP画像でないことを示す。

【0177】

`decoding_unit_start_flag` は、アクセスユニットの範囲内で復号化ユニットの始まりを示し、したがって、次にくるスライスは、アクセスユニットの終端まで、または同じ復号化ユニットに属する他の復号化ユニットの始まりまで続く。

20

【0178】

0の値を有する `single_slice_flag` は、接頭辞スライスNALユニットおよび付随するサブピクチャSEIメッセージの範囲内で与えられる情報が次のアクセスユニット、他のスライス接頭辞の発生または他の完全なスライス・ヘッダの始まりまですべての以下のVCL-NALユニットにあてはまることを示す。値1を有する `single_slice_flag` は、スライス接頭辞NALユニットおよび付随するサブピクチャSEIメッセージに示されるすべての情報が順序を復号化する際の次のVCL-NALユニットのためにだけ有効なことを示す。

【0179】

`tile_idc` は、以下のスライスに存在するタイルの量を示す。0に等しい `tile_idc` は、タイルが以下のスライスにおいて用いられないことを示す。1に等しい `tile_idc` は、単一のタイルが以下のスライスにおいて用いられることを示し、したがってそのタイル識別子は信号を送られる。2の値を有する `tile_idc` は、複数のタイルが以下のスライスの範囲内で用いられることを示し、したがって、タイルの数および第1のタイル識別子は信号を送られる。

30

【0180】

`prefix_slice_header_data_present_flag` は、復号化順序において続くスライスに対応するスライス・ヘッダのデータが与えられたスライス接頭辞で信号を送られることを示す。

【0181】

`slice_header_data()` は、テキストの後期に定められる。それは関連したスライス・ヘッダ情報を含み、`dependent_slice_flag` が1に等しく設定された場合、それはスライス・ヘッダによってカバーされない。

40

【0182】

スライス・ヘッダおよび実際のスライス・データを切り離すことがヘッダおよびスライス・データのより柔軟な伝送方式を可能にすることに注意されたい。

【0183】

`num_tiles_in_prefixed_slices_minus1` は、1なしで次の復号化ユニットマイナス1において使用されるタイルの数を示す。

【0184】

50

`first_title_id_in_prefixed_slices`は、次の復号化ユニットにおける第1のタイトルのタイトル識別子を示す。

【0185】

スライス接頭辞の単純な構文/バージョン1のために、ないときには、以下の構文要素は、以下の通りにデフォルト値に設定されることができる：

- `decoding_unit_start`は1に等しい、すなわち、スライス接頭辞は、常に復号化ユニットのスタートを示す。

- `single_slice_flag`は0に等しい、すなわち、スライス接頭辞は、復号化ユニットのすべてのスライスにあてはまる。

【0186】

スライス接頭辞NALユニットは、24のNALユニットタイプを有し、NALユニットは図21に従って延長されるテーブルを概観する。

【0187】

すなわち、簡単に図19~21を要約すると、ここで見られる構文の詳細は、特定のケット・タイプが上記の確認された分散型のケットに起因している、ここでは見本としてNALユニットタイプ24を公開する。さらに、特に、図20の構文例は、分散型のケットの「スコープ」、これらの分散型のケット自身の範囲内でそれぞれの構文要素によって制御されるスイッチングメカニズム、ここでは`single_slice_flag`がこのスコープを制御するために用いられること、すなわち、それぞれ、このスコープの定義のための異なる選択肢の間のスイッチングをすることの上述の選択肢が明らかにされる。さらに、それぞれの分散型のケットの分散型のケットの「スコープ」に属しているケットに含まれるスライス24のための共通のスライス・ヘッダデータを含むという点で、図1~16の上述した実施例が延長されることができると明らかにされた。すなわち、共通のスライス・ヘッダデータがそれぞれの分散型のケットの範囲内で含まれるかどうかを示すこれらの分散型のケットの範囲内でそれぞれのフラグによって制御される機構があることができる。

【0188】

もちろん、HEVCの現在バージョンにおいて特定されるように、スライス・ヘッダデータのどの一部がスライス・ヘッダ接頭辞に移されるかについてちょうど提示される概念がスライス・ヘッダに変化を必要とする。図22の表は、このようなスライス・ヘッダのための可能な構文を示し、現在のバージョンに従ってスライス・ヘッダに存在する確かな構文要素が`slice_header_data()`と呼ばれる下の階層にシフトされる。スライス・ヘッダおよびスライス・ヘッダデータのこの構文は、拡張したスライス・ヘッダ接頭辞NALユニット概念が使われるオプションに適用されるだけである。

【0189】

図22において、`slice_header_data_present_flag`は、現在のスライスのためのスライス・ヘッダデータがアクセスユニット、すなわち最も最近発生しているスライス接頭辞NALユニットの最後のスライス接頭辞NALユニットにおいて信号を送られる値から予測されることを示す。

【0190】

スライス・ヘッダから離れたすべての構文要素は、図23の表において与えられるように、構文要素スライス・ヘッダデータによって信号を送られる。

【0191】

すなわち、図22および図23の概念を図12~16の実施例に移転して、その中で記載されている分散型のケットは、これらの分散型のケットに組み込む概念、ペイロード・ケットに符号化されるスライス・ヘッダ構文の一部(サブポーション)24、すなわちVCL NALユニットによって延長されることができる。編入は、任意でもよい。すなわち、分散型のケットのそれぞれの構文要素は、このようなスライス・ヘッダ構文がそれぞれの分散型のケットに含まれるかどうかを示すことができる。組み込まれる場合、それぞれの分散型のケットに組み込まれるそれぞれのスライス・ヘッダデ

10

20

30

40

50

ータはそれぞれの分散型のパケットの「スコープ」に属しているパケットに含まれるすべてのスライスに適用されることができる。分散型のパケットに含まれるスライス・ヘッダデータがこの分散型のパケットのスコープに属するペイロード・パケットのいずれかに符号化されるスライスによって採用されるかどうかは、例えば図22の `slice_header_data_present_flag` によって信号を送られることができる。この計測によって、それぞれの分散型のパケットの「スコープ」に属しているパケットに符号化されるスライスのスライス・ヘッダが小型化され、したがって、スライス内のちょうど言及したフラグを用いて、上述の図12～16に見られるネットワークエンティティのようなスライス・ヘッダおよびビデオ・データストリームを受信するいずれかのデコーダは、スライスのスライス・ヘッダ内のちょうど言及したフラグに応答し、分散型のパケットに組み込まれたスライス・ヘッダデータを、スライス接頭辞へのスライス・ヘッダデータの移動の信号を送るスライスの範囲内のそれぞれのフラグ、すなわちそれぞれの分散型のパケットの場合、この分散型のパケットのスコープに属するペイロード・パケットに符号化されるスライスのスライス・ヘッダにコピーするようにしている。

10

20

30

40

50

【0192】

さらに、図12～16の実施例を実施するための構文例を続行して、SEIメッセージ構文が、図24に示すようにすることができる。スライスまたはサブピクチャSEIメッセージタイプを導入するために、SEIペイロード構文は図25の表に示されるように構成されることができる。180～184の範囲の `payloadType` を有するSEIメッセージだけは、スライス接頭辞NALユニットの範囲内で、サブピクチャ・レベルだけ独占的に送られることができる。さらに、140に等しい `payloadType` を有する関心領域SEIメッセージは、サブピクチャ・レベル上のスライス接頭辞NALユニットまたはアクセスユニットレベル上の定期的なSEIメッセージのどちらでも送信されることができる。

【0193】

すなわち、図12～16に関して上述された実施例上に図25および24に示される詳細を移す中で、図12～16のこれらの実施例に示される分散型のパケットは、スライス接頭辞NALユニットの範囲内で、例えば、各SEIメッセージの始めに `payloadType` によって信号が送られる特定のタイプのSEIメッセージを含む特定のNALユニットタイプ、例えば24、を有するスライス接頭辞NALユニットを用いて実現されることができる。現在記載されている特定の構文例において、`payloadType = 180` および `payloadType = 181` は図11～14の実施例に従うタイミング制御パケットという結果になり、その一方で、`payloadType = 140` は図15の実施例に従うROIパケットという結果になり、`payloadType = 182` は図16の実施例に従うタイル識別パケットという結果になる。以下において本願明細書に記載されている特定の構文例は、1つのまたはちょうど言及された `payloadType` オプションのサブセットを含むことができる。これを越えて、図25は、図11～16の上述の実施例のいずれかが各々と結合されることができることを明らかにする。さらに、図25は図12～16の実施例またはそれらの組み合わせも分散型のパケットによって延長されることができることを明らかにし、その後、`payloadType = 184` によって説明される。すでに上述したように、`payloadType = 183` に関して後述する拡張はいかなる分散型のパケットもそのスコープに属しているいかなるペイロード・パケットにも符号化されるスライスのスライス・ヘッダのための共通のスライス・ヘッダデータを組み込むことができたという可能性において終わる。

【0194】

以下の表は、スライスまたはサブピクチャ・レベルに使われることができるSEIメッセージを定める。サブピクチャおよびアクセスユニットレベルに使われることができる関心のあるSEIメッセージの領域も示される。

【0195】

NALユニットタイプ24のスライス接頭辞NALユニットが、そこに含まれるSEI

メッセージタイプ180を有するときにはいつでも、図26は、例えば、発生しているサブピクチャ・バッファSEIメッセージのための実施例を示し、このように、タイミング制御パケットを形成する。

【0196】

意味論は、以下の通りに定められることができる：

`seq_parameter_set_id`は、シーケンスHRD属性を含むシーケンス・パラメータ・セットを特定する。`seq_parameter_set_id`の値は、バッファリングピリオドSEIメッセージと関連した主要な符号化画像によって参照されてセットされる画像パラメータの`seq_parameter_set_id`の値に等しい。`seq_parameter_set_id`の値は範囲は0から31においてあり、次を含んでいる。

10

【0197】

`initial_cpb_removal_delay[SchedSelIdx]`そして、`initial_alt_cpb_removal_delay[SchedSelIdx]`は、復号化ユニット(サブピクチャ)の`SchedSelIdx`のCPBのための最初のCPB除去遅延を特定する。構文要素は、`initial_cpb_removal_delay_length_minus1+1`によって与えられるビットの長さを有し、90kHzクロックを単位にする。構文要素の値は、0ではなく、 $90000 * (CpbSize[SchedSelIdx] \div BitRate[SchedSelIdx])$ 、90kHzクロックユニットのCPBサイズの時間相当を上回らない。

20

【0198】

全ての符号化ビデオ・シーケンスの上に、`initial_cpb_removal_delay[SchedSelIdx]`の合計および復号化ユニット(サブピクチャ)につき`initial_cpb_removal_delay_offset[SchedSelIdx]`は`SchedSelIdx`の値ごとに一定であり、そして、`initial_alt_cpb_removal_delay[SchedSelIdx]`および`initial_alt_cpb_removal_delay_offset[SchedSelIdx]`の合計は`SchedSelIdx`の値ごとに一定である。

30

【0199】

図27は、同様にサブピクチャ・タイミングSEIメッセージのための例を示し、意味論は、以下の通りに記載されることができ：

【0200】

もしあるならば、同じ復号化ユニット(サブピクチャ)における先行するアクセスユニットの最近のサブピクチャ・バッファリングピリオドSEIメッセージと関連し、さもなければ、サブピクチャ・タイミングSEIメッセージと関連する復号化ユニット(サブピクチャ)データをバッファから除去する前に、先行するアクセスユニットにおける最近のバッファリングピリオドSEIメッセージと関連する復号化ユニット(サブピクチャ)のCPBからの除去の後、どれくらいのクロックが待つために刻んだかを特定する。この値は、HSS(仮定的な`StreamScheduler[2]0`)のためのCPBへの復号化ユニット(サブピクチャ)データの到着の最も早い可能な時間を計算するためにも用いる。構文要素は、ビットの長さが`cpb_removal_delay_length_minus1+1`によって与えられる固定長符号である。`cpb_removal_delay`は、逆に $modulo 2(cpb_removal_delay_length_minus1+1)$ カウンタの剰余である。

40

【0201】

`du_dpb_output_delay`は、復号化ユニット(サブピクチャ)のDPB出力時間を計算するために用いられる。それは、画像の復号化ユニット(サブピクチャ)がDPBから出力される前にCPBから復号化された復号化ユニット(サブピクチャ)の除去の後、どれくらいのクロックが待つために刻むかについて特定する。

50

【0202】

これがサブピクチャのアップデートを可能にすることに注意されたい。このようなシナリオにおいて、アップデートしていない復号化ユニットは、最後の復号化画像の中で不変のままであり、それらは、見えるままである。

【0203】

図26および27を要約して、そこに含まれる具体的な詳細を図12~14の実施例に移行すると、復号化ユニットのためのデコーダバッファ検索時間が、すなわち付加的に復号化バッファ検索時間と関連して、差別的に符号化方法の関連するタイミング制御パッケージにおいて信号を送られることができるということが出来る。すなわち、特定の復号化ユニットのための復号器バッファ検索時間を得るために、ビデオ・データストリームを受信するデコーダは、特定の復号化ユニットの前に置かれるタイミング制御パッケージから得られるデコーダ検索時間を直ちに先行する復号化ユニットのデコーダ検索時間、すなわち特定の復号化ユニットに先行するものおよび復号化ユニットが続くようなこの方法の手に加える。各々いくつかの画像またはそのパーツの符号化ビデオ・シーケンスの始まりで、タイミング制御パッケージは、さらに、または、代わりに、先行する復号化ユニットのデコーダバッファ検索時間と差別的に比べたよりも完全に符号化されたデコーダバッファ検索時間を含む。

10

【0204】

図28は、サブピクチャスライス情報SEIメッセージがどのように見えるかを示す。意味論は、以下の通りに定められることができる：

20

【0205】

1の値を有するslice_header_data_flagは、スライス・ヘッダデータがSEIメッセージに存在することを示す。SEIにおいて与えられているスライス・ヘッダデータは、アクセス装置、他のSEIメッセージのスライス・データの発生、スライスNAL装置またはスライス接頭辞NAL装置の端まで復号化順序を受け継ぐすべてのスライスに有効である。

【0206】

図29は、サブピクチャ・タイル情報SEIメッセージのための実施例を示す、意味論は、以下の通りに定められることができる：

【0207】

tile_priorityは、復号化順序を引き継ぐ前に置かれたスライスのすべてのタイルの優先度を示す。tile_priorityの値はすべてこみで範囲0~7の範囲にあり、7は最高優先度を示す。

30

【0208】

1の値を有するmultiple_tiles_in_prefixed_slices_flagは、復号化順序を引き継ぐために接頭辞スライスにおける1以上のタイルを有することを示す。0の値を有するmultiple_tiles_in_prefixed_slices_flagは、次の接頭辞スライスが1つのタイルだけを含むことを示す。

【0209】

num_tiles_in_prefixed_slices_minus1は、復号化順序を受け継ぐ接頭辞スライスのタイルの数を示す。

40

【0210】

first_tile_id_in_prefixed_slicesは、復号化順序を受け継ぐ接頭辞スライスの第1のタイルのtile_idを示す。

【0211】

すなわち、図16の実施例は、図16において言及されるタイル識別パッケージを実現するための図29の構文を使用して実施されることが出来る。そこに示されるように、特定のフラグ、ここでは、multiple_tiles_in_prefixed_slices_flagは、単に1つのタイルだけか複数のタイルがそれぞれの分散型のタイル

50

識別パケットの範囲に属しているペイロード・パケットのいずれかに符号化される現在の画像 18 のサブポーションによってカバーされるかどうか分散型のタイル識別パケットの範囲内で示すために用いてもよい。フラグが複数のタイルを覆うことの信号を送る場合、更なる構文要素がそれぞれの分散型のパケット、ここでは見本として、それぞれの分散型のタイル識別パケットの範囲に属するいかなるペイロード・パケットのいかなるサブポーションによって終われるタイルの数を示す `num__tiles__in__prefixed__slices__minus 1` に含まれる。最後に、更なる構文要素、ここでは見本として、`first__tile__id__in__prefixed__slices` は、現在の分散型のタイル識別パケットによって示されるタイルの数の中のタイルの ID を示し、それは復号化順序に従って第 1 のものである。図 29 の構文を図 16 の実施例に転移して、第 5 のペイロード・パケット 32 を前に置いているタイル識別パケット 72 は、例えば、1 にセットされている `multiple__tiles__in__prefixed__slices__flag`、1 にセットされている `num__tiles__in__prefixed__slices__minus 1` を有するすべての 3 つのちょうど述べられた構文要素を備え、それによって、2 つのタイルが現在のスコープに属する 2 つのタイル、および 3 にセットされる `first__tile__id__in__prefixed__slices` に属することを示し、そして、現在のタイル識別パケット 72 のスコープに属している復号化順序のタイルの進行が 3 つ目のタイル (`tile__id = 2` を有する) で始まることを示す。

10

【0212】

図 29 も、タイル識別パケット 72 がおそらく `tile__priority`、すなわちそのスコープに属しているタイルの優先度を示すこともできることを明らかにする。ROI 態様と同様に、ネットワークエンティティ 68 は、例えば特定のペイロード・パケットの転送の要請のような伝送作業を制御するためにこのような優先度情報を使用することができる。

20

【0213】

図 30 は、サブピクチャ・タイル範囲情報 SEI メッセージのための構文例を示し、意味論は、以下として定義されることができる：

【0214】

1 の値を有する `multiple__tiles__in__prefixed__slices__flag` は、復号化順序を受け継ぐために、接頭辞スライスに 1 つ以上のタイルがあることを示す。0 の値を有する `multiple__tiles__in__prefixed__slices__flag` は、以下の接頭辞スライスが 1 つのタイルだけを含むことを示す。

30

【0215】

`num__tiles__in__prefixed__slices__minus 1` は、復号化順序を受け継ぐための接頭辞スライスのタイルの数を示す。

【0216】

`tile__horz__start [i]` は、画像の範囲内でピクセルの *i* 番目のタイルの水平方向の始まりを示す。

【0217】

`tile__width [i]` は、画像の範囲内でピクセルの *i* 番目のタイルの幅を示す。

40

【0218】

`tile__vert__start [i]` は、画像の範囲内でピクセルの *i* 番目のタイルの水平方向の始まりを示す。

【0219】

`tile__height [i]` は、画像の範囲内でピクセルの *i* 番目のタイルの高さを示す。

【0220】

タイル範囲 SEI メッセージは表示操作、例えばタイルを複数の画面表示シナリオのス

50

クリーンに割り当てるために用いられることに注意されたい。

【0221】

図30は、このように、それぞれのタイル識別パケットのスコープに属しているタイルがそれらのタイルIDよりむしろ現在の画像18の範囲内でそれらの場所によって示されるという点で、図16のタイル識別パケットに関する図29の実施構文例が変化に富んでもよいことを明らかにする。すなわち、それぞれの分散型のタイル識別パケットのスコープに属しているペイロード・パケットのいずれかに符号化されるそれぞれのサブポーションによってカバーされる復号化順序における第1のタイルのタイルIDの信号を送るよりはむしろ、現在のタイル識別パケットに属しているタイルごとに、その位置は、例えば各タイル*i*の上左コーナー位置、ここでは見本として`tile__horz__start`および`tile__vert__start`、およびタイル*i*の幅および高さ、ここでは見本として`tile__width`および`tile__height`によって信号を送られることができる。

10

【0222】

関心領域SEIメッセージのための構文例は図31に示される。さらに正確であるために、図32は、第1の変形を示す。特に、関心領域SEIメッセージが、例えば、1つ以上の関心領域の信号を送るために、アクセスユニットレベルまたはサブピクチャ・レベルで使われることができる。図32の第1の変形によれば、複数のROIが現在のスコープの中にある場合、個々のROIは、1つのROI SEIメッセージの範囲内のそれぞれのROIパケットのスコープのすべてのROIの信号を送ることよりむしろ、ROI SEIメッセージごとに一度信号を送られる。

20

【0223】

図31によれば、関心領域SEIメッセージは、個々に各ROIの信号を送る。意味論は、以下の通りに定められることができる：

【0224】

`roi__id`は、関心領域の識別子を示す。

【0225】

`roi__priority`は、復号化順序を受け継ぐために、SEIメッセージがサブピクチャ・レベルかアクセスユニットレベルのどちらで送られるかによって、接頭辞スライスにおいて関心領域またはすべてのスライスに属するすべてのタイルの優先度を示す。`roi__priority`の値は、包括的に0~7の範囲であり、7は、最高優先度を示す。両方とも、ROI情報SEIメッセージの`roi__priority`およびサブピクチャ・タイル情報SEIメッセージの`tile__priority`が与えられる場合、両方の最も高い値は個々のタイルの優先度にあてはまる。

30

【0226】

`num__tiles__in__roi__minus 1`は、関心領域に属する復号化順序を受け継ぐ接頭辞スライスにおけるタイルの数を示す。

【0227】

`roi__tile__id[i]`は、復号化順序を受け継ぐ接頭辞スライスに対する関心領域に属する*i*番目のタイルの`tile__id`を示す。

40

【0228】

すなわち、図31は、図15で示すようなROIパケットが、それぞれのROIパケットおよびスコープに属するペイロード・パケットが参照する関心領域のIDの信号を送ることができることを示す。任意には、ROI優先度インデックスは、ROI IDとともに信号を送られることができる。しかしながら、両方の構文要素は、任意である。それから、構文要素`num__tiles__in__roi__minus 1`は、それぞれのROI 60に属しているそれぞれのROIパケットのスコープにおけるタイルの数を示すことができる。それから、`roi__tile__id`は、ROI 60に属している*i*番目のタイルのタイル-IDを示す。例えば、画像18が図16に示された方法でタイル70に再分割されることを想像すると、図15のROI 60が画像18の左側の半分に対応する画

50

像に対応することは復号化順序における第1および第3のタイルによって形成される。それから、ROIパケットは図16のアクセスユニット30の第1のペイロード・パケット32の前に配置されることができ、そして、このアクセスユニット30の第4のおよび第5のペイロード・パケット32の間に更なるROIパケットが続く。それから、第1のROIパケットは、`num__tile__in__roi__minus1`が0にセットされるとともに`roi__tile__id[0]`が0にセットされ、(このことにより、復号化順序において第1のタイルに注意を向ける)、第5のペイロード・パケット32の前の第2のROIパケットは、2にセットされている`roi__tile__id[0]`とともに0にセットされている`num__tiles__in__roi__minus1`を有する(このことにより、画像18の左側の底の四半期で復号化順序の第3のタイルを意味している)。

10

【0229】

第2の変形によれば、関心SEIメッセージの領域の構文は、図32に示すようにすることができる。ここで、単一のSEIメッセージのすべてのROIは、信号を送られる。特に、上述のように図31に関するのと同じ構文が使われるが、構文要素、ここでは見本として`num__rois__minus1`により信号を送られる数で、それぞれのROI SEIメッセージまたはROIパケットが参照する多数のROIのそれぞれのROIのための構文要素を乗算する。任意に、更なる構文要素、ここでは見本として、`roi__presentation__on__separate__screen`は、それぞれのROIが別々のスクリーンに示されることに適しているかどうかROIごとに示すことができる。

20

【0230】

意味論的なものは、以下の通りとすることができる：

【0231】

`num__rois__minus1`は、接頭辞スライスのROIまたは復号化順序を受け継ぐ規則的なスライスを示す。

【0232】

`roi__id[i]`は、i番目の関心領域の識別子を示す。

【0233】

`roi__priority[i]`は、復号化順序を受け継ぐために、SEIメッセージがサブピクチャ・レベルかアクセスユニットレベルのどちらで送られるかによって、接頭辞スライスにおいてi番目の関心領域またはすべてのスライスに属するすべてのタイルの優先度を示す。`roi__priority`の値は、包括的に0~7の範囲であり、7は、最高優先度を示す。両方とも、roi情報SEIメッセージの`roi__priority`およびサブピクチャ・タイル情報SEIメッセージの`tile__priority`が与えられる場合、両方の最も高い値は個々のタイルの優先度にあてはまる。

30

【0234】

`num__tiles__in__roi__minus1[i]`は、i番目の関心領域に属する復号化順序を受け継ぐ接頭辞スライスのタイルの数を示す。

【0235】

`roi__tile__id[i][n]`は、復号化順序を受け継ぐ接頭辞スライスに対する関心のi番目の領域に属するn番目のタイルの`tile__id`を示す。

40

【0236】

`roi__presentation__on__separate__screen[i]`は、i番目の`roi__id`に関連した関心領域が別々のスクリーン上のプレゼンテーションに適していることを示す。

【0237】

かくして、一時的に、今までに記載されている各種実施形態を要約すると、スライス・レベル当たりのNALユニットヘッダに含まれるものを越えた更なる高水準構文アイテムと同様にSEIメッセージを適用できるようにする更なる高レベルの構文シグナル伝達戦略が提示された。したがって、我々は、スライス接頭辞NALユニットを解説した。スライス接頭辞および`slice__level`/サブピクチャSEIメッセージの構文および

50

意味論は、低い遅延 / サブピクチャ C P B 動作、タイル・シグナリングおよび R O I シグナリングのための使用事例とともに記載されていた。拡張した構文は、加えて、スライス接頭辞の以下のスライスのスライス・ヘッダの信号部分に提示された。

【 0 2 3 8 】

完全性のために、図 3 3 は、図 1 2 ~ 1 4 の実施例に従ってタイミング制御バケットのために使われることができる構文のための更なる実施例を示す。意味論は以下の通りでありえる：

【 0 2 3 9 】

現在のアクセスユニットの復号化順序の最後の復号化ユニットおよび復号化ユニット情報 S E I メッセージと関連した復号化ユニットの名目上の C P B 時間との間に、クロックサブティックを単位にして、`du__spt__cpb__removal__delay__increment` は、期間を特定する。アネックス C で特定されるように、この値は H S S のための C P B に復号化ユニットデータの到着の最も早い可能な時間を計算するためにも用いられる。構文要素は、ビットの長さが `du__cpb__removal__delay__increment__length__minus 1 + 1` によって与えられる固定長コードによって表される。復号化ユニット情報 S E I メッセージと関連した復号化ユニットが現在のアクセスユニットの最後の復号化ユニットであるときに、`du__spt__cpb__removal__delay__increment` の値は 0 に等しい。

10

【 0 2 4 0 】

1 に等しい `dpb__output__du__delay__present__flag` は復号化ユニット情報 S E I メッセージの `pic__spt__dpb__output__du__delay` 構文要素の存在を特定する。0 に等しい `dpb__output__du__delay__present__flag` は復号化ユニット情報 S E I メッセージの `pic__spt__dpb__output__du__delay` 構文要素の欠如を特定する。

20

【 0 2 4 1 】

`SubPicHrdFlag` が 1 に等しいときに、`pic__spt__dpb__output__du__delay` は画像の D P B 出力時間を計算するために用いられる。それは、C P B からアクセスユニットの最後の復号化ユニットの除去の後復号化画像が D P B から出力される前に、どれくらいのサブクロックティックを待つかについて特定する。ないときには、`pic__spt__dpb__output__du__delay` の値は `pic__dpb__output__du__delay` に等しいと推測される。構文要素 `pic__spt__dpb__output__du__delay` の長さは、`dpb__output__delay__du__length__minus 1 + 1` によってビットにおいて与えられる。

30

【 0 2 4 2 】

同じアクセスユニットと関連する全ての復号化ユニット情報 S E I メッセージが同じ動作点に適用され、1 に等しい `dpb__output__du__delay__present__flag` が `pic__spt__dpb__output__du__delay` と同じ値を有することが、ビットストリーム一致の要件である。出力タイミング同調デコーダから出力されるいかなる画像の `pic__spt__dpb__output__du__delay` にも由来する出力時間は、復号化順序におけるいかなる次の C V S のすべての画像の `pic__spt__dpb__output__du__delay` にも由来する出力時間に先行する。

40

【 0 2 4 3 】

この構文要素の値によって決められる画像アウトプット順序は、`PicOrderCntVal` の値によって確立するのと同じ順序である。

【 0 2 4 4 】

復号化順序において、`no__output__of__prior__pics__flag` が 1 に等しいか 1 に等しいと認められる I R A P 画像に先行するため「バンピング」プロセスによって出力されない画像のために、`pic__spt__dpb__output__du__delay` に由来する出力時間は、同じ C V S 内の全ての画像に対して `PicOrderCntVal` の値の増加と共に増加している。C V S のいかなる 2 つの画像のためにも、

50

SubPicHrdFlagが1に等しいときの2つの画像の出力時間の差は、SubPicHrdFlagが0に等しいときの差と同一である。

【0245】

さらに、図34は、ROIパケットを用いてROI領域の信号を送るための更なる実施例を示す。図34によれば、ROIパケットの構文は、そのスコープに属するいくつかのペイロード・パケット32に符号化された画像の全てのサブポーションがROIに属するかどうかを示す1つのフラグのみを含む。「スコープ」は、ROIパケットまたはregion_refresh_info SEIメッセージの発生まで伸びる。フラグが1である場合、領域はそれぞれの次のペイロード・パケットに符号化され、0である場合、逆のものが適用される、すなわち、画像18のそれぞれのサブポーションはROI 60に属しない。

10

【0246】

上述の実施例のいくつかを再度議論する前に、言い換えればタイル、スライス、WPP、サブストリーム、サブディビジョニングのような上で使用されているいくつかの言葉を説明する前に、前記実施例のHigh Levelシグナリングがあるいは、例えば[3-7]のような輸送仕様において定められることができる点に留意する必要がある。換言すれば、前述のパケットおよび形成シーケンス34は、いくらかがスライスのようなアプリケーションレイヤーのサブポーションを有する転送パケットであり、全体的にまたは断片化されてそこにパケット化されるように包含され、いくらかは方法において後者と上述のような目的との間で分散される。換言すれば、上述の分散型のパケットは、アプリケーションレイヤーのビデオ・コーデックに定義付けられるが、その代わりに転送プロトコルに定義付けられた特別な転送パケットになることができる別のタイプのSEIメッセージとして定義付けられるように制限されない。

20

【0247】

換言すれば、本願明細書の一態様によれば、前記実施例でビデオ内容の画像のサブポーション（符号化木ブロックまたはスライスを参照）を単位にしてその中で符号化されるビデオ内容を有するビデオ・データストリームがみられ、そして、各サブポーションがビデオ・データストリームのパケットのシーケンス（NALユニット）の1つ以上のペイロード・パケット（VCL NALユニットを参照）にそれぞれ符号化され、パケットのシーケンスはアクセスユニットのシーケンスに分割され、それにより、各アクセスユニットはビデオ内容のそれぞれの画像に関するペイロード・パケットを集めて、パケットのシーケンスがタイミング制御パケット（スライス接頭辞）をその中に分散し、それによりタイミング制御パケットがアクセスユニットを復号化ユニットに再分割し、それにより少なくともいくつかのアクセスユニットは2つ以上の復号化ユニットに再分割され、各タイミング制御パケットは復号化ユニットのためのデコーダバッファ検索時間、パケットのシーケンスのそれぞれのタイミング制御パケットが続くペイロード・パケットの信号を送る。

30

【0248】

上述の通り、ビデオ内容が画像のサブポーションを単位にするデータストリームに符号化される領域は、例えば符号化モード（例えばイントラ・モード、インターモード、細別情報など）のような前兆となる符号化、予測パラメータ（例えば、動きベクトル、外挿方向等）および/または残余のデータ（例えば、変換係数等）に関する構文要素をカバーすることができ、これらの構文要素は、例えば、それぞれ、符号化ツリー・ブロック、予測ブロックおよび残余の（例えば、変換等）ブロックのような画像のローカルポーションと関係している。

40

【0249】

上述の通り、ペイロード・パケットは、1つ以上のスライス（完全に、それぞれ）を各々含むことができる。スライスは、それぞれに復号化可能でもよいかまたはその独立復号化を妨げる相互関係を示すことができる。たとえば、エントロピー・スライスはそれぞれにエントロピー復号化可能であるが、スライス境界を越えた予測は禁止されることができ、従属するスライスは、WPP処理、すなわち重なり合う時間に従属するスライスを並

50

列に符号化／復号化する能力をもってスライス境界を越えてエントロピーおよび予測符号化を用いるが、個々の従属するスライスおよび従属するスライスによって呼ばれるスライスの符号化／復号化の交互の開始の符号化／復号化を可能にすることができる。アクセスユニットのペイロード・パケットがそれぞれのアクセス装置の範囲内で配置される経時的な命令は、前もってデコーダに知られていることができる。

【0250】

アクセスユニットのペイロード・パケットがそれぞれのアクセスユニットの範囲内に配置されている順序は、前もってデコーダに知られている。たとえば、符号化／復号化順序は、例えば上記例の符号化ツリー・ブロックの中のスキャン順序などのようなサブポーションの中で定められることができる。

10

【0251】

例えば、以下の図を参照していただきたい。現在符号化／復号化画像100は、図35および36において、見本として画像110の四半部に対応し、参照符号112a～112dで示されるタイルに分けられることができる。すなわち、全部の画像110は、図37の場合のような1つのタイルを形成することができるかまたは複数のタイルに分割されるすることができる。タイル分割は、タイルが列および行だけに配置される規則的なものに制限されるることができる。異なる実施例は、以下に示される。

【0252】

見られるように、画像110は、更に符号化（ツリー）ブロック（図中の小さい箱で上記でCTBと呼ばれている）114に再分割され、その中で、符号化順序116が定められる（ここでは、ラスタ・スキャン順序であるが、異なるものであってもよい）。タイル112a～dへのブロック114の画像の再分割は、タイルがブロック114の互いに素であるように制限されることができる。さらに、ブロック114およびタイル112a～dは、列および行の規則的配列となるように制限されることができる。

20

【0253】

タイル（すなわち1以上の）がある場合、符号化（復号化）順序116のラスタを最初に完全なタイルをスキャンし、それから - ラスタ・スキャンタイル順序で - 次のタイルにタイル順序で推移する。

【0254】

タイルは空間的な隣接から推測される空間予測および文脈選択によりタイル境界の非交差のため互いに独立している符号化／復号化であるので、エンコーダ10およびデコーダ12は、例えばタイル境界の交差が許されているインループまたはポストフィルタリングを除いて、タイル112（元は70で示された）に再分割された画像を互いに独立して並列に符号化／復号化することができる。

30

【0255】

画像110は、スライス118a～d、180 - 元は参照符号24で示されていた - に更に再分割されることができる。スライスは、タイルの部分だけ、1つの完全なタイルまたは複数のタイルを完全に含むことができる。このように、スライスへの分割は、図35のケースのようにタイルを再分割することもできる。各スライスは、完全に1つの符号化ブロック114を含み、符号化順序116で連続的な符号化ブロック114から成り、その結果、順序は図のインデックスが割り当てられたスライス118a～dの中で定められる。図35～37のスライス分割は、説明の便宜上選ばれたものである。タイル境界は、データストリームにおいて信号を送った。画像110は、図37に図示するように、単一のタイルを形成することができる。

40

【0256】

空間予測がタイル境界を横断して適用されないという点で、エンコーダ10およびデコーダ12はタイル境界に従うように構成されるすることができる。文脈適応、すなわちさまざまなエントロピー（算術的）文脈の確率適応は、全てのスライスを通じて続けることができる。しかしながら、スライス118a, bに関して図36に示すように、スライスは - 符号化順序116に沿って - タイル境界線（スライスで内側にある場合）を横切るときは

50

いつでも、スライスは、次に、各サブセクションの始まりを示しているポインタ (`c . p . e n t r y _ p o i n t _ o f f s e t`) を含むスライスを有するサブセクション (サブストリームまたはタイル) に再分割される。デコーダ ループにおいて、フィルタは、タイル境界を横切ることができる。このようなフィルタは、デブロッキング・フィルタ、 `S a m p l e A d a p t i v e O f f s e t (S A O)` フィルタおよび `A d a p t i v e l o o p` フィルタ (`A L F`) の1つ以上を含むことができる。起動する場合、後者はタイル / スライス境界を通じて適用されることができる。

【0257】

各任意の第2のおよび以下のサブセクションは、次のサブセクションに始めに1つのサブセクションの始まりからオフセットを示しているポインタを有するスライスの範囲内でバイト整列されて配置されるそれらの始まりを有することができる。サブセクションは、スキャン順序116のスライスの範囲内で配置される。図38は、見本としてサブセクション119iに再分割されている図37のスライス180cを有する実施例を示す。

10

【0258】

図に関して、タイルのサブパートを形成しているスライスがタイル112aの行を有する端部でなければならないということではないことに注意されたい。例えば、図37および図38のスライス118aを参照。

【0259】

以下の図は、上記の図38の画像110と関連したアクセスユニットに関するデータストリームの典型的な部分を示す。ここで、各ペイロード・パケット122a~d - 以前参照符号32によって示された - は、見本として、単に1つのスライス118aだけに適応する。2つのタイミング制御パケット124a, b - 以前示されたそばに参照記号36 - は、説明の便宜上アクセスユニット120に分散するとして示される: 124aはパケット順序126 (復号化 / 符号化時間軸に対応する) のパケット122aに先行し、124bはパケット122cに先行する。したがって、アクセスユニット120は2つの復号化ユニット128a, b - 以前参照符号38によって示された - に分けられ、その第1の1つはパケット122a, b (任意のフィルタデータパケット (それぞれ、第1および第2のパケット122a, bに続く) およびSEIパケットに続く任意のアクセスユニット (第1のパケット122aに先行する)) から成り、その第2の1つはパケット118c, d (任意のフィルタデータパケット (それぞれ、パケット122c, dに続く)) から成る。

20

30

【0260】

上述のように、パケットのシーケンスの各パケットは、複数のパケット・タイプ (`n a l _ u n i t _ t y p e`) から、正確に1つのパケット・タイプに割り当てられることができる。ペイロード・パケットおよびタイミング制御パケット (および、任意のフィルタデータおよびSEIパケット) は、例えば、異なるパケット・タイプのものである。パケットのシーケンスの特定のパケット・タイプのパケットのインスタンス化は、特定の限界に従属してもよい。これらの限界は、アクセス装置境界130a, bが検出可能であるように、各アクセスユニットの範囲内でパケットによって従われることになっているパケット・タイプ (図17を参照) の中の順序を定めることができ、いかなる除去可能なパケット・タイプのパケットがビデオ・データストリームから除去される場合であっても、パケットのシーケンス内の同じ位置に残る。たとえば、ペイロード・パケットは、除去不可能なパケット・タイプのものである。しかしながら、タイミング制御パケット、フィルタデータパケットおよびSEIパケットは、上述のように、除去可能なパケット・タイプのものでもよい、すなわち、それらは、非VCLNALユニットでもよい。

40

【0261】

上記の実施例において、タイミング制御パケットは、`s l i c e _ p r e f i x _ r b s p 0` の構文によって、上で明確に例証された。

【0262】

このようなタイミング制御パケットの分散を用いて、エンコーダは、ビデオ内容の個々

50

の画像を符号化する過程でデコーダ側でバッファ・スケジューリングを調整することを可能にされる。たとえば、エンコーダは、端末間の遅延を最小化するためにバッファ・スケジューリングを最適化することを可能にされる。この点に関しては、エンコーダは、ビデオ内容の個々の画像のためのビデオ内容の画像領域全体の符号化複合性の個々の配布を考慮に入れることを可能にされる。特に、エンコーダは、パケットごとに（すなわち、現在のパケットが終了するとすぐにそれは出力される）、パケット122、122a-d、122a-d1-3のシーケンスを連続的に出力することができる。タイミング制御パケットを用いて、エンコーダは、現在の画像のサブポーションのいくつかが残りのサブポーションを有するそれぞれのペイロード・パケットにすでに符号化されたが、まだ符号化されなかった場合には、復号化側で時々バッファ・スケジューリングを調整することが可能である。

10

【0263】

したがって、パケットのシーケンスがアクセスユニットのシーケンスに分割され、各アクセスユニットがビデオ内容のそれぞれの画像に関連するペイロード・パケットを集めるように、ビデオ・データストリームのパケットのシーケンス（NALユニット）の1つ以上のペイロード・パケット（VCL NALユニット）にそれぞれ各サブポーションを符号化することで、ビデオ内容の画像のサブポーションのユニット（符号化ツリー・ブロック、タイルまたはスライス参照）のビデオ・データストリームのビデオ内容に符号化するためのエンコーダは、パケットのシーケンスにタイミング制御パケット（スライス接頭辞）を分散させるように構成され、それにより、タイミング制御パケットはアクセスユニットを復号化ユニットに再分割し、それにより、少なくともいくつかのアクセスユニットは複数の復号化ユニットに再分割され、各タイミング制御パケットは復号化ユニットのために復号化バッファ検索時間の信号を送り、そのペイロード・パケットはパケットのシーケンスにおけるそれぞれのタイミング制御パケットに続く。

20

【0264】

ちょうど概説されたビデオ・データストリームを受信しているいかなるデコーダも、タイミング制御パケットに含まれるスケジュール情報を利用するかどうかは自由である。しかしながら、デコーダが情報を利用することができる間、コーデック・レベルに従っているデコーダは示されたタイミングの後にデータを復号化することが可能でなければならない。利用される場合、デコーダはそのデコーダバッファを供給して、復号化ユニットを単位にするそのデコーダバッファを空にする。「デコーダバッファ」は、上述のように、復号化ピクチャ・バッファおよび/または符号化ピクチャ・バッファを含むことができる。

30

【0265】

したがって、各アクセスユニットがビデオ内容のそれぞれの画像に関連するペイロード・パケットを集めるように、ビデオ内容の画像のサブポーション（符号化ツリー・ブロック、タイルまたはスライス参照）を単位にしてそこに符号化されるビデオ内容を有し、各サブポーションがそれぞれビデオ・データストリームのパケットのシーケンス（NALユニット）の1つ以上のペイロード・パケット（VCL NALユニットを参照）に符号化されるビデオ・データストリームを復号化するためのデコーダは、アクセスユニットをタイミング制御パケットで復号化ユニットに再分割したパケットのシーケンスに分散されたタイミング制御パケットを探すように構成され、それにより、少なくともいくつかのアクセスユニットは複数の復号化ユニットに分割され、各タイミング制御パケットから復号化ユニットのためにデコーダバッファ検索時間を引き出し、そのペイロード・パケットはパケットのシーケンスのそれぞれのタイミング制御パケットに続き、復号化ユニットのためのデコーダバッファ検索時間によって規定された時間で予定されたデコーダのバッファから復号化ユニットを検索する。

40

【0266】

タイミング制御パケットを探すことは、このことにより含まれるNALユニットヘッダおよび構文要素を検査しているデコーダを含むことができる、すなわちnal_unit_typeである。後のフラグの値がいくつかの値に等しい場合、すなわち、上述の例で

50

は、124である場合、現在検査されるパケットはタイミング制御パケットである。すなわち、タイミング制御パケットは、`subpic_timing`と同様に疑似コード`subpic_buffering`に関して上で説明される情報を含むかまたは伝達する。すなわち、タイミング制御パケットは、デコードのための最初のCPB除去遅延を伝達するか特定し、またはそれぞれのデコードユニットのCPBからの除去の後、どれくらいのクロックが経過するかについて特定することができる。

【0267】

意図せずに更にアクセスユニットを復号化ユニットに分割することなくタイミング制御パケットの反復伝送を可能にするために、タイミング制御パケットの範囲内のフラグは、現在のタイミング制御パケットがアクセスユニット内において符号化ユニットに再分割されるかどうかの信号を明確に送る（復号化ユニットの始まりを示している`decoding_unit_start_flag = 1`および逆の状況の信号を送っている`decoding_unit_start_flag = 0`を比較）。

10

【0268】

タイル識別パケットがデータストリームに分散するという点で、分散型の復号化ユニット関連のタイル識別情報を使用する態様は分散型の復号化ユニット関連のタイミング制御パケットを使用する態様と異なる。上述のタイミング制御パケットは加えて、データストリームに分散することができ、または、デコードバッファ検索時間は共通に同じパケットの範囲内で下に説明されたタイル識別情報とともに伝達される。したがって、上述のセクションにおいて前方に持ってこられる詳細が、以下の説明における問題を明らかにするために使用されることができる。

20

【0269】

上記の実施例から導き出せる本願明細書の更なる態様は、そこにおいて符号化されるビデオ内容を有するビデオ・データストリームを明らかにし、ビデオ内容の画像が空間的に再分割するスライスを単位にして予測およびエントロピー符号化を用い、ビデオ内容の画像が空間的に再分割されるタイルの内部に予測符号化および/またはエントロピー符号化を制限するスライスの間で符号化順序を使用して、符号化順序におけるスライスのシーケンスは符号化順序における符号化ビデオ・データストリームのスライスのシーケンスは、符号化順序のビデオ・データストリームにおけるパケットのシーケンス（NALユニット）のパケットをペイロードにパケット化し、パケットのシーケンスはアクセスユニットのシーケンスに分割され、その結果、各アクセスユニットはビデオ内容のそれぞれの画像に関するスライスがパケット化されたペイロード・パケットを集め、パケットのシーケンスは、パケットのシーケンスのそれぞれのタイル識別パケットの後に1つ以上のペイロード・パケットにパケット化されたスライス（潜在的にたった1つの）によってすぐ覆われるタイル（潜在的にたった1つの）を確認してそこへ分散するタイル識別パケットを有する。

30

【0270】

たとえば、すぐ前の図がデータストリームを示していることを参照されたい。パケット124aおよび124bは、現在タイル識別パケットを表す。明確なシグナリング（`signaling_slice_flag = 1`を比較）によって、または、慣例につき、タイル識別パケットは、直ちに続くペイロード・パケット122aにパケット化されたスライスによって覆われるタイルを確認することができるだけである。あるいは、明確なシグナリングによって、または、慣例につき、タイル識別パケット124aは、現在のアクセスユニット120の端部130bのより初期のものまでパケットのシーケンスのそれぞれのタイル識別パケット124aの後に1つ以上のペイロード・パケットにパケット化されたスライスによって覆われるタイルおよび、それぞれ、次の復号化ユニット128bを始めることを確認することができる。例えば、図35を参照：各スライス118a-d1-3が別にそれぞれのパケット122a-d1-3へのパケット化である場合、復号化ユニットへの再分割は、パケットが{122a1-3}および{122b1-3}に従って3つの復号化ユニットにグループ化されるようにされ、第3の復号化ユニットのパケット{122c1-3、1

40

50

2 2 d 1 3} にパケット化されるスライス { 1 1 8 c 1-3、 1 1 8 d 1-3 } は例えばタイル 1 1 2 c および 1 1 2 d を覆い、対応するスライス接頭辞は例えば完全な復号化ユニットに言及するとき “ c ” および “ d ”、すなわちこれらのタイル 1 1 2 c および 1 1 2 d を示す。

【 0 2 7 1 】

このように、更に下に記載のネットワークエンティティは、パケットのシーケンスの識別パケットの直後の 1 つ以上のペイロード・パケットと各タイル識別パケットを関連させるために、この明確なシグナリングまたは慣例を使用することができる。識別が信号を送られることができる方法は、疑似コード `subpic_tile_info` を経由して上で見本として記載されていた。関連するペイロード・パケットは、「接頭辞スライス」として前述していた。当然、実施例は、変更されることができる。例えば、構文要素「`tile_priority`」は、外されることができる。さらに、構文要素の中の順序は切替えられることができ、そして、可能なビット長に関する記述子および構文要素の原則の符号化は単に図示するだけである。

10

【 0 2 7 2 】

ビデオ・データストリームを受信するネットワークエンティティ（すなわち、そこにおいて符号化されるビデオ内容を有するビデオ・データストリームを明らかにし、ビデオ内容の画像が空間的に再分割するスライスを単位にして予測およびエントロピー符号化を用い、ビデオ内容の画像が空間的に再分割されるタイルの内部に予測符号化および/またはエントロピー符号化を制限するスライスの中で符号化順序を使用して、符号化順序におけるスライスのシーケンスは、符号化順序のビデオ・データストリームにおけるパケットのシーケンス（NAL ユニット）のパケットをペイロードにパケット化し、パケットのシーケンスはアクセスユニットのシーケンスに分割され、その結果、各アクセスユニットはビデオ内容のそれぞれの画像に関するスライスがパケット化されたペイロード・パケットを集め、パケットのシーケンスは、その中に分散するタイル識別パケットを有する）は、タイル識別パケットに基づいて、パケットのシーケンスのそれぞれのタイル識別パケットの後に 1 つ以上のペイロード・パケットにスライス・パケット化によって覆われるタイルを確認するように構成することができる。ネットワークエンティティは、伝送作業を決定するために、識別結果を使用することができる。たとえば、ネットワークエンティティは、再生のための異なる優先度を有する異なるタイルを扱うことができる。たとえば、パケット損失の場合には、より高い優先度のタイルに関するそれらのペイロード・パケットは、低い優先度のタイルに関するペイロード・パケット上を再伝送されることが好ましい。すなわち、ネットワークエンティティは、より高い優先度のタイルに関する失われたペイロード・パケットの再伝送を最初に要請することができる。単に（転送率に応じて）十分な時間が残されているだけの場合には、ネットワークエンティティは、低い優先度のタイルに関する失われたペイロード・パケットの再伝送を要請することに移る。しかしながら、ネットワークエンティティは、特定のタイルに関するタイルまたはペイロード・パケットを異なるスクリーンに割り当てることが可能であるプレイバックユニットでもよい。

20

30

【 0 2 7 3 】

分散型の関心領域情報を使用する態様に関して、以下の ROI パケットが上述したタイミング制御パケットおよび/またはタイル識別パケットによって、スライス接頭辞に関して上述したように共通のパケットの範囲内でその情報内容を結合することによっても、または、別々のパケットの形で共存することができる点に留意する必要がある。

40

【 0 2 7 4 】

上記の通りに分散型の関心領域情報を使用する態様で、換言すれば、ビデオ・データストリームはそこにおいて予測およびエントロピー符号化を用いて符号化されるビデオ内容を有し、スライスの中の符号化順序を用いて、ビデオ内容の画像が空間的に分割され、ビデオ内容の画像が分割されるタイルの内部への予測符号化の予測および/またはエントロピー符号化を制限し、符号化順序のスライスのシーケンスが、符号化順序のビデオ・データストリームのパケットのシーケンス（NAL ユニット）のペイロード・パケットにパケ

50

ット化され、各アクセスユニットがビデオ内容のそれぞれの画像に関するスライスをそこへパケット化されたペイロード・パケットを集めるようにパケットのシーケンスはアクセスユニットのシーケンスに分割され、パケットのシーケンスは、それぞれ、画像のROIに属する画像のタイルを確認してそこに分散するROIパケットを有する。

【0275】

ROIパケットに関して、類似のコメントは、タイル識別パケットに関して以前に与えられているそれらとして有効である：ROIパケットは、単にそれぞれのROIパケットがそれが「接頭辞スライス」に関して上述したように1つ以上のペイロード・パケットに直ちに先行するために関連する1つ以上のペイロード・パケットに含まれるスライスによって覆われるそれらのタイルの中だけの画像のROIに属する画像のタイルを確認することができる。

10

【0276】

ROIパケットは、これらのROI (`c.p.num__rois__minus1`)の各々のための関連するタイルを確認することに関する前に置かれたスライスにつき複数のROIを確認することを可能にすることができる。それから、ROIごとに、優先度 (`c.p.roi__priority[i]`) に関してROIをランク付けすることを可能にする優先度が送信されることができる。ビデオの画像シーケンスの間、時間とともにROIの「追跡」を可能にするために、ROIパケットに示されるROIが画像の境界を越えて/全域で、すなわち時間とともに (`c.p.roi__id[i]`)、互いに関連するように、各ROIは、ROIインデックスによってインデックスを付けられるようにすることができる。

20

【0277】

ビデオ・データストリームを受信するネットワークエンティティ (すなわち、ビデオ・データストリームはそこにおいて予測およびエントロピー符号化を用いて符号化されるビデオ内容を有し、スライス間の符号化順序を用いて、ビデオ内容の画像が空間的に分割され、ビデオ内容の画像が分割されるタイルの内部への予測符号化の予測を制限し、符号化順序のスライスのシーケンスが、符号化順序のビデオ・データストリームのパケットのシーケンス (NALユニット) のペイロード・パケットにパケット化され、各アクセスユニットがビデオ内容のそれぞれの画像に関するスライスをそこへパケット化されたペイロード・パケットを集めるようにパケットのシーケンスはアクセスユニットのシーケンスに分割される) は、タイル識別パケットに基づいて、画像のROIに属するタイルを覆うスライスをパケット化しているパケットを確認するように構成される。

30

【0278】

ネットワークエンティティは、タイル識別パケットに関して前述したように前のセクションにおいて同様の方法のROIパケットによって伝達される情報を利用することができる。以前のセクションと同様に現在のセクションに関して、上述のようにデータストリームにおいて明確に信号を送り、または慣例によりエンコーダまたはデコーダに知られる画像におけるタイルの位置に関して、単に、画像のスライスのスライス順序を調査することにより、およびこれらのスライスが覆う現在の画像の部分の進捗を調査することにより、単に以下によって、例えばMANEまたはデコーダなどのようないかなるネットワークエンティティもどのタイルが現在検査されるペイロード・パケットのスライスによって覆われるかについて確認することが可能な点に留意する必要がある。あるいは、各スライス (スキャン順序の画像で第1のものを除いて) は、デコーダがスライス順序の方向へのこの第1の符号化ブロックから各スライス (その再建) を画像に配置することができるように、それが (同じ符号) を参照する第1の符号化ブロック (例えばCTB) の表示/インデックス (符号化ツリー・ブロックを単位にして測定される `slice__address`) を備えていることができる。したがって、後のタイル識別パケットによって伝達されるインデックスが1以上の差で以前のものとは異なる場合、それらの2つのタイル識別パケット間のペイロード・パケットがその間でタイル・インデックスを有するタイルをカバーすることに従って次のタイル識別パケットに遭遇すると、即座に、ネットワークエンティティ

40

50

に対して明らかになるので、上述したタイル情報パッケージが単にそれぞれのタイル識別パッケージの後に付随する1つ以上のペイロード・パッケージのスライスを含む第1のタイル (`first_tile_id_in_prefixed_slices`) のインデックスから成るだけである場合、それは十分でありえる。スライスが符号化ブロックの中のこのラスタ・スキャン順序に沿ってスライス順序に従って各々に続くので、上記したように、細別および符号化ブロック細別が、例えば、ラスタ・スキャン順序を有する行/列-型の細別に基づいてある両方のタイルがいずれが、例えばタイルおよび符号化ブロック行型、すなわちこのラスタ・スキャン順序のタイル・インデックス増加であるかについてそれらの間で定められる場合、これは真である。

【0279】

10

上述の実施例から導き出される上述の信号を送るパッケージ化された分散型のスライス・ヘッダの態様は、記載した前記実施例から導き出せる上述した態様またはいかなるそれらの組み合わせのいずれか1つとでも結合可能である。前に明確に記載されたスライス接頭辞は、例えば、バージョン2によってすべてのこれらの態様を統一する。現在の態様の効果は、それらが前に置かれたスライス/ペイロード・パッケージに対して外の自己充足的なパッケージにおいて伝えられるように、より容易にスライス・ヘッダデータをネットワークエンティティに利用できるようにする可能性であり、そして、スライス・ヘッダデータの反復伝送は可能にされる。

【0280】

20

したがって、本願明細書の更なる態様はパッケージ化および分散型のスライス・ヘッダ・シグナリングの態様であって、換言すれば、ビデオ内容の画像のサブポーション(符号化ツリー・ブロックまたはスライスを参照)を単位にしてその中で符号化されるビデオ内容を有するビデオ・データストリームを公開するとみなされることができ、各サブポーションがビデオ・データストリームのパッケージのシーケンス(NALユニット)の1つ以上のペイロード・パッケージ(VCL NALユニットを参照)に符号化され、パッケージのシーケンスは各アクセスユニットがビデオ内容のそれぞれの画像に関するペイロード・パッケージを集めるようにアクセスユニットのシーケンスに分割され、パッケージのシーケンスは、そこへ、パッケージのシーケンスのそれぞれのスライス・ヘッダ・パッケージに続く1つ以上のペイロード・パッケージ間の、および欠けているスライス・ヘッダデータを伝達しているスライス・ヘッダ・パッケージ(スライス接頭辞)を分散させた。

30

【0281】

ビデオ・データストリーム(すなわち、ビデオ内容の画像のサブポーション(符号化ツリー・ブロックまたはスライスを参照)を単位にしてその中で符号化されるビデオ内容を有するビデオ・データストリームを公開するとみなされることができ、各サブポーションがビデオ・データストリームのパッケージのシーケンス(NALユニット)の1つ以上のペイロード・パッケージ(VCL NALユニットを参照)に符号化され、パッケージのシーケンスは各アクセスユニットがビデオ内容のそれぞれの画像に関するペイロード・パッケージを集めるようにアクセスユニットのシーケンスに分割され、パッケージのシーケンスは、そこへスライス・ヘッダ・パッケージを分散させた)を受信するネットワークエンティティは、スライス・ヘッダ・パッケージのスライス・ヘッダデータから引き出されるパッケージのスライスのためのペイロードデータに沿って、およびパッケージのシーケンスのそれぞれのスライス・ヘッダ・パッケージに続くが、1つ以上のペイロード・パッケージが続くスライス・ヘッダ・パッケージから引き出されるスライス・ヘッダを適用する1つ以上のペイロード・パッケージのためのスライス・ヘッダをスキッピング、読み込みを行って、スライス・ヘッダを読むように構成される。

40

【0282】

上述した態様に関して本当であったので、パッケージここではスライス・ヘッダ・パッケージは、MANEまたはデコーダのようないかなるネットワークエンティティも、復号化ユニットの始まりまたはそれぞれのパッケージによって前に置かれる1つ以上のペイロード・パッケージの実行の始まりに指示する機能を有することもできる。したがって、現在の態様

50

に従うネットワークエンティティは、スライス・ヘッダを読むことが `decoding_unit_start_flag` と結合してこのパケット、すなわち、`single_slice_flag` の上述した構文要素に基づいて、例えば、スキップされなければならないペイロード・パケットを確認することができ、その中で、後のフラグは、上述のように、復号化ユニットの範囲内で特定のスライス・ヘッダ・パケットのコピーの再伝送を可能にする。たとえば、1つの復号化ユニットの範囲内のスライスのスライス・ヘッダがスライスのシーケンスに沿って変化することができるので、これは役立つ、したがって、復号化ユニットの初めのスライス・ヘッダ・パケットが(1に等しく)設定される `decoding_unit_start_flag` を有することができると共に、その間に配置されるスライス・ヘッダ・パケットは設定されないこのフラグを有することができ、いかなるネットワークエンティティも新しい復号化ユニットを始めることとして不正にこのスライス・ヘッダ・パケットの発生を読み取ることを防止する。

10

20

30

40

50

【0283】

いくつかの態様が装置の前後関係に記載されていたにもかかわらず、これらの態様も対応する方法の説明を表すことは明らかであり、ブロックまたは装置は、方法ステップまたは方法ステップの特徴に対応する。同様に、方法ステップの前後関係に記載されている態様は、対応する装置の対応するブロックまたは部材または特徴の説明を表す。いくつかのまたは全ての方法のステップは、ハードウェア装置、例えばマイクロプロセッサ、プログラム可能なコンピュータまたは電子回路(を用いること)によって実行されることができる。いくつかの実施形態では、最も重要な方法ステップの一つ以上は、この種の装置によって実行されることができる。

【0284】

発明のビデオ・データストリームは、デジタル記憶媒体に保存されることができ、またはワイヤレス伝送媒体またはインターネットなどの有線の伝送媒体のような伝送媒体上に送信されることができる。

【0285】

特定の実施要件に応じて、本発明の実施例は、ハードウェアにおいて、または、ソフトウェアで実施されることができる。実施はその上に格納される電子的に読み込み可能な制御信号を有するデジタル記憶媒体、例えばフレキシブルディスク、DVD、ブルーレイ、CD、ROM、PROM、EPROM、EEPROMまたはフラッシュメモリを使用して実行されることができ、それぞれの方法が実行されるように、それはプログラム可能なコンピュータシステムと協働する(または協働することができる)。したがって、デジタル記憶媒体は、計算機可読でもよい。

【0286】

本発明によるいくつかの実施例は、本願明細書において記載されている方法のうちの1つが実行されるように、プログラム可能なコンピュータシステムと協働することができる、電子的に読み込み可能な制御を有するデータキャリアは信号を有するデータキャリアを含む。

【0287】

通常、本発明の実施例は、コンピュータ・プログラム製品がコンピュータで動くときに、プログラムコードが方法のうちの1つを実行するための実施プログラムコードを有するコンピュータ・プログラム製品として実施されることができる。プログラムコードは、例えば、機械読み取り可読キャリアに格納されることができる。

【0288】

他の実施例は、本願明細書において記載されていて、機械読み取り可読キャリアに格納される方法のうちの1つを実行するためのコンピュータプログラムを含む。

【0289】

換言すれば、発明の方法の実施例は、従って、コンピュータプログラムがコンピュータで動くとき、本願明細書において記載されている方法のうちの1つを実行するためのプログラムコードを有するコンピュータプログラムである。

【0290】

発明の方法の更なる実施例は、従って、その上に記録されて、本願明細書において記載されている方法のうちの1つを実行するためのコンピュータプログラムから成っているデータキャリア（またはデジタル記憶媒体またはコンピュータ可読媒体）である。データキャリア、デジタル記憶媒体または記録媒体は、典型的には有形および/または非移行である。

【0291】

発明の方法の更なる実施例は、従って、本願明細書において記載されている方法のうちの1つを実行するためのコンピュータプログラムを表しているデータストリームまたは信号のシーケンスである。データストリームまたは信号のシーケンスは、データ通信接続、例えばインターネットを介して転送されるように構成されることができる。

10

【0292】

更なる実施例は、本願明細書の方法の1つを実行するために適用されるように構成される処理手段、例えばコンピュータまたはプログラム可能な論理装置を含む。

【0293】

更なる実施例は、本願明細書において記載されている方法のうちの1つを実行するためのコンピュータプログラムをインストールしたコンピュータを含む。

【0294】

本発明による更なる実施例は、レシーバに本願明細書において記載されている方法のうちの1つを実行するためのコンピュータプログラムを伝達する（例えば、電子的に、または、光学的に）ように構成される装置またはシステムを含む。レシーバは、例えば、コンピュータ、モバイル機器、メモリデバイス等でもよい。装置またはシステムは、例えば、コンピュータプログラムを受取人に譲渡するためのファイル・サーバを含む。

20

【0295】

いくつかの実施形態では、プログラム可能な論理装置（例えばフィールド・プログラム可能なゲート・アレイ）は、本願明細書において記載されている方法の機能のいくらかまたは全てを実行するために用いることができる。いくつかの実施形態では、フィールド・プログラム可能なゲート・アレイは、本願明細書において記載されている方法のうちの1つを実行するために、マイクロプロセッサと協働することができる。通常、方法は、いかなるハードウェア装置によっても好ましくは実行される。

30

【0296】

上記した実施例は、本発明の原理のために、単に図示するだけである。配置の修正変更および本願明細書において記載されている詳細が他の当業者にとって明らかであるものと理解される。

したがって、間近に迫った特許クレームの範囲だけによって制限されることが意図するところであり、本願明細書において実施例の説明および説明として示される具体的な詳細だけによって制限されることは意図していないところである。

【0297】

参照

[1] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Syst. Video Technol., vol. 13, N7, July 2003.

40

[2] JCT-VC, "High-Efficiency Video Coding (HEVC) text specification Working Draft 7", JCTVC-I1003, May 2012.

[3] ISO/IEC 13818-1: MPEG-2 Systems specification.

[4] IETF RFC 3550 - Real-time Transport Protocol.

50

- [5] Y.-K. Wang et al. , "RTP Payload Format for H.264 Video", IETF RFC6184, <http://tools.ietf.org/html/>
- [6] S. Wenger et al., "RTP Payload Format for Scalable Video Coding", IETF RFC6190, <http://tools.ietf.org/html/rfc6190>
- [6] T. Schierl et al., "RTP Payload Format for High Efficiency Video Coding", IETF internet draft, <http://datatracker.ietf.org/doc/draft-schierl-payload-rtp-h265/>

【 図 1 】

	記述子
buffering_period(payloadSize) {	
seq_parameter_set_id	ue(v)
if(!sub_pic_cpb_params_present_flag)	
rap_cpb_params_present_flag	u(1)
if(!NalHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag	
rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	
if(VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(v)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(v)
if(sub_pic_cpb_params_present_flag	
rap_cpb_params_present_flag) {	
initial_alt_cpb_removal_delay[SchedSelIdx]	u(v)
initial_alt_cpb_removal_delay_offset[SchedSelIdx]	u(v)
}	
}	
}	
}	

FIG 1

【 図 2 】

	記述子
pic_timing(payloadSize) {	
if(CpbDpbDelaysPresentFlag) {	
cpb_removal_delay	u(v)
dpb_output_delay	u(v)
if(sub_pic_cpb_params_present_flag) {	
num_decoding_units_minus1	ue(v)
for(i = 0; i <= num_decoding_units_minus1; i++) {	
num_nalus_in_du_minus1[i]	ue(v)
du_cpb_removal_delay[i]	u(v)
}	
}	
}	

FIG 2

【 図 3 a 】

記述子	記述子
vui_parameters() {	
aspect_ratio_info_present_flag	u(1)
if(aspect_ratio_info_present_flag) {	
aspect_ratio_idc	u(8)
if(aspect_ratio_idc == Extended_SAR) {	
sar_width	u(16)
sar_height	u(16)
}	
}	
overscan_info_present_flag	u(1)
if(overscan_info_present_flag)	
overscan_appropriate_flag	u(1)
video_signal_type_present_flag	u(1)
if(video_signal_type_present_flag) {	
video_format	u(3)
video_full_range_flag	u(1)
colour_description_present_flag	u(1)
if(colour_description_present_flag) {	
colour_primaries	u(8)
transfer_characteristics	u(8)
matrix_coefficients	u(8)
}	
}	
chroma_loc_info_present_flag	u(1)
if(chroma_loc_info_present_flag) {	
chroma_sample_loc_type_top_field	ue(v)
chroma_sample_loc_type_bottom_field	ue(v)
}	
neutral_chroma_indication_flag	u(1)
}	

FIG 3A

FIG 3
FIG 3A
FIG 3B

【 図 3 b 】

field_seq_flag	u(1)
timing_info_present_flag	u(1)
if(timing_info_present_flag) {	
num_units_in_tick	u(32)
time_scale	u(32)
fixed_pic_rate_flag	u(1)
}	
nal_hrd_parameters_present_flag	u(1)
if(nal_hrd_parameters_present_flag)	
hrd_parameters() {	
vcl_hrd_parameters_present_flag	u(1)
if(vcl_hrd_parameters_present_flag)	
hrd_parameters() {	
if(nal_hrd_parameters_present_flag vcl_hrd_parameters_present_flag)	
low_delay_hrd_flag	u(1)
sub_pic_cpb_params_present_flag	u(1)
if(sub_pic_cpb_params_present_flag)	
num_units_in_sub_tick	u(32)
}	
}	
}	
}	
bitstream_restriction_flag	u(1)
if(bitstream_restriction_flag) {	
tiles_fixed_structure_flag	u(1)
motion_vectors_over_pic_boundaries_flag	u(1)
max_bytes_per_pic_denom	ue(v)
max_bits_per_min_cu_denom	ue(v)
log2_max_mv_length_horizontal	ue(v)
log2_max_mv_length_vertical	ue(v)
}	
}	

FIG 3B

FIG 3
FIG 3A
FIG 3B

【 図 4 】

記述子	記述子
hrd_parameters() {	
cpb_cnt_minus1	ue(v)
bit_rate_scale	u(4)
cpb_size_scale	u(4)
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
bit_rate_value_minus1[SchedSelIdx]	ue(v)
cpb_size_value_minus1[SchedSelIdx]	ue(v)
cbr_flag[SchedSelIdx]	u(1)
}	
initial_cpb_removal_delay_length_minus1	u(5)
cpb_removal_delay_length_minus1	u(5)
dpb_output_delay_length_minus1	u(5)
time_offset_length	u(5)
}	

FIG 4

【 図 5 】

記述子	記述子
filler_data_rossp() {	
while(next_bits(8) == 0xFF)	
ff_byte /* equal to 0xFF */	f(8)
rossp_trailing_bits()	
}	

FIG 5

【 図 6 】

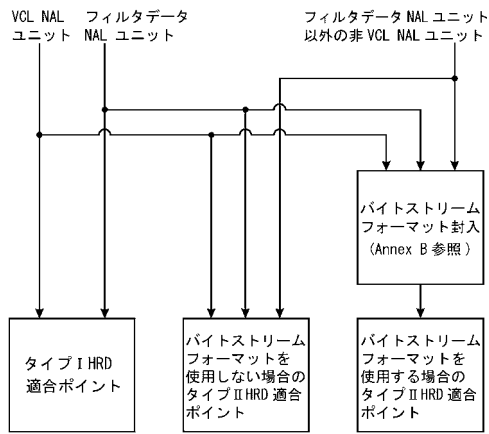


FIG 6

【 図 7 】

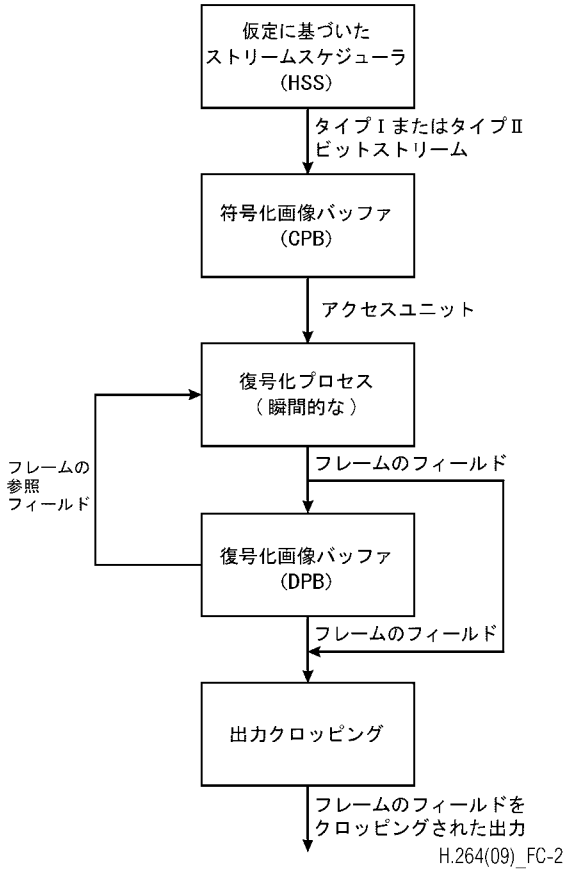


FIG 7

【 図 8 】

記述子	記述子
slice_header() {	
first_slice_in_pic_flag	u(1)
pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
if(!dependent_slice_flag) {	
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(RapPicFlag) {	
rap_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
}	
if(!ldrPicFlag) {	
....	
}	
if(tiles_or_entropy_coding_sync_idc == 1 tiles_or_entropy_coding_sync_idc == 2) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset[i]	u(v)
}	
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte	u(8)
}	
byte_alignment()	
}	

FIG 8

【 図 9 】

記述子	記述子
pic_parameter_set_irspx() :	
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
sign_data_hiding_flag	u(1)
if(sign_data_hiding_flag)	
sign_hiding_threshold	u(4)
cabac_init_present_flag	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
[Ed. (B8): not present in HM software]	
pic_init_qp_minus26	se(v)
constrained_intra_pred_flag	u(1)
slice_granularity	u(2)
diff_cu_qp_delta_depth	ue(v)
cb_qp_offset	se(v)
cr_qp_offset	se(v)
weighted_pred_flag	u(1)
weighted_bipred_idc	u(2)
output_flag_present_flag	u(1)
transquant_bypass_enable_flag	u(1)
dependent_slice_enabled_flag	u(1)
tiles_or_entropy_coding_sync_idc	u(2)
if(tiles_or_entropy_coding_sync_idc == 1) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
uniform_spacing_flag	u(1)
if(!uniform_spacing_flag) {	
for(i = 0; i < num_tile_columns_minus1; i++)	
column_width[i]	ue(v)
for(i = 0; i < num_tile_rows_minus1; i++)	
row_height[i]	ue(v)
}	
loop_filter_across_tiles_enabled_flag	u(1)
} else if(tiles_or_entropy_coding_sync_idc == 2)	
cabac_independent_flag	u(1)
....	
}	

FIG 9

【 図 10 a 】

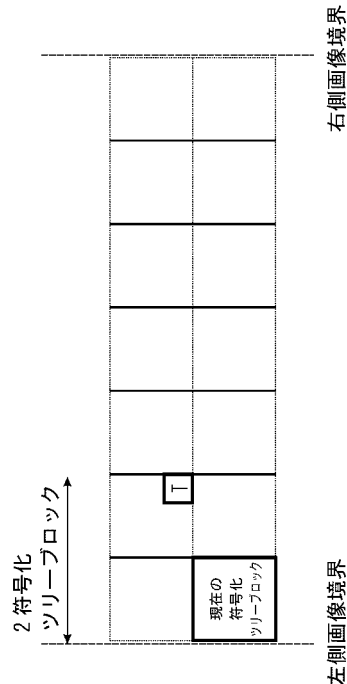


FIG 10A

【図10b】

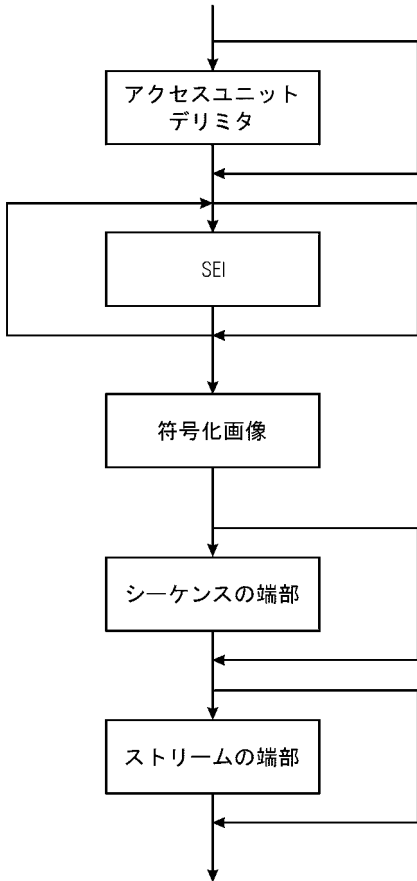


FIG 10B

【図12】

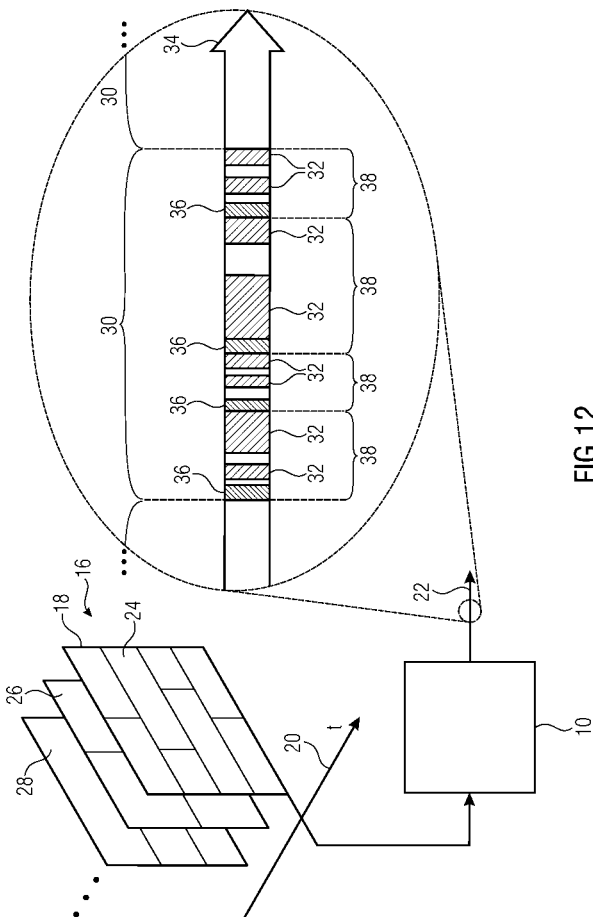


FIG 12

【図11】

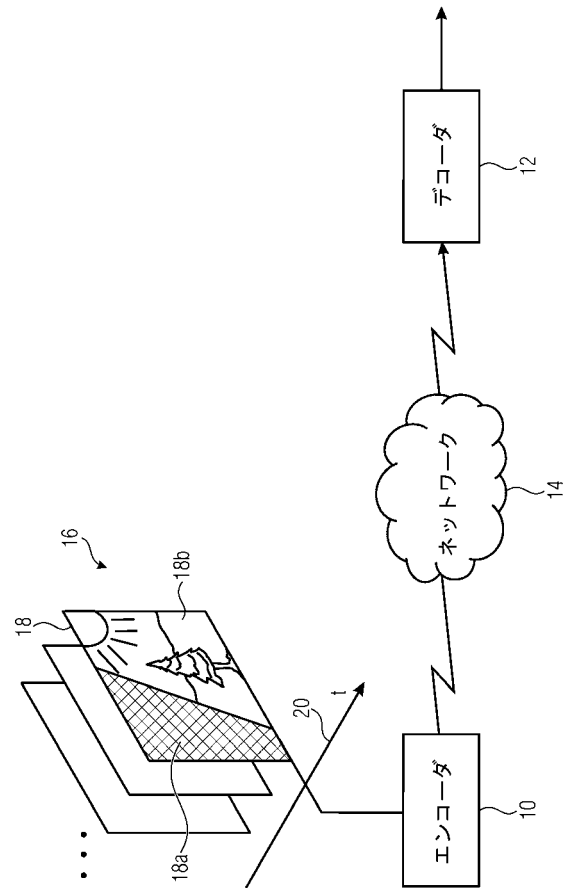


FIG 11

【図13】

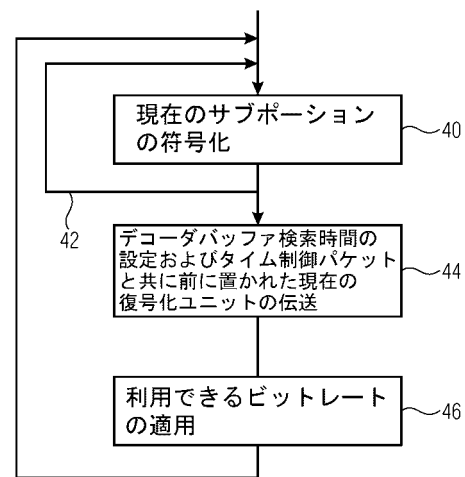


FIG 13

【図14】

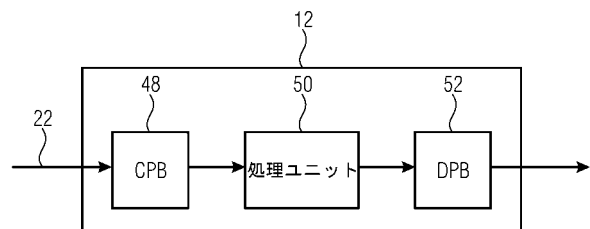


FIG 14

【図15】

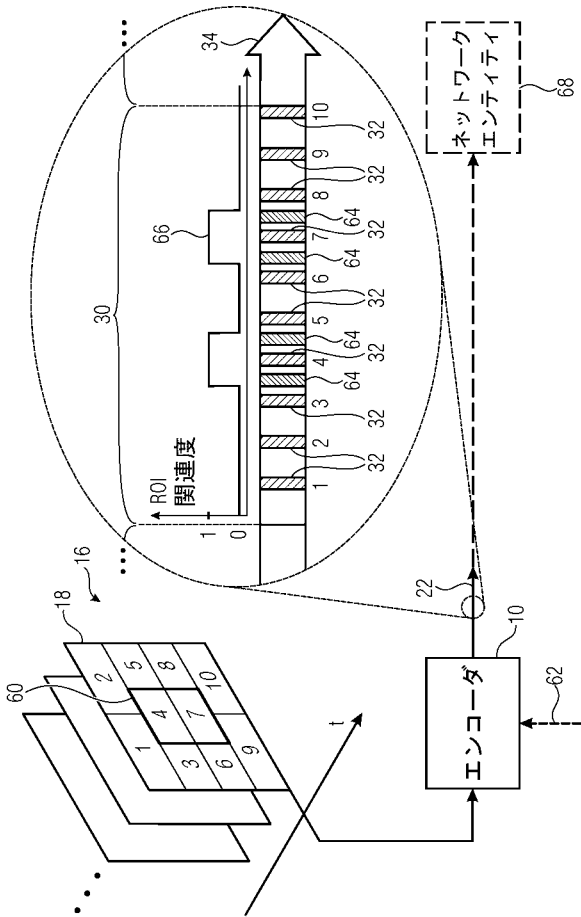


FIG 15

【図16】

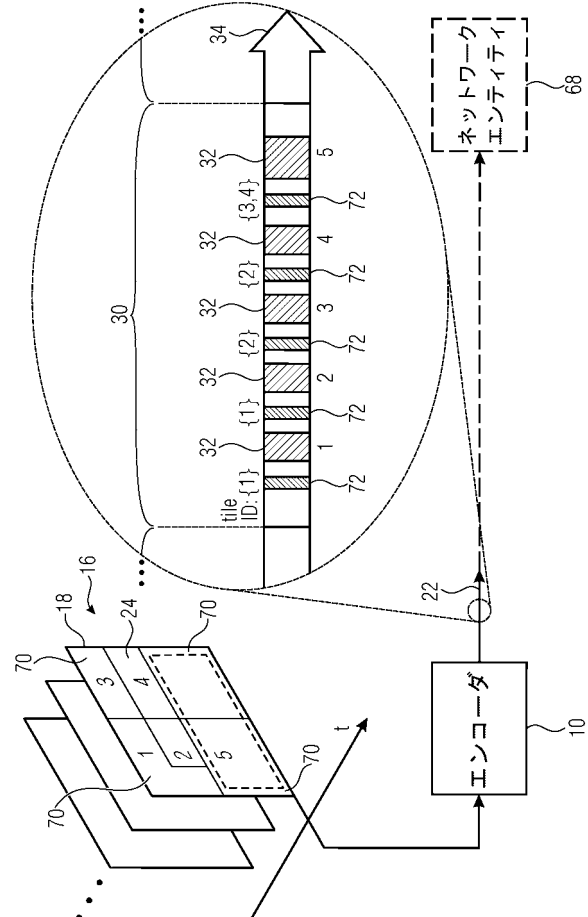


FIG 16

【図17】

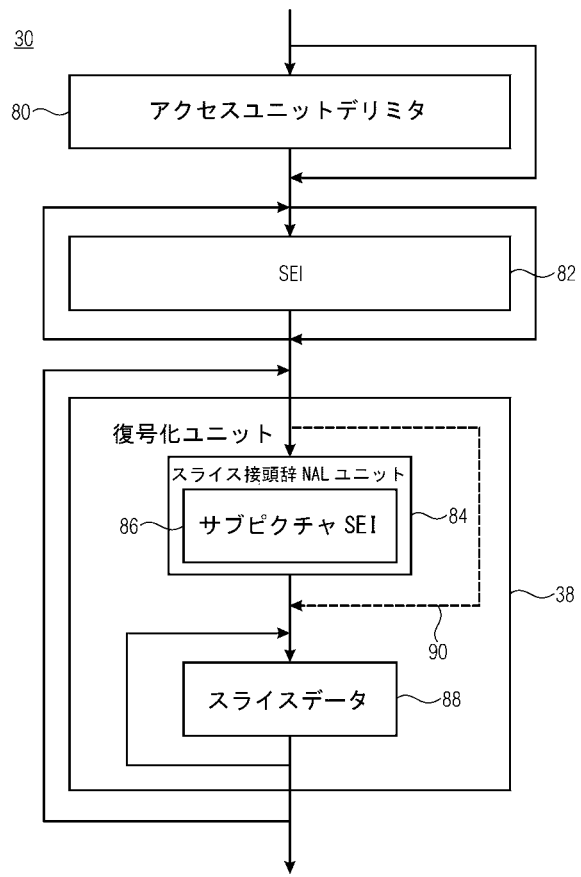


FIG 17

【図18】

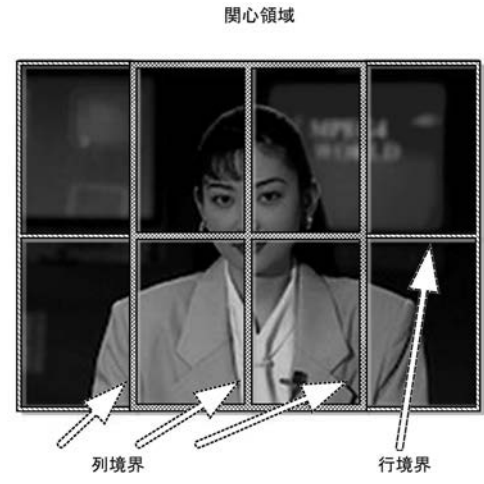


FIG 18

【図19】

記述子	
slice_prefix_rbsp() {	
reserved_one_8bits	u(8)
do	
sei_message()	
while(more_rbsp_data())	
rbsp_trailing_bits()	
}	

FIG 19

【 図 2 0 】

記述子	
slice_prefix_rbsp() {	
slice_prefix_id	u(2)
rap_flag	u(1)
decoding_unit_start_flag	u(1)
single_slice_flag	u(1)
tile_idc	u(2)
prefix_slice_header_data_present_flag	u(1)
if(prefix_slice_header_data_present_flag)	
slice_header_data()	
if(tile_idc > 0) {	
if(tile_idc == 2) /* multiple tiles in prefixed slices */	
num_tiles_in_prefixed_slices_minus1	u(16)
first_tile_id_in_prefixed_slices	u(16)
}	
do	
sei_message()	
while(more_rbsp_data())	
rbsp_trailing_bits()	
}	

FIG 20

【 図 2 1 】

nal_unit_type	NAL ユニットの内容及び RBSP 構文構造	NAL ユニット タイプクラス
0	Unspecified	non-VCL
1	Coded slice of a non-RAP non-TFD and non-TLA picture slice_layer_rbsp()	VCL
2	Coded slice of a TFD picture slice_layer_rbsp()	VCL
3	Coded slice of a non-TFD TLA picture slice_layer_rbsp()	VCL
4, 5	Coded slice of a CRA picture slice_layer_rbsp()	VCL
6, 7	Coded slice of a BLA picture slice_layer_rbsp()	VCL
8	Coded slice of an IDR picture slice_layer_rbsp()	VCL
9, 2, 3, 4	Reserved	n/a
24	Slice Prefix NAL unit	non-VCL
25	Video parameter set video_parameter_set_rbsp()	non-VCL
26	Sequence parameter set seq_parameter_set_rbsp()	non-VCL
27	Picture parameter set pic_parameter_set_rbsp()	non-VCL
28	Adaptation parameter set aps_rbsp()	non-VCL
29	Access unit delimiter access_unit_delimiter_rbsp()	non-VCL
30	Filler data filler_data_rbsp()	non-VCL
31	Supplemental enhancement information (SEI) sei_rbsp()	non-VCL
32-47	Reserved	n/a
48-63	Unspecified	non-VCL

FIG 21

【 図 2 2 】

記述子	
slice_header() {	
first_slice_in_pic_flag	u(1)
pic_parameter_set_id	ue(v)
if(!first_slice_in_pic_flag)	
slice_address	u(v)
if(dependent_slice_enabled_flag && !first_slice_in_pic_flag)	
dependent_slice_flag	u(1)
slice_header_data_present_flag	u(1)
if(slice_header_data_present_flag && !dependent_slice_flag)	
slice_header_data()	
if(tiles_or_entropy_coding_sync_idc == 1 tiles_or_entropy_coding_sync_idc == 2) {	
num_entry_point_offsets	ue(v)
if(num_entry_point_offsets > 0) {	
offset_len_minus1	ue(v)
for(i = 0; i < num_entry_point_offsets; i++)	
entry_point_offset[i]	u(v)
}	
}	
if(slice_header_extension_present_flag) {	
slice_header_extension_length	ue(v)
for(i = 0; i < slice_header_extension_length; i++)	
slice_header_extension_data_byte	u(8)
}	
byte_alignment()	
}	

FIG 22

【 図 2 3 a 】

記述子	
slice_header_data() {	
slice_type	ue(v)
if(output_flag_present_flag)	
pic_output_flag	u(1)
if(separate_colour_plane_flag == 1)	
colour_plane_id	u(2)
if(RapPicFlag) {	
rap_pic_id	ue(v)
no_output_of_prior_pics_flag	u(1)
}	
if(!IdrPicFlag) {	
pic_order_cnt_lsb	u(v)
short_term_ref_pic_set_sps_flag	u(1)
if(!short_term_ref_pic_set_sps_flag)	
short_term_ref_pic_set(num_short_term_ref_pic_sets)	
else	
short_term_ref_pic_set_idx	u(v)
if(long_term_ref_pics_present_flag) {	
num_long_term_pics	ue(v)
for(i = 0; i < num_long_term_pics; i++) {	
poc_lsb_l[i]	u(v)
delta_poc_msb_present_flag[i]	u(1)
if(delta_poc_msb_present_flag[i])	
delta_poc_msb_cycle_l[i]	ue(v)
used_by_curr_pic_l[i]	u(1)
}	
}	
}	
if(sample_adaptive_offset_enabled_flag) {	

FIG 23A

FIG 23A
FIG 23B
FIG 23C

FIG 23

【 図 2 3 b 】

slice_sample_adaptive_offset_flag[0]	u(1)
if(slice_sample_adaptive_offset_flag[0]) {	
slice_sample_adaptive_offset_flag[1]	u(1)
slice_sample_adaptive_offset_flag[2]	u(1)
}	
}	
if(adaptive_loop_filter_enabled_flag)	
aps_id	ue(v)
if(slice_type == P slice_type == B) {	
if(sps_temporal_mvp_enable_flag)	
pic_temporal_mvp_enable_flag	u(1)
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
if(lists_modification_present_flag)	
ref_pic_list_modification()	
if(slice_type == B)	
mvd_l1_zero_flag	u(1)
if(cabac_init_present_flag && slice_type != I)	
cabac_init_flag	u(1)
slice_qp_delta	se(v)
if(deblocking_filter_control_present_flag) {	
if(deblocking_filter_override_enabled_flag)	
deblocking_filter_override_flag	u(1)

FIG 23B

FIG 23	FIG 23A
	FIG 23B
	FIG 23C

【 図 2 4 】

sei_message() {	記述子
payloadType = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadType += 255	
}	
last_payload_type_byte	u(8)
payloadType += last_payload_type_byte	
payloadSize = 0	
while(next_bits(8) == 0xFF) {	
ff_byte /* equal to 0xFF */	f(8)
payloadSize += 255	
}	
last_payload_size_byte	u(8)
payloadSize += last_payload_size_byte	
sei_payload(payloadType, payloadSize)	
}	

FIG 24

【 図 2 3 c 】

if(deblocking_filter_override_flag) {	
slice_header_disable_deblocking_filter_flag	u(1)
if(!slice_header_disable_deblocking_filter_flag) {	
beta_offset_div2	se(v)
tc_offset_div2	se(v)
}	
}	
}	
if(pic_temporal_mvp_enable_flag) {	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
if(slice_type != I &&&)	
collocated_ref_idx	ue(v)
}	
if(weighted_pred_flag && slice_type == P)	
pred_weight_table()	
if(slice_type == P slice_type == B)	
five_minus_max_num_merge_cand	ue(v)
if(adaptive_loop_filter_enabled_flag) {	
slice_adaptive_loop_filter_flag	u(1)
if(slice_adaptive_loop_filter_flag &&)	
alf_param()	
if(slice_adaptive_loop_filter_flag)	
alf_cu_control_param()	
}	
if(seq_loop_filter_across_slices_enabled_flag &&)	
slice_loop_filter_across_slices_enabled_flag	u(1)
}	

FIG 23C

FIG 23	FIG 23A
	FIG 23B
	FIG 23C

【 図 2 5 a 】

sei_payload(payloadType, payloadSize) {	記述子
if(payloadType == 0)	
buffering_period(payloadSize)	
else if(payloadType == 1)	
pic_timing(payloadSize)	
else if(payloadType == 2)	
pan_scan_rect(payloadSize)	
else if(payloadType == 3)	
filler_payload(payloadSize)	
else if(payloadType == 4)	
user_data_registered_itu_t_135(payloadSize)	
else if(payloadType == 5)	
user_data_unregistered(payloadSize)	
else if(payloadType == 6)	
recovery_point(payloadSize)	
else if(payloadType == 9)	
scene_info(payloadSize)	
else if(payloadType == 15)	
full_frame_snapshot(payloadSize)	
else if(payloadType == 16)	
progressive_refinement_segment_start(payloadSize)	
else if(payloadType == 17)	
progressive_refinement_segment_end(payloadSize)	
else if(payloadType == 19)	
film_grain_characteristics(payloadSize)	
else if(payloadType == 20)	
deblocking_filter_display_preference(payloadSize)	
else if(payloadType == 22)	
post_filter_hint(payloadSize)	
else if(payloadType == 23)	
tone_mapping_info(payloadSize)	

FIG 25A

FIG 25	FIG 25A
	FIG 25B

【 図 2 5 b 】

else if(payloadType == 45)	
frame_packing_arrangement(payloadSize)	
else if(payloadType == 47)	
display_orientation(payloadSize)	
else if(payloadType == 128)	
sop_description(payloadSize)	
else if(payloadType == 129)	
field_indication(payloadSize)	
else if(payloadType == 130)	
decoded_picture_hash(payloadSize)	
else if(payloadType == 140)	
roi_info(payloadSize)	
else if(payloadType == 180)	
subpic_buffering(payloadSize)	
else if(payloadType == 181)	
subpic_timing(payloadSize)	
else if(payloadType == 182)	
subpic_tile_info(payloadSize)	
else if(payloadType == 183)	
subpic_slice_info(payloadSize)	
else if(payloadType == 184)	
subpic_tile_dimensions_info(payloadSize)	
if(!byte_aligned()) {	
bit_equal_to_one /* equal to 1 */	
while(!byte_aligned())	
bit_equal_to_zero /* equal to 0 */	
}	

FIG 25B

FIG 25
FIG 25A
FIG 25B

【 図 2 6 】

subpic_buffering(payloadSize) {	記述子
seq_parameter_set_id	u(1)
if(!HrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(16)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(16)
}	
}	
if(!VclHrdBpPresentFlag) {	
for(SchedSelIdx = 0; SchedSelIdx <= cpb_cnt_minus1; SchedSelIdx++) {	
initial_cpb_removal_delay[SchedSelIdx]	u(16)
initial_cpb_removal_delay_offset[SchedSelIdx]	u(16)
}	
}	
}	

FIG 26

【 図 2 7 】

subpic_timing(payloadSize) {	記述子
reserved_one_8bits	u(8)
du_cpb_removal_delay	u(16)
du_dpb_output_delay	u(16)
}	

FIG 27

【 図 2 9 】

subpic_tile_info(payloadSize) {	記述子
reserved_one_4bits	u(4)
tile_priority	u(3)
multiple_tiles_in_prefixes_slices_flag	u(1)
if(multiple_tiles_in_prefixes_slices_flag)	
num_tiles_in_prefixes_slices_minus1	u(16)
first_tile_id_in_prefixes_slices	u(16)
}	

FIG 29

【 図 2 8 】

subpic_slice_info(payloadSize) {	記述子
reserved_one_7bits	u(7)
slice_header_data_flag	u(1)
if(slice_header_data_flag)	
slice_header_data()	
byte_alignment()	
}	

FIG 28

【 図 3 0 】

subpic_tile_dimensions_info(payloadSize) {	記述子
reserved_one_7bits	u(7)
multiple_tiles_in_prefixes_slices_flag	u(1)
if(multiple_tiles_in_prefixes_slices_flag)	
num_tiles_in_prefixes_slices_minus1	u(16)
for(i = 0; i <= num_tiles_in_roi_minus1; i++) {	
tile_horz_start[i]	u(16)
tile_width[i]	u(16)
tile_vert_start[i]	u(16)
tile_height[i]	u(16)
}	
}	

FIG 30

【 図 3 1 】

roi_info(payloadSize) {	記述子
reserved_one_8bits	u(8)
roi_id	u(5)
roi_priority	u(3)
num_tiles_in_roi_minus1	u(16)
for(i = 0; i <= num_tiles_in_roi_minus1; i++) {	
roi_tile_id[i]	u(16)
}	
}	

FIG 31

【 図 3 2 】

roi_info(payloadSize) {	記述子
reserved_one_3bits	u(3)
num_rois_minus1	u(5)
for(i = 0; i <= num_rois_minus1; i++) {	u(5)
roi_id[i]	u(5)
roi_priority[i]	u(3)
num_tiles_in_roi_minus1[i]	u(16)
for(n = 0; n <= num_tiles_in_roi_minus1; n++) {	
roi_tile_id[i][n]	u(16)
}	
roi_presentation_on_separate_screen[i]	u(1)
}	
}	

FIG 32

【 図 3 3 】

decoding_unit_info(payloadSize) {	記述子
decoding_unit_idx	ue(v)
if(!sub_pic_cpb_params_in_pic_timing_sei_flag)	
du_spt_cpb_removal_delay_increment	u(v)
dcb_output_du_delay_present_flag	u(1)
if(dcb_output_du_delay_present_flag)	
pic_spt_dcb_output_du_delay	u(v)
}	

FIG 33

【 図 3 4 】

region_refresh_info(payloadSize) {	記述子
refreshed_region_flag	u(1)
}	

FIG 34

【 図 3 5 】

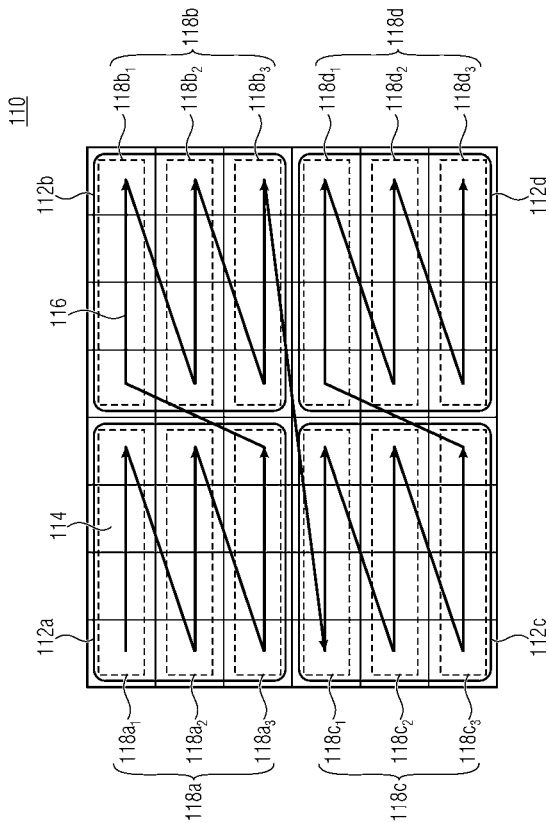


FIG 35

【 図 3 6 】

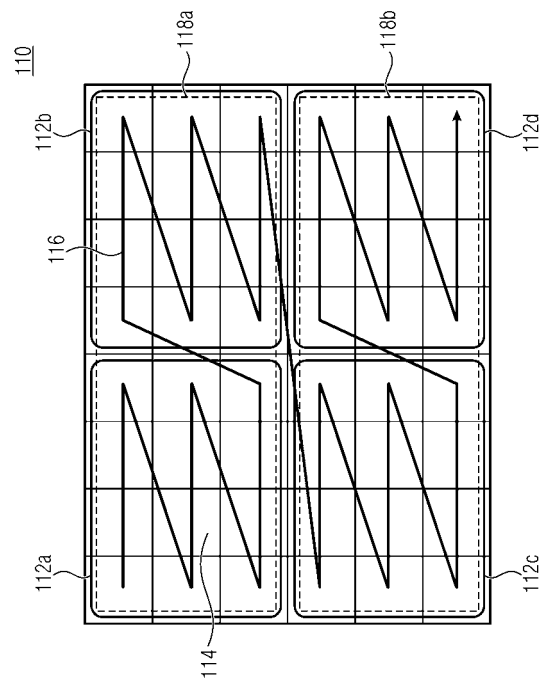


FIG 36

【 図 3 7 】

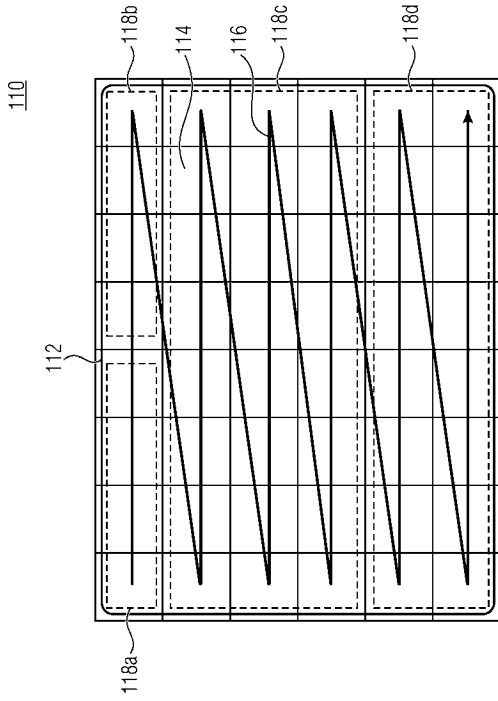


FIG 37

【 図 3 8 】

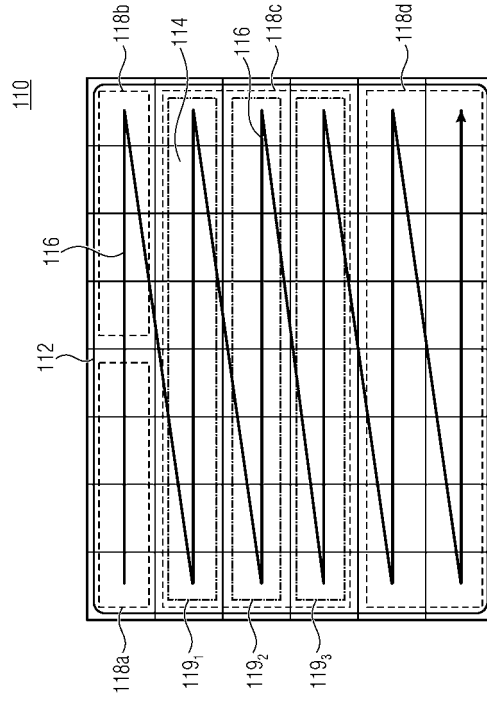


FIG 38

【 図 3 9 】

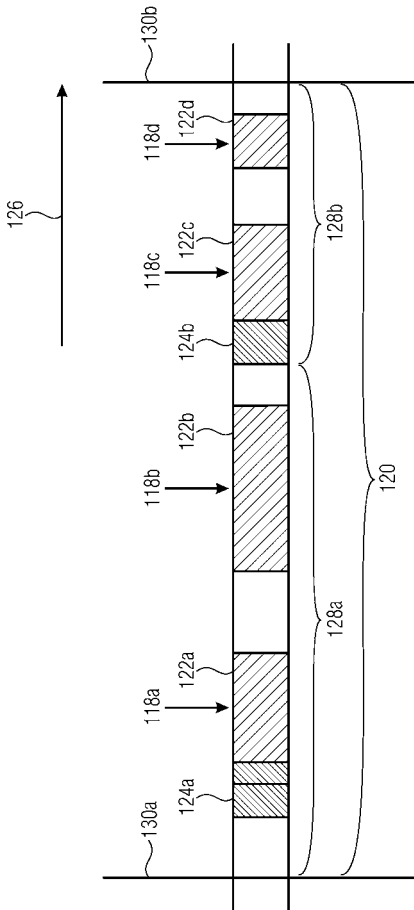


FIG 39

【手続補正書】

【提出日】平成30年12月3日(2018.12.3)

【手続補正1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項1】

ビデオ・データストリームにビデオ内容を、前記ビデオ内容の画像のサブポーション単位で符号化するためのエンコーダであって、前記エンコーダは、

各サブポーションを前記ビデオ・データストリームのバケットのシーケンスの1つ以上のペイロード・パケットに符号化し、前記パケットのシーケンスは、アクセスユニットのシーケンスに分割され、各アクセスユニットは前記ビデオ内容のそれぞれの画像に関連する前記ペイロード・パケットに関連付けられ、

タイミング制御パケットを、前記タイミング制御パケットがアクセスユニットのうちの少なくとも1つを複数の復号化ユニットに再分割するように、バケットのシーケンスに分散させるように構成され、各タイミング制御パケットは、複数の復号化ユニットのうちの対応する1つに関連付けられ、前記対応する復号化ユニットのためのデコーダバッファ検索時間をシグナリングし、その前記ペイロード・パケットは、前記パケットのシーケンスにおいて、それぞれの前記タイミング制御パケットに後続し、

前記パケットのシーケンスは異なるタイプのS E Iパケットを含み、

前記パケットのシーケンスはそれぞれ、前記各パケットのパケット・ヘッダにパケット・タイプフィールドを含み、前記パケット・タイプフィールドの内容は、前記ペイロード・パケットと前記タイミング制御パケットとの間で異なっていて、前記ペイロード・パケットと前記タイミング制御パケットが異なるパケット・タイプとなるようになっており、前記ペイロード・パケットは除去不可能なパケット・タイプであり、前記タイミング制御パケットは除去可能なパケット・タイプであり、

前記各S E IパケットのそれぞれはS E Iパケット・タイプフィールドを含み、タイミング制御パケットのS E Iパケット・タイプフィールドの内容は、別のS E Iパケット・タイプの内容とは異なる、エンコーダ。

【請求項2】

前記サブポーションが、スライスであり、各ペイロード・パケットが1つ以上のスライスを含む、請求項1に記載のエンコーダ。

【請求項3】

前記スライスは、個別に復号化可能なスライスと、W P P処理のために、スライス境界を越えたエントロピーおよび予測復号化を用いた復号化が可能な従属スライスとを含む、請求項2に記載のエンコーダ。

【請求項4】

前記パケットのシーケンスにおける複数の前記パケット・タイプのパケットの出現には所定の制限が課され、いずれかの除去可能なパケット・タイプが前記ビデオ・データストリームから除去されても、前記パケットのシーケンスの範囲内の同じ位置に止まり、前記制限は前記パケットが各アクセスユニット内で従うべき順序を規定し、前記制限が、制限されるインスタンスを検出することによって、前記制限を使用してアクセスユニットの境界を検出することができる、請求項1に記載のエンコーダ。

【請求項5】

前記エンコーダは、前記ビデオ内容の現在の画像を符号化する間に、

現在の画像の現在のサブポーションを現在の復号化ユニットの現在のペイロード・パケットに符号化し、

前記ビデオ・データストリームの範囲内で、現在のタイミング制御パケットを接頭辞と

する前記現在の復号化ユニットを送信すると共に、前記現在のタイミング制御パケットによってシグナリングされるデコーダバッファ検索時間を第1の時間的瞬間に設定し、

前記現在の画像の更なるサブポーションを、前記第1の時間的瞬間よりも遅い第2の時間的瞬間に符号化するように構成される、請求項1に記載のエンコーダ。

【請求項6】

ビデオ・データストリームにビデオ内容を、前記ビデオ内容の画像のサブポーション単位で符号化するための方法であって、前記方法は、

各サブポーションを前記ビデオ・データストリームのパケットのシーケンスの1つ以上のペイロード・パケットに符号化するステップであって、前記パケットのシーケンスはアクセスユニットのシーケンスに分割され、各アクセスユニットは前記ビデオ内容のそれぞれの画像に関連する前記ペイロード・パケットに関連付けられている、ステップと、

タイミング制御パケットを、前記タイミング制御パケットが前記アクセスユニットのうちの少なくとも1つを複数の復号化ユニットに再分割するように、パケットのシーケンスに分散させるステップであって、各タイミング制御パケットは、複数の復号化ユニットのうちの対応する1つに関連付けられ、前記対応する復号化ユニットのためのデコーダバッファ検索時間をシグナリングし、その前記ペイロード・パケットは、前記パケットのシーケンスにおいて、それぞれの前記タイミング制御パケットに後続するステップを含み、

前記パケットのシーケンスは異なるタイプのSEIパケットを含み、

前記パケットのシーケンスはそれぞれ、前記各パケットのパケット・ヘッダにパケット・タイプフィールドを含み、前記パケット・タイプフィールドの内容は、前記ペイロード・パケットと前記タイミング制御パケットとの間で異なっており、前記ペイロード・パケットと前記タイミング制御パケットが異なるパケット・タイプとなるようになっており、前記ペイロード・パケットは除去不可能なパケット・タイプであり、前記タイミング制御パケットは除去可能なパケット・タイプであり、

前記各SEIパケットのそれぞれはSEIパケット・タイプフィールドを含み、タイミング制御パケットのSEIパケット・タイプフィールドの内容は、別のSEIパケット・タイプの内容とは異なる、方法。

【請求項7】

ビデオ内容の画像のサブポーション単位で符号化されている前記ビデオ内容を含むビデオ・データストリームを復号化するためのデコーダであって、各サブポーションは前記ビデオ・データストリームのパケットのシーケンスの1つ以上のペイロード・パケットにそれぞれ符号化され、前記パケットのシーケンスはアクセスユニットのシーケンスに分割され、各アクセスユニットは前記ビデオ内容のそれぞれの画像に関連する前記ペイロード・パケットに関連付けられ、前記デコーダは、前記ビデオ・データストリームをバッファ処理するためのバッファまたは前記ビデオ・データストリームの復号化することでそこから取得した前記ビデオ内容の再構築を含み、前記デコーダは、

前記パケットのシーケンスに分散されているタイミング制御パケットを探し、

前記アクセスユニットのうちの少なくとも1つを前記探されたタイミング制御パケットに基づいて複数の復号化ユニットに再分割するように構成され、前記各タイミング制御パケットは、複数の復号化ユニットのうちの対応する1つに関連付けられ、前記対応する復号化ユニットのためのデコーダバッファ検索時間をシグナリングし、前記対応するタイミング制御パケットが示すデコーダバッファ検索時間に従って前記バッファから前記複数の復号化ユニットの前記ペイロード・パケットを検索し、

前記パケットのシーケンスは異なるタイプのSEIパケットを含み、

前記パケットのシーケンスはそれぞれ、前記各パケットのパケット・ヘッダにパケット・タイプフィールドを含み、前記パケット・タイプフィールドの内容は、前記ペイロード・パケットと前記タイミング制御パケットとの間で異なっており、前記ペイロード・パケットと前記タイミング制御パケットが異なるパケット・タイプとなるようになっており、前記ペイロード・パケットは除去不可能なパケット・タイプであり、前記タイミング制御パケットは除去可能なパケット・タイプであり、

前記各 S E I パケットのそれぞれは S E I パケット・タイプフィールドを含み、タイミング制御パケットの S E I パケット・タイプフィールドの内容は、別の S E I パケット・タイプの内容とは異なる、デコーダ。

【請求項 8】

前記デコーダは、前記タイミング制御パケットを探す際に、構文要素部を示すパケット・タイプを各パケット内で検査し、構文要素部を示すパケット・タイプの値が所定の値と等しければ、前記各パケットをタイミング制御パケットとして認めるように構成される、請求項 7 に記載のデコーダ。

【請求項 9】

ビデオ内容の画像のサブポーション単位で符号化された前記ビデオ内容を含むビデオ・データストリームを復号化するための方法であって、各サブポーションは前記ビデオ・データストリームのパケットのシーケンスの 1 つ以上のペイロード・パケットにそれぞれ符号化され、前記パケットのシーケンスはアクセスユニットのシーケンスに分割され、各アクセスユニットは前記ビデオ内容のそれぞれの画像に関連する前記ペイロード・パケットに関連付けられ、前記方法は、前記ビデオ・データストリームをバッファ処理するためのバッファまたは前記ビデオ・データストリームの復号化によりそこから取得した前記ビデオ内容の再構築を用い、前記方法は、

前記パケットのシーケンスに分散されているタイミング制御パケットを探すステップと

、
前記アクセスユニットのうちの少なくとも 1 つを前記探されたタイミング制御パケットに基づいて複数の復号化ユニットに再分割するステップであって、前記各タイミング制御パケットは複数の復号化ユニットのうちの対応する 1 つに関連付けられ、前記対応する復号化ユニットのためのデコーダバッファ検索時間をシグナリングするステップと、

前記対応するタイミング制御パケットが示すデコーダバッファ検索時間に従って前記バッファから前記複数の復号化ユニットの前記ペイロード・パケットを検索するステップを含み、

前記パケットのシーケンスは異なるタイプの S E I パケットを含み、

前記パケットのシーケンスはそれぞれ、前記各パケットのパケット・ヘッダにパケット・タイプフィールドを含み、前記パケット・タイプフィールドの内容は、前記ペイロード・パケットと前記タイミング制御パケットとの間で異なっていて、前記ペイロード・パケットと前記タイミング制御パケットが異なるパケット・タイプとなるようになっており、前記ペイロード・パケットは除去不可能なパケット・タイプであり、前記タイミング制御パケットは除去可能なパケット・タイプであり、

前記各 S E I パケットのそれぞれは S E I パケット・タイプフィールドを含み、タイミング制御パケットの S E I パケット・タイプフィールドの内容は、別の S E I パケット・タイプの内容とは異なる、方法。

【請求項 10】

ビデオ内容の画像のサブポーション単位で符号化された前記ビデオ内容を含むビデオ・データストリームを伝送するための方法であって、各サブポーションはビデオ・データストリームのパケットのシーケンスの 1 つ以上のペイロード・パケットにそれぞれ符号化され、前記パケットのシーケンスはデコーダで復号化するためのアクセスユニットのシーケンスに分割され、各アクセスユニットは前記ビデオ内容のそれぞれの画像に関連する前記ペイロード・パケットに関連付けられ、前記方法は、

前記パケットのシーケンスに分散されているタイミング制御パケットを探すステップと

、
前記アクセスユニットのうちの少なくとも 1 つを前記探されたタイミング制御パケットに基づいて複数の復号化ユニットに再分割するステップであって、前記各タイミング制御パケットは複数の復号化ユニットのうちの対応する 1 つに関連付けられ、前記対応する復号化ユニットのためのデコーダバッファ検索時間をシグナリングするステップと、

各タイミング制御パケットから、前記対応する復号化ユニットのための前記デコーダバ

ッファ検索時間を引き出すステップであって、そのペイロード・パケットは前記パケットのシーケンスにおいて前記各タイミング制御パケットに後続する、ステップと、

前記復号化ユニットのデコーダバッファ検索時間に依存して前記ビデオ・データストリームの伝送を実行するステップを含み、

前記パケットのシーケンスは異なるタイプの S E I パケットを含み、

前記パケットのシーケンスはそれぞれ、前記各パケットのパケット・ヘッダにパケット・タイプフィールドを含み、前記パケット・タイプフィールドの内容は、前記ペイロード・パケットと前記タイミング制御パケットとの間で異なっており、前記ペイロード・パケットと前記タイミング制御パケットが異なるパケット・タイプとなるようになっており、前記ペイロード・パケットは除去不可能なパケット・タイプであり、前記タイミング制御パケットは除去可能なパケット・タイプであり、

前記各 S E I パケットのそれぞれは S E I パケット・タイプフィールドを含み、タイミング制御パケットの S E I パケット・タイプフィールドの内容は、別の S E I パケット・タイプの内容と異なる、方法。

【請求項 1 1】

コンピュータ上で動作するときに、請求項 6 に記載の方法を実行するためのプログラムコードを含むコンピュータプログラムを記憶した、非一時的コンピュータ可読媒体。

【請求項 1 2】

コンピュータ上で動作するときに、請求項 9 に記載の方法を実行するためのプログラムコードを含むコンピュータプログラムを記憶した、非一時的コンピュータ可読媒体。

【請求項 1 3】

コンピュータ上で動作するときに、請求項 1 0 に記載の方法を実行するためのプログラムコードを含むコンピュータプログラムを記憶した、非一時的コンピュータ可読媒体。

フロントページの続き

(72)発明者 ヘンケル アナスタージア
ドイツ連邦共和国 1 0 2 4 5 ベルリン ファレンハイトシュトラッセ 8

(72)発明者 マルベ デトレフ
ドイツ連邦共和国 1 2 1 6 1 ベルリン ズュートヴェストコルゾ 7 0

(72)発明者 グルーエネベルク カルステン
ドイツ連邦共和国 1 3 5 9 9 ベルリン アディッケスシュトラッセ 4 3

(72)発明者 スクーピン ローベルト
ドイツ連邦共和国 1 0 5 5 5 ベルリン シュレースヴィガー ウーファー 5

Fターム(参考) 5C159 KK13 LC09 MA00 ME01 RBO2 RB09 RB16 RC04 RC11 SS08
TC34 UA02 UA05