(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5837217号
(P5837217)

(45)発行日　平成27年12月24日(2015.12.24)　(24)登録日　平成27年11月13日(2015.11.13)

(51)Int.Cl.		F I	
H04N 19/70	(2014.01)	H04N	19/70
H04N 19/91	(2014.01)	H04N	19/91
H04N 19/436	(2014.01)	H04N	19/436
H04N 19/96	(2014.01)	H04N	19/96
H04N 19/46	(2014.01)	H04N	19/46

請求項の数 38　(全 59 頁)

(21)出願番号	特願2014-540122 (P2014-540122)
(86)(22)出願日	平成24年11月2日 (2012.11.2)
(65)公表番号	特表2014-535223 (P2014-535223A)
(43)公表日	平成26年12月25日 (2014.12.25)
(86)国際出願番号	PCT/US2012/063265
(87)国際公開番号	WO2013/067311
(87)国際公開日	平成25年5月10日 (2013.5.10)
審査請求日	平成26年7月4日 (2014.7.4)
(31)優先権主張番号	61/555,932
(32)優先日	平成23年11月4日 (2011.11.4)
(33)優先権主張国	米国 (US)
(31)優先権主張番号	61/557,259
(32)優先日	平成23年11月8日 (2011.11.8)
(33)優先権主張国	米国 (US)

(73)特許権者　595020643
クゥアルコム・インコーポレイテッド
ＱＵＡＬＣＯＭＭ　ＩＮＣＯＲＰＯＲＡＴＥＤ
アメリカ合衆国、カリフォルニア州　９２１２１－１７１４、サン・ディエゴ、モアハウス・ドライブ　５７７５
(74)代理人　100108855
弁理士　蔵田　昌俊
(74)代理人　100109830
弁理士　福原　淑弘
(74)代理人　100103034
弁理士　野河　信久
(74)代理人　100075672
弁理士　峰　隆司

最終頁に続く

(54)【発明の名称】複数の符号化ピクチャ区画を含むネットワーク抽象化レイヤユニットによるビデオコーディング

(57)【特許請求の範囲】

【請求項１】

ビデオデータを符号化する方法であって、

第１の構文要素と第２の構文要素とをエントロピー符号化することと、

一連のオフセット構文要素を生成することであって、前記オフセット構文要素のそれぞれの長さが前記第１の構文要素に基づいて判断可能であり、前記一連のオフセット構文要素内のオフセット構文要素の数が前記第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能であり、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記生成することと、

前記ビデオデータのコード化表現を形成するビットストリームを生成することであって、前記ビットストリームは、エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含む、前記生成することと

を備える方法。

【請求項２】

前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す請求項１に記載の方法。

【請求項３】
前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含む補助強化情報（ＳＥＩ）メッセージを生成することをさらに備える請求項１に記載の方法。
【請求項４】
前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む請求項３に記載の方法。
【請求項５】
前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを生成することをさらに備える請求項４に記載の方法。
【請求項６】
前記オフセット構文要素のそれぞれの長さは前記第１の構文要素の値に１０を加えた値に等しい請求項１に記載の方法。
【請求項７】
ビデオデータを記憶するように構成された記憶媒体と、
第１の構文要素と第２の構文要素とをエントロピー符号化し、
一連のオフセット構文要素を生成することであって、前記オフセット構文要素のそれぞれの長さが前記第１の構文要素に基づいて判断可能であり、前記一連のオフセット構文要素内のオフセット構文要素の数が前記第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能であり、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記生成し、
前記ビデオデータのコード化表現を形成するビットストリームを生成することであって、前記ビットストリームは、エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含む、前記生成する
ように構成された１つまたは複数のプロセッサと
を備えるコンピューティングデバイス。
【請求項８】
前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す請求項７に記載のコンピューティングデバイス。
【請求項９】
前記１つまたは複数のプロセッサは、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含む補助強化情報（ＳＥＩ）メッセージを生成するように構成された請求項７に記載のコンピューティングデバイス。
【請求項１０】
前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む請求項９に記載のコンピューティングデバイス。
【請求項１１】
前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを生成することをさらに備える請求項７に記載のコンピューティングデバイス。
【請求項１２】
前記オフセット構文要素のそれぞれの長さは前記第１の構文要素の値に１０を加えた値に等しい請求項７に記載のコンピューティングデバイス。
【請求項１３】
前記コンピューティングデバイスはビデオエンコーダを備える請求項７に記載のコンピ

ューティングデバイス。
【請求項１４】
第１の構文要素と第２の構文要素とをエントロピー符号化するための手段と、
一連のオフセット構文要素を生成するための手段であって、前記オフセット構文要素のそれぞれの長さが前記第１の構文要素に基づいて判断可能であり、前記一連のオフセット構文要素内のオフセット構文要素の数が前記第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能であり、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記生成するための手段と、
ビデオデータのコード化表現を形成するビットストリームを生成するための手段であって、前記ビットストリームは、エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含む、前記生成するための手段と
を備えるコンピューティングデバイス。
【請求項１５】
コンピューティングデバイスの１つまたは複数のプロセッサによって実行されると、前記コンピューティングデバイスを、
第１の構文要素と第２の構文要素とをエントロピー符号化し、
一連のオフセット構文要素を生成することであって、前記オフセット構文要素のそれぞれの長さが前記第１の構文要素に基づいて判断可能であり、前記一連のオフセット構文要素内のオフセット構文要素の数が前記第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能であり、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記生成し、
ビデオデータのコード化表現を形成するビットストリームを生成することであって、前記ビットストリームは、エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含む、前記生成する
ように構成する命令を記憶したコンピュータ可読記憶媒体。
【請求項１６】
ビデオデータを復号する方法であって、
前記ビデオデータのコード化表現を形成するビットストリームを受信することであって、前記ビットストリームは、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連の固定長オフセット構文要素とを含む、前記受信することと、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号することと、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断することと、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断することと、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析することと、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断することであって、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記判断することと、

前記符号化ピクチャ区画を復号することと
を備える方法。
【請求項１７】
前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す請求項１６に記載の方法。
【請求項１８】
前記ビットストリームは、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含む補助強化情報（ＳＥＩ）メッセージを含む請求項１６に記載の方法。
【請求項１９】
前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む請求項１８に記載の方法。
【請求項２０】
前記ビットストリームは、前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを含む請求項１６に記載の方法。
【請求項２１】
前記ＳＥＩ　ＮＡＬユニットは、前記ビットストリーム内で、前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にあり、
前記方法は、前記ビットストリーム内で前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にある前記ＳＥＩ　ＮＡＬユニットに少なくとも部分的に基づいて、前記ＳＥＩメッセージが前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットに適用可能であると判断することをさらに備える請求項２０に記載の方法。
【請求項２２】
前記符号化ピクチャ区画の並列復号のために前記符号化ピクチャ区画を異なる処理コアに供給することと、前記処理コアにおいてそれぞれの前記符号化ピクチャ区画を復号することとをさらに備える請求項１６に記載の方法。
【請求項２３】
前記オフセット構文要素のそれぞれの長さは前記第１の構文要素の値に１０を加えた値に等しい請求項１６に記載の方法。
【請求項２４】
ビデオデータを記憶するように構成されたデータ記憶媒体と、
前記ビデオデータのコード化表現を形成するビットストリームを受信することであって、前記ビットストリームは、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含む、前記受信し、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号し、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断し、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断し、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析し、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断することであって、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記判断し、
前記符号化ピクチャ区画を復号する

ように構成された１つまたは複数のプロセッサと
を備えるコンピューティングデバイス。
【請求項２５】
前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す請求項２４に記載のコンピューティングデバイス。
【請求項２６】
前記ビットストリームは、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含む補助強化情報（ＳＥＩ）メッセージを含む請求項２４に記載のコンピューティングデバイス。
【請求項２７】
前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む請求項２６に記載のコンピューティングデバイス。
【請求項２８】
前記ビットストリームは、前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを含む請求項２４に記載のコンピューティングデバイス。
【請求項２９】
前記ＳＥＩ　ＮＡＬユニットは、前記ビットストリーム内で、前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にあり、
前記１つまたは複数のプロセッサは、前記ビットストリーム内で前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にある前記ＳＥＩ　ＮＡＬユニットに少なくとも部分的に基づいて、前記ＳＥＩメッセージが前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットに適用可能であると判断するように構成された請求項２８に記載のコンピューティングデバイス。
【請求項３０】
前記１つまたは複数のプロセッサは、前記符号化ピクチャ区画の並列復号のために前記符号化ピクチャ区画を異なる処理コアに供給するように構成された請求項２４に記載のコンピューティングデバイス。
【請求項３１】
前記コンピューティングデバイスはビデオデコーダを備える請求項２４に記載のコンピューティングデバイス。
【請求項３２】
前記第１の構文要素の値に１０を加えた値が前記一連のオフセット構文要素のビット数で表した長さに等しい請求項２４に記載のコンピューティングデバイス。
【請求項３３】
ビデオデータのコード化表現を形成するビットストリームを受信するための手段であって、前記ビットストリームは、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含む、前記受信するための手段と、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号するための手段と、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断するための手段と、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断するための手段と、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析するための手段と、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断するための手段であって、前記符号化ピクチャ区

画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記判断するための手段と、

前記符号化ピクチャ区画を復号するための手段と

を備えるコンピューティングデバイス。

【請求項３４】

コンピューティングデバイスの１つまたは複数のプロセッサによって実行されると、前記コンピューティングデバイスを、

ビデオデータのコード化表現を形成するビットストリームを受信することであって、前記ビットストリームは、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含む、前記受信し、

前記第１の構文要素と前記第２の構文要素とをエントロピー復号し、

前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断し、

前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断し、

前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析し、

前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断することであって、前記符号化ピクチャ区画の各々は、それぞれのエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのそれぞれのグループを含む、前記判断し、

前記符号化ピクチャ区画を復号する

ように構成する命令を記憶したコンピュータ可読記憶媒体。

【請求項３５】

前記コンピューティングデバイスは、

集積回路、

マイクロプロセッサ、または

ワイヤレスハンドセット、

のうちの少なくとも１つを備える、請求項７に記載のコンピューティングデバイス。

【請求項３６】

前記コンピューティングデバイスは、

集積回路、

マイクロプロセッサ、または

ワイヤレスハンドセット、

のうちの少なくとも１つを備える、請求項２４に記載のコンピューティングデバイス。

【請求項３７】

前記ビデオデータをキャプチャするように構成されたカメラをさらに備える、請求項７に記載のコンピューティングデバイス。

【請求項３８】

前記ビデオデータを表示するように構成されたディスプレイをさらに備える、請求項２４に記載のコンピューティングデバイス。

【発明の詳細な説明】

【技術分野】

【０００１】

本出願は、２０１１年１１月４日出願の米国仮特許出願第６１／５５５９３２号および２０１１年１１月８日出願の米国仮特許第６１／５５７２５９号の優先権を主張し、両出願の全内容が参考として本明細書に援用される。

【０００２】
本開示はビデオコーディング（すなわちビデオデータの符号化および／または復号）に関する。
【背景技術】
【０００３】
デジタルテレビ、デジタル衛星放送システム、ワイヤレス放送システム、携帯情報端末（ＰＤＡ）、ラップトップまたはデスクトップコンピュータ、タブレットコンピュータ、電子書籍リーダー、デジタルカメラ、デジタル記録デバイス、デジタルメディアプレイヤー、ビデオゲームデバイス、ビデゲームコンソール、セルラーまたは衛星無線電話、いわゆる「スマートフォン」、ビデオ電話会議デバイス、ビデオストリーミングデバイスなどを含む様々なデバイスに、デジタルビデオ機能を組み込むことができる。デジタルビデオデバイスは、ＭＰＥＧ－２、ＭＥＰＥＧ－４、ＩＴＵ－Ｔ Ｈ．２６３、ＩＴＵ－Ｔ Ｈ．２６４／ＭＰＥＧ－４、パート１０、アドバンストビデオコーディング（ＡＶＣ）、現在策定中の高効率ビデオコーディング（ＨＥＶＣ）標準、およびこれらの標準の拡張版によって規定される標準に記載されているものなどのビデオ圧縮技術を実装する。ビデオデバイスは、このようなビデオ圧縮技術を実装することにより、デジタルビデオ情報の送信、受信、符号化、復号および／または格納をより効率的に行うことができる。
【０００４】
ビデオ圧縮技術は、空間（ピクチャ内）予測および／または時間（ピクチャ間）予測を行って、ビデオシーケンスに固有の冗長性を低減または除去する。ブロックベースのビデオコーディングの場合、ビデオスライス（すなわちビデオフレームまたはビデオフレームの一部）をビデオブロックに区画化することができ、ビデオブロックはツリーブロック、コーディングユニット（ＣＵ）および／またはコーディングノードとも呼ばれることがある。ピクチャのイントラコード化（Ｉ）スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルを基準にして空間予測を使用して符号化される。ピクチャのインターコード化（ＰまたはＢ）スライス内のビデオブロックは、同じピクチャ内の隣接ブロック内の参照サンプルを基準にした空間予測、または他の参照ピクチャ内の参照サンプルを基準にした時間予測を使用することができる。ピクチャはフレームとも呼ばれることがあり、参照ピクチャは参照フレームとも呼ばれることがある。
【０００５】
空間または時間予測の結果、コーディングするブロックの予測ブロックが得られる。残差データは、コーディングする元のブロックと予測ブロックとの間の画素差を表す。インターコード化ブロックは、予測ブロックを形成する参照サンプルのブロックを指す動きベクトルと、コード化ブロックと予測ブロックとの差を示す残差データとに従って符号化される。イントラコード化ブロックは、イントラコーディングモードと残差データとに従って符号化される。さらに圧縮するために、残差データは画素領域から変換領域に変換可能であり、その結果、残差係数が得られ、この残差係数は量子化され得る。量子化された係数は、最初は２次元配列に配列されており、係数の１次元ベクトルを生成するためにスキャンされ、エントロピーコーディングを適用してさらなる圧縮が実現され得る。
【発明の概要】
【０００６】
一般に、ビデオエンコーダは、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームを生成する。各オフセット構文要素の長さは、第１の構文要素に基づいて判断可能である。一連のオフセット構文要素中のオフセット構文要素の数は、第２の構文要素に基づいて判断可能である。ネットワーク抽象化レイヤ（ｎｅｔｗｏｒｋ ａｂｓｔｒａｃｔｉｏｎ ｌａｙｅｒ：ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置は、オフセット構文要素に基づいて判断可能である。ビデオデコーダがビットストリームを受信し、第１および第２の構文要素をエントロピー復号し、第１および第２の構文要素を使用してオフセット構文要素を構文解析する。ビデオデコーダ

はオフセット構文要素を使用して、ＮＡＬユニット内での符号化ピクチャ区画の位置を判断する。ビデオデコーダは、これらの符号化画素区画を復号する。

【０００７】

一態様では、本開示は第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とのエントロピー符号化を備える方法について記載する。各オフセット構文要素の長さは、第１の構文要素に基づいて判断可能である。一連のオフセット構文要素内のオフセット構文要素の数は、第２の構文要素に基づいて判断可能である。ＮＡＬユニット内の複数の符号化ピクチャ区画の位置は、オフセット構文要素に基づいて判断可能である。この方法は、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームの生成も備える。

【０００８】

他の態様では、本開示は、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とをエントロピー符号化するように構成された１つまたは複数のプロセッサを備えるコンピューティングデバイスについて記載する。各オフセット構文要素の長さは、第１の構文要素に基づいて判断可能である。一連のオフセット構文要素中のオフセット構文要素の数は、第２の構文要素に基づいて判断可能である。ＮＡＬユニット内の複数の符号化ピクチャ区画の位置は、オフセット構文要素に基づいて判断可能である。１つまたは複数のプロセッサは、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームを生成するようにも構成される。

【０００９】

他の態様では、本開示は、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とをエントロピー符号化するための手段を備えるコンピューティングデバイスについて記載する。各オフセット構文要素の長さは、第１の構文要素に基づいて判断可能である。一連のオフセット構文要素内のオフセット構文要素の数は、第２の構文要素に基づいて判断可能である。ＮＡＬユニット内の複数の符号化ピクチャ区画の位置は、オフセット構文要素に基づいて判断可能である。このコンピューティングデバイスは、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームを生成するための手段も備える。

【００１０】

他の態様では、本開示は、コンピューティングデバイスの１つまたは複数のプロセッサによって実行されると、コンピューティングデバイスを、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とをエントロピー符号化するように構成する命令を記憶するコンピュータ可読記憶媒体について記載する。各オフセット構文要素の長さは、第１の構文要素に基づいて判断可能である。一連のオフセット構文要素内のオフセット構文要素の数は、第２の構文要素に基づいて判断可能である。ＮＡＬユニット内の複数の符号化ピクチャ区画の位置は、オフセット構文要素に基づいて判断可能である。さらに、命令は、コンピューティングデバイスを、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームを生成するように構成する。

【００１１】

他の態様では、本開示は、ビデオデータを復号する方法を記載する。この方法は、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含むビットストリームを受信することを備える。また、この方法は、第１の構文要素と第２の構文要素とをエントロピー復号することも備える。さらに、この方法は、第１の構文要素に基づいて各オフセット構文要素の長さを判断することを備える。またこの方法は、第２の構文要素に基づいて一連のオフセット構文要素内のオフセット構文要素の数を判断することも備える。さらに

この方法は、各オフセット構文要素の長さと一連のオフセット構文要素内のオフセット構文要素の数とに少なくとも部分的に基づいて、オフセット構文要素を構文解析することを備える。さらにこの方法は、オフセット構文要素に基づいて、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置を求めることを備える。またこの方法は、符号化ピクチャ区画を復号することを備える。

【００１２】

他の態様では、本開示は、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含むビットストリームを受信するように構成された１つまたは複数のプロセッサを備えるコンピューティングデバイスについて記載する。１つまたは複数のプロセッサは、第１の構文要素と第２の構文要素とをエントロピー復号するようにも構成される。１つまたは複数のプロセッサは、第１の構文要素に基づいて各オフセット構文要素の長さを判断するようにも構成される。さらに、１つまたは複数のプロセッサは、第２の構文要素に基づいて、一連のオフセット構文要素内のオフセット構文要素の数を判断するように構成される。１つまたは複数のプロセッサは、各オフセット構文要素の長さと一連のオフセット構文要素内のオフセット構文要素の数とに少なくとも部分的に基づいて、オフセット構文要素を構文解析するようにも構成される。さらに、１つまたは複数のプロセッサは、オフセット構文要素に基づいてＮＡＬユニット内の複数の符号化ピクチャ区画の位置を判断するように構成される。１つまたは複数のプロセッサは、符号化ピクチャ区画を復号するようにも構成される。

【００１３】

他の態様では、本開示は、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含むビットストリームを受信するための手段を備えるコンピューティングデバイスについて記載する。このコンピューティングデバイスは、第１の構文要素と第２の構文要素とをエントロピー復号するための手段も備える。さらに、このコンピューティングデバイスは、第１の構文要素に基づいて各オフセット構文要素の長さを判断するための手段も備える。コンピューティングデバイスは、第２の構文要素に基づいて一連のオフセット構文要素内のオフセット構文要素の数を判断するための手段も備える。さらに、コンピューティングデバイスは、各オフセット構文要素の長さと一連のオフセット構文要素内のオフセット構文要素の数とに少なくとも部分的に基づいて、オフセット構文要素を構文解析するための手段を備える。さらに、コンピューティングデバイスは、オフセット構文要素に基づいて、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置を求めるための手段も備える。コンピューティングデバイスは、符号化ピクチャ区画を復号するための手段も備える。

【００１４】

他の態様では、本開示は、コンピューティングデバイスの１つまたは複数のプロセッサによって実行されると、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含むビットストリームを受信するようにコンピューティングデバイスを構成する命令を記憶するコンピュータ可読記憶媒体について記載する。さらに命令は、コンピューティングデバイスに第１の構文要素と第２の構文要素とをエントロピー復号させる。また命令は、第１の構文要素に基づいて各オフセット構文要素の長さを判断するようにコンピューティングデバイスを構成する。さらに命令は、コンピューティングデバイスに第２の構文要素に基づいて一連のオフセット構文要素内のオフセット構文要素の数を判断させる。さらに、命令はコンピューティングデバイスに、各オフセット構文要素の長さと一連のオフセット構文要素内のオフセット構文要素の数とに少なくとも部分的に基づいてオフセット構文要素を構文解析させる。また命令は、コンピューティングデバイスに、オフセット構文要素に基づいてＮＡＬユニット内の複数の符号化ピクチャ区画の位置を判断させる。さらに命令は、コンピューティングデバイスに符号化ピクチャ区画を復号させる。

【００１５】
本開示の１つまたは複数の例の詳細を添付図面と以下の説明とに記載する。その他の特徴、目的および利点は、以下の説明、図面および特許請求の範囲から明らかになるであろう。
【図面の簡単な説明】
【００１６】
【図１】本開示に記載の技術を使用可能な例示のビデオコーディングシステムを示すブロック図。
【図２】本開示に記載の技術を実装可能な例示のビデオエンコーダを示すブロック図。
【図３】本開示に記載の技術を実装可能な例示のビデオデコーダを示すブロック図。
【図４】本開示の１つまたは複数の技術によるビデオエンコーダの例示の動作を示すフローチャート。
【図５】本開示の１つまたは複数の技術によるビデオデコーダの例示の動作を示すフローチャート。
【図６】本開示の１つまたは複数の技術による、バイト整列スライスヘッダーを有するコード化スライスＮＡＬユニットを生成するためのビデオエンコーダの例示の動作を示すフローチャート。
【図７】本開示の１つまたは複数の技術による、バイト整列スライスヘッダーを生成するビデオエンコーダの例示の動作を示すフローチャート。
【図８】本開示の１つまたは複数の技術による、ビデオデコーダがバイト整列スライスヘッダーを有するコード化スライスＮＡＬユニットを受信するビデオデコーダの例示の動作を示すフローチャート。
【図９】波面並列処理を示す概念図。
【図１０】ピクチャを複数のタイルに区画化する場合の例示のコーディング順序を示す概念図。
【図１１】本開示の１つまたは複数の態様による、ＮＡＬユニット内の符号化ピクチャ区画のオフセットに関連付けられた構文要素を表す可変長値を含むビットストリームを生成する、ビデオエンコーダの例示の動作を示すフローチャート。
【図１２】本開示の１つまたは複数の態様による、ＮＡＬユニット内の符号化ピクチャ区画のオフセットに関連付けられた構文要素を表す可変長値を含むビットストリームによって表されたビデオデータを復号するビデオデコーダの例示の動作を示すフローチャート。
【発明を実施するための形態】
【００１７】
ピクチャは複数のツリーブロックを含む。ツリーブロックは、ピクチャ内の２次元画素ブロックに関連付けられる。ビデオエンコーダがピクチャを複数のピクチャ区画に分割する。例えば、ビデオエンコーダは、ピクチャをエントロピースライス、タイル、または波面並列処理（ｗａｖｅｆｒｏｎｔ　ｐａｒａｌｌｅｌ　ｐｒｏｃｅｓｓｉｎｇ：ＷＰＰ）波に分割することができる。言い換えると、本開示では「ピクチャ区画」という用語を使用してエントロピースライス、タイル、またはＷＰＰ波を一般的に指す場合がある。ピクチャ区画は、ピクチャのツリーブロックの重なり合わないサブセットに関連付けられる。例えば、ピクチャの各ツリーブロックは、ピクチャ区画のうちの厳密に１つと関連付けられ得る。ビデオエンコーダは、スライスに関連付けられた１つまたは複数の符号化ピクチャ区画を含むコード化スライスネットワーク抽象化レイヤ（ＮＡＬ）ユニットを生成することができる。
【００１８】
ある例では、ビデオデコーダが、スライスの符号化ピクチャ区画を並列して復号することができる。スライスの符号化ピクチャ区画を並列して復号するために、ビデオエンコーダは、コード化スライスＮＡＬユニット内で符号化ピクチャ区画の入口点を通知してもよい。符号化ピクチャ区画の入口点は、符号化ピクチャ区画の第１のビットの位置とすることができる。ビデオデコーダは、符号化ピクチャ区画の入口点を判断することができるた

め、ビデオデコーダは、並列復号のためにデコーダの異なる処理コアに異なる符号化ピクチャ区画を割り当てることが可能であってもよい。

【００１９】

タイルの入口点を通知するために、ビデオエンコーダは一連のオフセット構文要素を生成することができる。オフセット構文要素のそれぞれは、タイル中のバイト数を示すことができる。さらに、ビデオエンコーダは、２つの追加構文要素を生成してもよい。スライス内のタイルの数は、第１の追加構文要素に基づいて判断され得る。各オフセット構文要素内で使用されるビットの数は、第２の追加構文要素に基づいて判断され得る。既存の実装では、追加構文要素のそれぞれは常に５ビットの符号なし整数を使用するビットストリームで表される。

【００２０】

追加構文要素を常に５ビットの符号なし整数として表すことで、ビットストリーム中のビットの数が無用に増える可能性がある。さらに、追加構文要素を常に５ビットの符号なし整数で表すことにより、オフセット構文要素の数が３２個に制限される可能性がある。本開示の１つまたは複数の技術によると、ビデオエンコーダがこれらの追加構文要素をエントロピー符号化することができる（例えば、ｕｅ（ｖ）、左ビットが最初の符号なし整数０次指数ゴロムコード化（Ｅｘｐ－Ｇｏｌｕｌｍｎｂ）構文要素を使用することができる）。ビデオエンコーダが追加構文要素をエントロピー符号化するため、追加構文要素は可変長ビット列を使用するビットストリームで表される。多くの場合、これらの可変長ビット列は、５ビットより少ない符号なし整数を使用して追加構文要素を表すことができる。さらに、これらの追加構文要素のエントロピーコーディングにより、ビデオエンコーダは３２個を超えるオフセット構文要素を通知することができる。

【００２１】

場合によっては、特定のビデオデコーダを、符号化ピクチャ区画を並列して復号するように構成しなくてもよい。すなわち、ビデオデコーダによっては、並列処理に関して異なる機能を有してもよい。そのような場合、符号化ピクチャ区画の入口点を通知するために使用するビットはビデオデコーダにとっては用途がない。

【００２２】

本開示の技術によれば、ビデオエンコーダは、ビデオデータの複数の符号化ピクチャ区画を含むＶＣＬ　ＮＡＬユニットを生成することができる。さらに、ビデオエンコーダは、ＮＡＬユニット内の符号化ピクチャ区画の入口点を示す補助強化情報（Ｓｕｐｐｌｅｍｅｎｔａｌ　Ｅｎｈａｎｃｅｍｅｎｔ　Ｉｎｆｏｒｍａｔｉｏｎ：ＳＥＩ）メッセージを生成してもよい。説明の便宜上、本開示ではコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージを「入口点ＳＥＩメッセージ」と呼ぶことがある。入口点ＳＥＩメッセージは、例えば、ＳＥＩ型を有する非ＶＣＬ　ＮＡＬユニットなど、非ＶＣＬ　ＮＡＬユニットで提供し得る。

【００２３】

特定のビデオデコーダの構成を示すデータを使用してメディアアウェアネットワーク要素（ｍｅｄｉａ－ａｗａｒｅ　ｎｅｔｗｏｒｋ　ｅｌｅｍｅｎｔ：ＭＡＮＥ）を構成してもよい。このようなデータに基づいて、ビデオデコーダが符号化ピクチャ区画を並列して復号するように構成されている場合にのみ、入り口的ＳＥＩメッセージをビデオデコーダに送信するようにＭＡＮＥを構成することができる。このＭＡＮＥはビデオデコーダが使用するように構成されていないデータは送信しないため、帯域幅を節約することができる。さらに、入口点がＳＥＩメッセージで通知されるため、コード化スライスＮＡＬユニットがコード化ピクチャ区画の位置を示すデータを含んでいるか否かにかかわらず、コード化スライスＮＡＬユニットで通知する必要がない場合がある。これは、コード化スライスＮＡＬユニット内のビット数を削減する効果があり、それによって帯域幅がさらに節約される。

【００２４】

添付図面に、いくつか例を示す。添付図面において参照番号で示す要素は、以下の説明

で同様の参照番号で示す要素に対応する。本開示では、序数語（例えば「第１」、「第２」、「第３」など）で始まる名称を有する要素は、それらの要素に特定の順序があることを必ずしも意味しない。そのような序数語は、同じかまたは同様の種類の、異なる要素を指すために使用するに過ぎない。

【００２５】

図１は、本開示の技術を使用することができる例示のビデオコーディングシステム１０を示すブロック図である。本明細書では、「ビデオコーダ」という用語はビデオエンコーダとビデオデコーダの両方を総称するために使用する。本開示では、「ビデオコーディング」または「コーディング」という用語は、ビデオ符号化またはビデオ復号の総称である場合がある。

【００２６】

図１に示すようにビデオコーディングシステム１０は、送信元デバイス１２と送信先デバイス１４とを含む。送信元デバイス１２は符号化ビデオデータを生成する。したがって、送信元デバイス１２をビデオ符号化デバイスまたはビデオ符号化装置と呼ぶこともある。送信先デバイス１４は送信元デバイス１２によって生成された符号化ビデオデータを復号することができる。したがって、送信先デバイス１４をビデオ復号デバイスまたはビデオ復号装置と呼ぶこともある。送信元デバイス１２と送信先デバイス１４とは、ビデオコーディングデバイスまたはビデオコーディング装置の一例とすることができる。

【００２７】

送信元デバイス１２と送信先デバイス１４は、デスクトップコンピュータ、モバイルコンピューティングデバイス、ノートブック（例えばラップトップ）コンピュータ、タブレットコンピュータ、セットトップボックス、いわゆる「スマート」フォンなどの電話ハンドセット、テレビ、カメラ、表示デバイス、デジタルメディアプレイヤー、ビデオゲーム機、車載コンピュータなど、様々なデバイスからなり得る。

【００２８】

送信先デバイス１４は、チャネル１６を介して送信元デバイス１２から符号化ビデオデータを受け取ることができる。チャネル１６は、送信元デバイス１２から送信先デバイス１４に符号化ビデオデータを移動することが可能な種類の媒体またはデバイスを備え得る。一例では、チャネル１６は、送信デバイス１２が符号化ビデオデータを送信先デバイス１４にリアルタイムで直接送信することができるようにする、１つまたは複数の通信媒体を備え得る。この例では、送信元デバイス１２は、ワイヤレス通信プロトコルなどの通信標準に従って符号化ビデオデータを変調することができ、変調されたビデオデータを送信先デバイス１４に送信することができる。１つまたは複数の通信媒体は、無線周波数（ＲＦ）スペクトルまたは１つまたは複数の物理伝送線などの、ワイヤレスおよび／または有線通信媒体を含み得る。１つまたは複数の通信媒体は、ローカルエリアネットワーク、ワイドエリアネットワーク、またはグローバルネットワーク（例えばインターネット）などのパケット式ネットワークの一部を形成することができる。１つまたは複数の通信媒体は、ルータ、交換機、基地局、または、送信元デバイス１２から送信先デバイス１４への通信を容易にするその他の機器を含み得る。

【００２９】

他の例では、チャネル１６は、送信元デバイス１２によって生成された符号化ビデオデータを記憶する記憶媒体を含み得る。この例では、送信先デバイス１４はディスクアクセスまたはカードアクセスによって記憶媒体にアクセスすることができる。記憶媒体は、Ｂｌｕ－ｒａｙ（登録商標）ディスク、ＤＶＤ、ＣＤ－ＲＯＭ、フラッシュメモリ、または符号化ビデオデータを記憶するためのその他の適切なデジタル記憶媒体など、様々なローカルアクセスデータ記憶媒体を含み得る。

【００３０】

他の例では、チャネル１６は、送信元デバイス１２によって生成された符号化ビデオを記憶するファイルサーバまたはその他の中間記憶デバイスを含み得る。この例では、送信先デバイス１４は、ストリーミングまたはダウンロードにより、ファイルサーバまたはそ

の他の中間記憶デバイスで記憶された符号化ビデオデータにアクセスすることができる。ファイルサーバは、符号化ビデオデータを記憶し、送信先デバイス１４に符号化ビデオデータを送信することができる種類のサーバとすることができる。ファイルサーバの例としては、（例えばウェブサイトのための）ウェブサーバ、ファイル転送プロトコル（ＦＴＰ）サーバ、ネットワーク接続ストレージ（ＮＡＳ）、およびローカルディスクドライブがある。

【００３１】

送信先デバイス１４は、インターネット接続などの標準データ接続を介して符号化ビデオデータにアクセス可能である。データ接続の種類の例としては、ファイルサーバに記憶された符号化ビデオデータにアクセスするのに適した、ワイヤレスチャネル（例えばＷｉＦｉ（登録商標）接続）、有線接続（例えばＤＳＬ、ケーブルモデムなど）、または両者の組み合わせなどがある。ファイルサーバからの符号化ビデオデータの送信は、ストリーミング送信、ダウンロード送信、または両者の組み合わせとすることができる。

【００３２】

本開示の技術は、ワイヤレス用途または環境には限定されない。本開示の技術は、無線テレビ放送、ケーブルテレビ送信、衛星テレビ送信、例えばインターネットを介したストリーミングビデオ送信など、様々なマルチメディア用途、ビデオデータをデータ記憶媒体に記憶するための符号化、データ記憶媒体に記憶されたビデオデータの復号、またはその他の用途をサポートするビデオコーディングに適用可能である。ある例では、ビデオコーディングシステム１０は、一方向または双方向ビデオ送信に対応してビデオストリーミング、ビデオ再生、ビデオブロードキャスト、および／またはテレビ電話などの用途をサポートするように構成され得る。

【００３３】

図１の例では、送信元デバイス１２はビデオ源１８，ビデオエンコーダ２０，および出力インターフェース２２を含む。ある例では、出力インターフェース２２は変調器／復調器（モデム）および／または送信器を含み得る。ビデオ源１８は、ビデオキャプチャーデバイス、例えばビデオカメラ、過去にキャプチャされたビデオデータを収めるビデオアーカイブ、ビデオコンテンツプロバイダからビデオデータを受信するためのビデオフィードインターフェース、および／またはビデオデータを生成するためのコンピュータグラフィクスシステム、またはこのようなビデオデータ源の組み合わせを含み得る。

【００３４】

ビデオエンコーダ２０は、ビデオ源１８からのビデオデータを符号化することができる。ある例では、送信元デバイス１２は、出力インターフェース２２を介して送信先デバイス１４に符号化ビデオデータを直接送信することができる。符号化ビデオデータは、送信先デバイス１４が復号および／または再生のために後でアクセスするように、記憶媒体またはファイルサーバに記憶することもできる。

【００３５】

図１の例では、送信先デバイス１４は入力インターフェース２８と、ビデオデコーダ３０と、表示デバイス３２とを含む。ある例では、入力インターフェース２８は、受信器および／またはモデムを含み得る。送信先デバイス１４の入力インターフェース２８は、チャネル１６を介して符号化ビデオデータを受信することができる。表示デバイス３２は、送信先デバイス１４に組み込まれていても外部にあってもよい。一般に、表示デバイス３２は復号ビデオデータを表示する。表示デバイス３２は、液晶ディスプレイ（ＬＣＤ）、プラズマディスプレイ、有機発光ダイオード（ＯＬＥＤ）ディスプレイ、またはその他の種類の表示デバイスを備え得る。

【００３６】

ビデオエンコーダ２０とビデオデコーダ３０とは、現在策定中の高効率ビデオコーディング（ＨＥＶＣ）標準などのビデオ圧縮標準に従って動作することができ、ＨＥＶＣテストモデル（ＨＭ）に準拠し得る。「ＨＥＶＣ草案４」または「ＷＤ４」と呼ばれる公開予定のＨＥＶＣ標準の最近の草案は、ブロス（Ｂｒｏｓｓ）らの「ＷＤ４：Ｗｏｒｋｉｎｇ

Ｄｒａｆｔ４ ｏｆ ＨＩｇｈ－Ｅｆｆｉｃｉｅｎｃｙ Ｖｉｄｅｏ Ｃｏｄｉｎｇ（高効率ビデオコーディング草案４）」（ＩＴＵ－Ｔ ＳＧ１６ ＷＰ３およびＩＳＯ／ＩＥＣ ＪＴＣ１／ＳＣ２９／ＷＧ１１のビデオコーディング共同チーム（ＪＣＴ－ＶＣ）、第６回会議：イタリア、トリノ２００１年７月）に記載されており、これは２０１２年９月２７日の時点でｈｔｔｐ：／／ｐｈｅｎｉｘ．ｉｎｔ－ｅｖｒｙ．ｆｒ／ｊｃｔ／ｄｏｃ＿ｅｎｄ＿ｕｓｅｒ／ｄｏｃｕｍｅｎｔｓ／６＿Ｔｏｒｉｎｏ／ｗｇ１１／ＪＣＴＶＣ－Ｆ８０３－ｖ３．ｚｉｐからダウンロード可能であり、その全内容が参考として本明細書に援用される。

【００３７】

あるいは、ビデオエンコーダ２０とビデオデコーダ３０とは、ＩＴＵ－Ｔ Ｈ．２６１、ＩＳＯ／ＩＥＣ ＭＰＥＧ－１ビジュアル、ＩＴＵ－Ｔ Ｈ．２６２またはＩＳＯ／ＩＥＣ ＭＰＥＧ－２ビジュアル、ＩＴＵ－Ｔ Ｈ．２６３、ＩＳＯ／ＩＥＣ ＭＰＥＧ－４ビジュアルおよびＩＴＵ－Ｔ Ｈ．２６４（ＩＳＯ／ＩＥＣ ＭＰＥＧ－４ ＡＶＣとも呼ばれる）、そのスケーラブルビデオコーディング（ＳＶＣ）およびマルチビュービデオコーディング（ＭＶＣ）拡張版などを含む、その他の企業標準または業界標準に従って動作することもできる。ただし、本開示の技術はいかなる特定のコーディング標準または技術にも限定されない。

【００３８】

さらに、図１は一例に過ぎず、本開示の技術は符号化デバイスと復号デバイスとの間でのデータ通信を必ずしも含まないビデオコーディング環境（例えばビデオ符号化またはビデオ復号）にも適用可能である。その他の例では、データはローカルメモリから取り出される、ネットワークを介してストリーミングされる、などが可能である。符号化デバイスはデータを符号化してメモリに格納してもよく、または、復号デバイスはメモリからデータを読み出して復号してもよく、あるいはその両方であってもよい。多くの例では、互いに通信せずデータを単にメモリに対して符号化、および／または、メモリから取り出して復号するデバイスによって符号化と復号が行われる。

【００３９】

ビデオエンコーダ２０とビデオデコーダ３０とはそれぞれ、１つまたは複数のマイクロプロセッサ、デジタル信号プロセッサ（ＤＳＰ）、特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、個別論理素子、ハードウェア、またはこれらの任意の組み合わせなど、様々な適切な回路のいずれかとして実装され得る。本技術の一部をソフトウェアで実装する場合、デバイスはそのソフトウェアのための命令を、適切な非一時的コンピュータ可読記憶媒体に記憶することができ、１つまたは複数のプロセッサを使用してハードウェアで命令を実行して本開示の技術を実行することができる。以上（ハードウェア、ソフトウェア、ハードウェアとソフトウェアの組み合わせなどを含む）のいずれも、１つまたは複数のプロセッサであると見なされ得る。ビデオエンコーダ２０とビデオデコーダ３０のそれぞれは、１つまたは複数のエンコーダまたはデコーダに含まれてもよく、両者のいずれかがそれぞれのデバイスの複合エンコーダ／デコーダ（ＣＯＤＥＣ）の一部として組み込まれてもよい。

【００４０】

本開示では、ビデオエンコーダ２０が特定の情報をビデオデコーダ３０などの他のデバイスに「通知する」と一般的にいう場合がある。しかし、ビデオエンコーダ２０は特定の構文要素をビデオデータの様々な符号化部分と関連付けることによって情報を通知してもよいことを理解すべきである。すなわち、ビデオエンコーダ２０は、特定の構文要素をビデオデータの様々な符号化部分に格納することにより、データを「通知」することができる。場合によっては、このような構文要素は、ビデオデコーダ３０によって受信され復号される前に符号化し（例えば記憶システムに）格納され得る。したがって、「通知」という用語は、圧縮ビデオデータを復号するために使用される構文要素および／またはその他のデータの通信を総称することがある。そのような通信は、リアルタイムまたはほぼリアルタイムで行われてもよい。あるいは、そのような通信は、符号化時に構文要素を符号化

ビットストリームの形でコンピュータ可読記憶媒体に記憶し、その媒体に記憶された後の任意の時点で復号デバイスによって読み出される場合などのように、ある期間にわたって行われてもよい。

【００４１】

上記で簡単に述べたようにビデオエンコーダ２０はビデオデータを符号化する。ビデオデータは１つまたは複数のピクチャを備え得る。各ピクチャはビデオの一部を形成する静止画像であってよい。場合によっては、ピクチャをビデオ「フレーム」と呼ぶことがある。ビデオエンコーダ２０は、ビデオデータのコード化表現を形成するビット列を含むビットストリームを生成することができる。ビットストリームはコード化ピクチャとそれに関連付けられたデータとを含み得る。コード化ピクチャはピクチャのコード化表現である。

【００４２】

ビットストリームを生成するために、ビデオエンコーダ２０は一連のコード化ピクチャとそれに関連付けられたデータとを生成することができる。コード化ピクチャはビデオデータ内のピクチャの符号化表現である。関連付けられたデータは、シーケンスパラメータセット（ＳＰＳ）、ピクチャパラメータセット（ＰＰＳ）、アダプテーションパラメータセット（ＡＰＳ）、およびその他の構文構造体を含み得る。ＳＰＳは、ゼロまたはそれ以上のピクチャシーケンスに適用可能なパラメータを含み得る。ＰＰＳは、ゼロまたはそれ以上のピクチャに適用可能なパラメータを含み得る。ＡＰＳは、ゼロまたはそれ以上のピクチャに適用可能なパラメータを含み得る。ＡＰＳ内のパラメータは、ＰＰＳ内のパラメータよりも変化する可能性が高い。

【００４３】

ピクチャの符号化表現を生成するために、ビデオエンコーダ２０はピクチャを複数のツリーブロックに区画化することができる。場合によっては、ツリーブロックを最大コーディングユニット（ＬＣＵ）または「コーディングツリーブロック」と呼ぶことがある。ツリーブロックは、Ｈ．２６４／ＡＶＣなど、以前の標準のマクロブロックに大まかに類似し得る。しかし、ツリーブロックは、必ずしも特定の大きさには限定されず、１つまたは複数のコーディングユニット（ＣＵ）を含み得る。

【００４４】

ツリーブロックのそれぞれは、ピクチャ内の等しい大きさの異なる画素ブロックに関連付けられ得る。各画素は、輝度（Ｙ）サンプルと、Ｃｂクロミナンスサンプルと、Ｃｒクロミナンスサンプルとを備え得る。したがって各ツリーブロックは、ピクチャの輝度（Ｙ）サンプルのブロックと、Ｃｂクロミナンスサンプルのブロックと、Ｃｒクロミナンスサンプルのブロックとに関連付けられ得る。説明の便宜上、本開示では、画素の２次元配列を画素ブロックと呼び、サンプルの２次元配列をサンプルブロックと呼ぶことがある。ビデオエンコーダ２０は、四分木区画化を使用して、ツリーブロックに関連付けられた画素ブロックをＣＵに関連付けられた画素ブロックに区画化することができ、したがって「ツリーブロック」と称される。

【００４５】

ビデオエンコーダ２０は、ピクチャを複数のスライスに区画化することができる。各スライスは、整数個のツリーブロックを含み得る。ピクチャの符号化の一部として、ビデオエンコーダ２０はピクチャの各スライスの符号化表現、（すなわちコード化スライス）を生成することができる。コード化スライスを生成するために、ビデオエンコーダ２０はスライスの各ツリーブロックを符号化して、スライスの各ツリーブロックの符号化表現（すなわちコード化ツリーブロック）を生成する。

【００４６】

コード化ツリーブロックを生成するために、ビデオエンコーダ２０は、ツリーブロックに関連付けられた画素ブロックに対して再帰的に四分木区画化を行って、画素ブロックを漸進的に縮小する画素ブロックに分割する。このより小さい画素ブロックのそれぞれはＣＵに関連付けられ得る。区画化されたＣＵは、その画素ブロックが他のＣＵに関連付けられた画素ブロックに区画化されたＣＵとすることができる。区画化されていないＣＵは、

その画素ブロックが他のＣＵに関連付けられた画素ブロックに区画化されていないＣＵとすることができる。

【００４７】

ビデオエンコーダ２０は、区画化されていない各ＣＵについて１つまたは複数の予測ユニット（ＰＵ）を生成することができる。ＣＵの各ＰＵは、そのＣＵの画素ブロック内の異なる画素ブロックに関連付けられ得る。ビデオエンコーダ２０は、ＣＵの各ＰＵについて予測画素ブロックを生成することができる。ＰＵの予測画素ブロックは、画素のブロックであってよい。

【００４８】

ビデオエンコーダ２０は、イントラ予測またはインター予測を使用してＰＵの予測画素ブロックを生成することができる。ビデオエンコーダ２０がイントラ予測を使用してＰＵの予測画素ブロックを生成する場合、ビデオエンコーダ２０はＰＵに関連付けられたピクチャの復号画素に基づいてＰＵの予測画素ブロックを生成することができる。ビデオエンコーダ２０がインター予測を使用してＰＵの予測画素ブロックを生成する場合、ビデオエンコーダ２０はＰＵに関連付けられたピクチャ以外の１つまたは複数のピクチャの復号画素に基づいて、ＰＵの予測画素ブロックを生成することができる。

【００４９】

ビデオエンコーダ２０は、ＣＵのＰＵの予測画素ブロックに基づいてそのＣＵの残差画素ブロックを生成することができる。ＣＵの残差画素ブロックは、ＣＵのＰＵのための予測画素ブロック内のサンプルとＣＵの元の画素ブロック内の対応するサンプルとの差を示すことができる。

【００５０】

さらに、非区画化ＣＵの符号化の一部として、ビデオエンコーダ２０はＣＵの残差画素ブロックに対して再帰的四分木区画化を行って、ＣＵの残差画素ブロックをＣＵの変換ユニット（ＴＵ）に関連付けられた１つまたは複数のより小さい残差画素ブロックに区画化することができる。このようにして、ＣＵの各ＴＵが輝度（Ｙ）サンプルの残差サンプルブロックと、クロミナンス（ＣｒまたはＣｂ）サンプルの２つの残差サンプルブロックとに関連付けられ得る。

【００５１】

ビデオコーダ２０は、ＴＵに関連付けられた残差サンプルブロックに１つまたは複数の変換を適用して、ＴＵに関連付けられた係数ブロック（すなわち係数のブロック）を生成することができる。概念上は、係数ブロックは係数の２次元行列とすることができる。ビデオエンコーダ２０は、係数ブロックに対して量子化処理を行うことができる。量子化とは、係数を量子化して係数を表すために使用されるデータの量をできるだけ削減することにより、さらなる圧縮を実現する処理を一般的に指す。

【００５２】

ビデオエンコーダ２０は、量子化係数ブロック内の係数を表す構文要素のセットを生成することができる。ビデオエンコーダ２０は、コンテキスト適応型２値算術コーディング（Ｃｏｎｔｅｘｔ　Ａｄａｐｔｉｖｅ　Ｂｉｎａｒｙ　Ａｒｉｔｈｍｅｔｉｃ　Ｃｏｄｉｎｇ：ＣＡＢＡＣ）演算などのエントロピー符号化演算を、これらの構文要素の少なくとも一部に適用してもよい。エントロピー符号化演算の実行の一部として、ビデオエンコーダ２０はコーディングコンテキストを選択してもよい。ＣＡＢＡＣの場合、コーディングコンテキストは０値のビンと１値のビンの確率を示すことができる。コンテキスト適応型可変長コーディング（ｃｏｎｔｅｘｔ－ａｄａｐｔｉｖｅ　ｖａｒｉａｂｌｅ　ｌｅｎｇｔｈ　ｃｏｄｉｎｇ：ＣＡＶＬＣ）の場合、コーディングコンテキストは１組の可変長コードとすることができる。ビデオエンコーダ２０はコーディングコンテキストを使用して１つまたは複数の構文要素を符号化することができる。

【００５３】

ビデオエンコーダ２０によって生成されたビットストリームは、一連のネットワーク抽象化レイヤ（ＮＡＬ）ユニットを含み得る。各ＮＡＬユニットは、ＮＡＬユニット内のデ

ータのタイプを示す標識およびデータを含むバイトを含む構文構造とすることができる。例えば、ＮＡＬユニットはＳＰＳ、ＰＰＳ、コード化スライス、補助強化情報（ｓｕｐｐｌｅｍｅｎｔａｌ　ｅｎｈａｎｃｅｄ　ｉｎｆｏｒｍａｔｉｏｎ：ＳＥＩ）、アクセスユニットデリミタ、フィラーデータ、またはその他のタイプのデータを表すデータを含み得る。ＮＡＬユニット内のデータは様々な構文構造を含み得る。コード化スライスＮＡＬユニットは、コード化スライスを含むＮＡＬユニットである。「ＶＣＬ　ＮＡＬユニット」は、コード化スライスＮＡＬユニットと、ＶＣＬ　ＮＡＬユニットに分類されるｎａｌ＿ｕｎｉｔ＿ｔｙｐｅの予約値を有するＮＡＬユニットのサブセットとの総称である。

【００５４】

ＳＥＩ　ＮＡＬユニットは、ＳＥＩメッセージを含むＮＡＬユニットである。ＳＥＩメッセージは、誤り耐性、タイミング、出力などに有用となり得る情報を含む。ただし、ＳＥＩメッセージは復号処理では必要ない。言い換えると、ビデオデコーダは、ＳＥＩメッセージ内の情報を使用せずにビデオデータをデコード可能とすることができる。したがって、Ｈ．２６４／ＡＶＣまたはＨＥＶＣなどのビデオコーディング標準の環境ではＳＥＩメッセージは標準的ではなく（すなわち有用な情報ではなく）、準拠するビデオデコーダはビットストリーム中に存在するＳＥＩメッセージを単純に廃棄してもかまわないという意味で任意的である。さらに、ビデオデータに関連付けられたＳＥＩメッセージを送信しなくても、ビデオデコーダはビデオデータを復号することができる。

【００５５】

ビデオエンコーダ２０は、ピクチャを複数のピクチャ区画に分割することができる。ピクチャ区画は、ピクチャのツリーブロックの重なり合わないサブセットに関連付けることができる。ビデオエンコーダ２０は様々な方法でピクチャを複数のピクチャ区画に分割することができる。以下に述べるように、ビデオエンコーダ２０は、ピクチャを複数のエントロピースライス、複数のタイル、または複数の波面並列処理（ＷＰＰ）波に分割することができる。本開示では、「ピクチャ区画」という用語を使用してエントロピースライス、タイルおよびＷＰＰ波を総称的に指すことがある。

【００５６】

上述のように、ビデオエンコーダ２０はピクチャを複数のエントロピースライスに分割することができる。エントロピースライスを軽量スライス、短いスライスヘッダーの付いたスライス、または短いスライスとも呼ぶことがある。エントロピースライスは、通常のスライスのＣＵのサブセットを含み得る。ある例では、ビデオエンコーダ２０は、エントロピースライスのうちのＣＵを、エントロピースライスのいずれも上限よりも多くのビン（例えばエントロピーコード化ビン）を含まないように区画化してもよい。場合によっては、単一のコード化スライスＮＡＬユニットがスライスの複数のエントロピースライスを含み得る。他の例では、各エントロピースライスが別個のＮＡＬユニットに含まれてもよい。

【００５７】

エントロピースライスの境界を越えるピクチャ内予測が許容される。例えば、ビデオコーダが特定のＰＵについてイントラ予測を行う場合、ビデオコーダは、隣接するＰＵが特定のＰＵとは異なるエントロピースライス内にある場合であっても、その隣接ＰＵ内のサンプルを使用することができる。この例では、ビデオコーダは、隣接するＰＵが特定のＰＵとは異なるスライス内にある場合、その隣接ＰＵのサンプルを使用することができなくてもよい。

【００５８】

しかし、ビデオコーダーが特定のＰＵに関連付けられたデータのエントロピーコーディングを行うとき、ビデオコーダは、特定のＰＵと隣接ＰＵとが同じエントロピースライス内にある場合にのみ、隣接ＰＵに関連付けられた情報に基づいてコーディングコンテキストを選択することが許容される。この制約のため、ビデオコーダは、１つのスライスの複数のエントロピースライスに対してエントロピーコーディング（すなわちエントロピー符号化または復号）動作を並列して行うことが可能であってもよい。ただし、ビデオデコー

ダ３０は、１つのスライスの複数のエントロピースライスの画素ブロックを並列して再構築することはできない。

【００５９】

ある例では、ビデオエンコーダ２０はピクチャを１つまたは複数のタイルに分割することができる。各タイルは、ピクチャ内の整数個のツリーブロックを備え得る。ビデオエンコーダ２０は、２つ以上の垂直方向のタイル境界と２つ以上の水平方向のタイル境界を画定することによって、ピクチャをタイルに分割してもよい。ピクチャの垂直方向の各辺は、垂直タイル境界とみなすことができる。現在のピクチャの水平方向の各辺は、水平タイル境界とみなすことができる。例えば、ピクチャについてビデオエンコーダ２０が４つの垂直タイル境界と３つの水平タイル境界を画定する場合、現在のピクチャは６つのタイルに分割される。

【００６０】

ビデオエンコーダ２０またはビデオデコーダ３０などのビデオコーダは、ラスタースキャン順に従ってピクチャのタイルをコーディングすることができる。さらに、ビデオコーダは、タイル内の各ツリーブロックをラスタースキャン順に従ってコーディングすることができる。このようにして、ビデオコーダは、ピクチャの所与のタイルの各ツリーブロックを、ピクチャの他のタイルのいずれのツリーブロックのコーディングよりも前にコーディングすることができる。したがって、ピクチャが複数のタイルに区画化されている場合にビデオコーダがピクチャのツリーブロックをコーディングする順序は、ピクチャが複数のタイルに区画化されていない場合とは異なり得る。

【００６１】

場合によっては、ビデオコーダは、特定のＣＵと空間的に隣接するＣＵとが同じタイルに属しており、その特定のＣＵと空間的に隣接するＣＵとが同じスライスに属している限り、その空間的に隣接するＣＵに関連付けられた情報を使用してその特定のＣＵに対してイントラ予測を行うことができる。空間的に隣接するＣＵは、特定のＣＵに関連付けられた画素ブロックに隣接する画素ブロックに関連付けられたＣＵとすることができる。場合によっては、ビデオコーダは、特定のＣＵと空間的に隣接するＣＵとが同じタイル内にある限り、その空間的に隣接するＣＵに関連付けられた情報を使用して、その特定のＣＵの構文要素をＣＡＢＡＣ符号化するためのコンテキストを選択してもよい。これらの制約のために、ビデオコーダは複数のタイルのツリーブロックを並列してコーディング可能であってもよい。

【００６２】

例えば、画素値予測、動き予測、コーディングモード予測、およびエントロピーコーディングコンテキスト予測を含む、タイル境界を越えるピクチャ内予測は、フラグｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｉｎｄｅｐｅｎｄｅｎｃｅ＿ｉｄｃによって制御することができる。このフラグが１の場合、タイル境界を越えるピクチャ内予測は許容されない。それ以外の場合、ピクチャ境界またはスライス境界でもあるタイル境界を除き、タイル境界を越えるピクチャ内予測が許容される。ピクチャ内予測が許容されない場合、タイルの機能は、ピクチャ内にタイルがない場合、または同様に、ピクチャ内にタイルが１つしかない場合と比較して、ツリーブロックのスキャン順序を変更することであってもよい。ピクチャ内予測が許容される場合、ツリーブロックのスキャン順序の変更のほかに、タイルは、並列コーディング（符号化および／または復号）に使用可能な独立区画化も提供することができる。

【００６３】

ピクチャ内予測がタイル境界を越えて許容されるか否かにかかわらず、ビデオエンコーダ２０は、単一のＮＡＬユニット中の単一のエントロピースライス内に複数の符号化タイルを含めてもよい。この場合、タイル境界を越えるピクチャ内予測が許容される場合でも、タイル境界を越えるエントロピーコーディングコンテキスト予測は許容されない。さらに、ＮＡＬユニットが１つのエントロピースライスしか含むことができず、エントロピースライスが１つのタイルのＣＵしか含まない例では、コード化タイルが別々のＮＡＬユニ

ット内にあるため、エントロピースライスの各符号化タイルの入口点を通知する必要がない場合もある。

【００６４】

他の例では、ビデオコーダは波面並列処理（ＷＰＰ）を使用してピクチャをコーディングすることができる。ビデオコーダがＷＰＰを使用してピクチャをコーディングする場合、ビデオコーダはピクチャのツリーブロックを複数の「ＷＰＰ波」に分割することができる。各ＷＰＰ波は、ピクチャ内のツリーブロックの異なる行に対応し得る。ビデオコーダがＷＰＰを使用してピクチャをコーディングする場合、ビデオコーダはツリーブロックの最上行からコーディングを開始してもよい。ビデオコーダが最上行の２つ以上のツリーブロックをコーディングした後、ビデオコーダはツリーブロックの上から２番目の行のコーディングをツリーブロックの最上行のコーディングと並列して開始することができる。ビデオコーダが上から２行目の２つ以上のツリーブロックをコーディングした後、ビデオコーダはツリーブロックの上から３番目の行のコーディングを、ツリーブロックの上の行のコーディングと並行して開始することができる。このパターンをピクチャ内のツリーブロックの各行を下って続けることができる。

【００６５】

ビデオコーダがＷＰＰを使用してピクチャをコーディングする場合、ビデオコーダは現在のツリーブロック外部の空間的に隣接するＣＵが現在のツリーブロックの左、左上、上、または右上にある限り、その隣接ＣＵに関連付けられた情報を使用して現在のツリーブロック内の特定のＣＵに対するイントラ予測を行うことができる。現在のツリーブロックが最上行以外の行の左端のツリーブロックである場合、ビデオコーダはすぐ上の行の２番目のツリーブロックに関連付けられた情報を使用して、現在のツリーブロックの構文要素をＣＡＢＡＣ符号化するためのコンテキストを選択してもよい。一方、現在のツリーブロックが行の左端のツリーブロックではない場合は、ビデオコーダは現在のツリーブロックの左のツリーブロックに関連付けられた情報を使用して、現在のツリーブロックの構文要素をＣＡＢＡＣ符号化するためのコンテキストを選択してよい。このようにして、ビデオコーダはすぐ上の行の２つ以上のツリーブロックの符号化の後、すぐ上の行のＣＡＢＡＣの状態に基づいて行のＣＡＢＡＣの状態を初期化することができる。

【００６６】

ある例では、ビデオコーダがＷＰＰを使用してピクチャをコーディングする場合、ピクチャのタイル境界はピクチャの水平方向と垂直方向の境界のみである。したがって、ピクチャの唯一のタイルは、ピクチャと同じ大きさである。ビデオコーダは、そのピクチャ、したがってピクチャの単一のタイルを、複数のＷＰＰ波に分割することができる。

【００６７】

前述のように、ビデオコーダ２０はスライスの符号化表現を含むコード化スライスＮＡＬユニットを生成することができる。このスライスは、整数個の連続コード化ツリーブロックに関連付けることができる。コード化スライスＮＡＬユニットは、スライスヘッダーとスライスデータとを含み得る。スライスデータはスライスに関連付けられた各ツリーブロックの符号化表現を含み得る。

【００６８】

ビデオエンコーダ２０は、ツリーブロックが属するピクチャ区画に従ってツリーブロックの符号化表現がスライスデータ内でグループ化されるように、コード化スライスＮＡＬユニットを生成することができる。例えば、コード化スライスＮＡＬユニットは、第１のピクチャ区画に関連付けられた各コード化ツリーブロックと、その後の第２のピクチャ区画に関連付けられた各コード化ツリーブロックと、その後の第３のピクチャ区画に関連付けられた各コード化ツリーブロックと、という具合にツリーブロックを含み得る。

【００６９】

説明の便宜上、本開示では「符号化ピクチャ区画」、「コード化サブスライス」、「コード化サブストーム」という用語を使用して、単一のピクチャ区画（例えばエントロピースライス、タイル、またはＷＰＰ波）に関連付けられたコード化スライスＮＡＬユニット

内のコード化ツリーブロックのグループを指す。ただし、ピクチャ区画が複数のスライス内のツリーブロックを含み得る状況もあり得るものと理解すべきである。以下で説明する図１０に、タイルが複数のスライスに属するツリーブロックを含み、スライスが複数のタイルに属するツリーブロックを含む例を示す。

【００７０】

ビデオデコーダ３０は、ビデオエンコーダ２０によって生成されたビットストリームを受信することができる。ビットストリームは、ビデオエンコーダ２０によって符号化されたビデオデータのコード化表現を含み得る。ビデオデコーダ３０は、ビットストリームに対し解析動作を実行することができる。ビデオデコーダ３０は、解析動作の実行の一部として、ビットストリームから構文要素を抽出してもよい。ビデオデコーダ３０は、ビットストリームからの構文要素の抽出の一部として、ビットストリーム内のデータに対してエントロピー復号（例えばＣＡＢＡＣ復号）動作を行ってもよい。ビデオデコーダ３０は、ビットストリームから抽出した構文要素に基づいてビデオデータのピクチャを再構築することができる。構文要素に基づいてビデオデータを再構築する処理は、概ね、それらの構文要素を生成するためにビデオエンコーダ２０によって行われた処理の逆であり得る。

【００７１】

ビデオデコーダ３０は、ＣＵに関連付けられた構文要素に基づいて、ＣＵのＰＵのための予測画素ブロックを生成することができる。さらに、ビデオデコーダ３０は、ＣＵのＴＵに関連付けられた係数ブロックを逆量子化することができる。ビデオデコーダ３０は、係数ブロックに逆変換を行ってＣＵのＴＵに関連付けられた残差画素ブロックを再構築することができる。ビデオデコーダ３０は、予測画素ブロックと残差画素ブロックとに基づいてＣＵの画素ブロックを再構築することができる。このようにして、ビデオデコーダ３０は、ビットストリーム内の構文要素に基づいてＣＵの画素ブロックを再構築し得る。

【００７２】

前述のように、ビデオデコーダ３０のある種の実装は、異なるピクチャ区画の画素ブロックを並列して復号可能な場合がある。例えば、コード化スライスＮＡＬユニットが第１の符号化画素区画と第２の符号化画素区画とを含む場合、ビデオデコーダ３０は、第１の符号化画素区画の画素ブロックと第２の符号化画素区画の画素ブロックとを並列して復号可能な場合がある。

【００７３】

ビデオデコーダ３０は、コード化スライスＮＡＬユニットをメモリに格納し得る。メモリはバイトアドレス指定可能とすることができる。ビデオデコーダ３０はコード化スライスＮＡＬユニットをバイトアドレス指定可能メモリに格納するため、符号化画素区画の始点がバイト内にある場合、ビデオデコーダ３０は符号化画素区画の始点に関連付けられたメモリアドレスを示すことができない場合がある。したがって、符号化画素区画の１つまたは複数がバイト内で始まる場合、ビデオデコーダ３０は、符号化画素区画を並列して復号することができない場合がある。あるいは、ビデオデコーダ３０は、ビット単位メモリアドレス指定、またはバイト単位とビット単位とのアドレス指定を使用して、符号化画素区画の並列復号を可能にしてもよいが、実装と計算が複雑化する。

【００７４】

この問題に対処するために、ビデオエンコーダ２０は、コード化スライスＮＡＬ内の各符号化画素区画がバイト整列されるようにコード化スライスＮＡＬユニットを生成する。すなわち、コード化スライスＮＡＬユニット内の各符号化画素区画はバイト境界から開始し得る。符号化画素区画がバイト整列される場合、符号化画素区画の先頭ビットはメモリ内のアドレス指定可能バイトの先頭ビットであり、符号化画素区画の最後のビットはメモリ内のアドレス指定可能バイトの最後のビットである。したがって、メモリ内の各バイトアドレス指定記憶場所にとっては、記憶場所は複数の符号化画素区画に関連付けられたビットを格納しない。

【００７５】

さらに、ビデオデコーダ３０が符号化画素区画を並列して復号する場合、ビデオデコー

ダ３０は異なる処理コアに異なる符号化画素区画を割り当てることができる。これらの処理コアは、符号化画素区画を並列して復号することができる。さらに、処理コアはメモリに対してスライスヘッダーを要求してもよい。メモリはこれに応答して処理コアにスライスヘッダーを送る。このようにして、各処理コアは、符号化画素区画の復号時にスライスヘッダーを使用することができてもよい。しかし、スライスヘッダーがバイト整列されていない場合、メモリはスライスヘッダー内にない何らかの情報も送らずにスライスヘッダーを処理コアに送ることができないことがある。例えば、スライスヘッダーがバイト整列されていない場合、ビデオデコーダ３０は、スライスヘッダーがバイト整列されるようにスライスヘッダーをトランスコーディングすることも考えられる。スライスヘッダーをトランスコーディングすると、ビデオデコーダ３０が複雑化する可能性があり、処理コアを、トランスコードされたスライスヘッダーを処理するように構成する必要が生じる可能性がある。

【００７６】

各符号化画素区画がバイト整列されたコード化スライスＮＡＬユニットを生成するために、ビデオエンコーダ２０は、スライスヘッダーがバイト境界で終わるようにコード化スライスＮＡＬユニットのスライスヘッダーにパディングしてもよい。すなわち、スライスヘッダーの最後のビットはメモリ内のアドレス指定可能バイトの最後のビットとすることができる。コード化スライスＮＡＬユニットのスライスヘッダーのパディングは、スライスヘッダーがバイト整列されるまでスライスヘッダーの後ろにビットを追加することを含み得る。

【００７７】

このようにして、ビデオエンコーダ２０は、各符号化画素区画がビデオデータ中のピクチャ内の異なる１組のコーディングユニットに関連付けられた、複数の符号化画素区画を生成することができる。さらに、ビデオエンコーダ２０は、スライスヘッダーをバイト境界に整列させる１つまたは複数のパディングビットを含むスライスヘッダーと複数の符号化画素区画とを含むＮＡＬユニットを生成し得る。同様に、ビデオデコーダ３０は、ＮＡＬユニットを受信することができる。ＮＡＬユニットは、スライスヘッダーとビデオデータの複数の符号化画素区画とを含み得る。スライスヘッダーは、スライスヘッダーをバイト境界に整列させる１つまたは複数のパディングビットを含み得る。ビデオデコーダ３０は、２つ以上の符号化画素区画を並列して復号してよい。

【００７８】

符号化画素区画を並列して復号するために、ビデオデコーダ３０は、並列して実行される異なる復号スレッドに符号化画素区画を割り当てることができる。復号スレッドは異なる処理コア上で実行され得る。符号化画素区画を異なる復号スレッドに割り当てるために、ビデオデコーダ３０は各符号化画素区画の先頭に関連付けられたメモリアドレスを判断する必要がある場合がある。

【００７９】

ビデオデコーダ３０が各符号化画素区画の先頭に関連付けられたメモリアドレスを判断できるようにするため、コード化スライスＮＡＬユニットのスライスヘッダーは入口点オフセットを含み得る。ある例では、入口点オフセットは、他の点を基準にした符号化画素区画のメモリアドレスのオフセットを示す。ある例では、オフセットはコード化スライスＮＡＬユニットの先頭を基準としてもよい。例えば、コード化スライスＮＡＬユニットのスライスヘッダー中の入口点オフセットは、特定の符号化画素区画が、コード化スライスＮＡＬユニットの３１２番目バイトから始まることを示してもよい。他の例では、コード化スライスＮＡＬユニット内の符号化画素区画の入口点オフセットは、コード化スライスＮＡＬユニット内の先行符号化画素区画の第１のビットを基準としてもよい。例えば、入口点オフセットはコード化スライスＮＡＬユニット内の特定の符号化画素区画が、コード化スライスＮＡＬユニット内の先行符号化画素区画の１２４バイト後から開始することを示してもよい。さらに、ある例では、入口点オフセットは、符号化画素区画の長さをビット数またはバイト数で示してもよい。

【００８０】
本開示の１つまたは複数の技術によると、入口点オフセットを示すオフセット構文要素に加えて、ビデオエンコーダ２０はスライスヘッダー内に２つの追加構文要素を含めることができる。スライス中のタイルの数は追加構文要素の最初の追加構文要素に基づいて決定してもよい。各オフセット構文要素中のビットの数は、追加構文要素の第２の構文要素に基づいて判断し得る。本開示の技術によると、ビデオエンコーダ２０は、追加構文要素をエントロピー符号化（例えば指数ゴロム符号化）してもよい。ビデオエンコーダ２０が追加構文要素をエントロピー符号化するため、追加構文要素は可変長ビット列を使用したビットストリームで表される。可変長ビット列を使用して追加構文要素を表すことにより、固定長ビット列を使用して追加構文要素を表す場合よりも少ないビットを含むビットストリームとなるとともに、ビデオエンコーダ２０は例えば５ビットを使用する固定長コーディングによって可能なよりも多くのオフセット構文要素を通知することができる。
【００８１】
このようにして、ビデオエンコーダ２０は、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とをエントロピー符号化することができる。各構文要素の長さは第１の構文要素に基づいて判断可能としてもよい。一連のオフセット構文要素中のオフセット構文要素の数は、第２の構文要素に基づいて判断可能としてもよい。ＮＡＬユニット内の複数の符号化画素区画の位置はオフセット構文要素に基づいて判断可能としてもよい。さらにビデオエンコーダ２０は、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームを生成することができる。
【００８２】
ビデオデコーダ３０は、第１の追加構文要素を表す可変長値と、第２の追加構文要素を表すエントロピー符号化可変長値と、一連のオフセット構文要素とを含むビットストリームを受信することができる。ビデオデコーダ３０は、第１と第２の追加構文要素をエントロピー復号することができる。さらに、ビデオデコーダ３０は、第１の構文要素に基づいて、オフセット構文要素のそれぞれの長さを判断することができ、第２の構文要素に基づいて、一連のオフセット構文要素中のオフセット構文要素の数を判断することができる。ビデオデコーダ３０は、オフセット構文要素のそれぞれの長さと一連のオフセット構文要素中のオフセット構文要素の数とに少なくとも部分的に基づいて、ビットストリーム内のオフセット構文要素を構文解析することができる。次にビデオデコーダ３０は、オフセット構文要素に少なくとも部分的に基づいて、ＮＡＬユニット内の複数の符号化画素区画の位置を判断することができる。
【００８３】
さらに、コード化スライスＮＡＬユニットのスライスヘッダー内の入口点オフセットを示す代わりに（またはそれに加えて）、コード化スライスＮＡＬユニットのスライスデータが入口点マーカを含んでもよい。入口点マーカは、入口点マーカに固有のビットパターンを指定し得る。入口点マーカは、スライスデータの第１の符号化画素区画のほか、各符号化画素区画の直前のスライスデータ内にあってもよい。ビデオデコーダ３０は、スライスデータをスキャンし、入口点マーカを識別することによって、各符号化画素区画の先頭に関連付けられたメモリアドレスを判断することができる。
【００８４】
ビデオデコーダ３０を複数の符号化画素区画を並列して復号するように構成する場合、スライスヘッダー内に入口点、またはスライスデータ内に入口点マーカを含めることは有用である。しかし、ビデオデコーダによっては、複数の符号化画素区画を並列して復号するように構成されない場合がある。複数の符号化画素区画を並列して復号するように構成されていないビデオデコーダが必要とするハードウェアは、複数の符号化画素区画を並列して復号するように構成されたビデオデコーダ３０が必要とするハードウェアよりも単純である。より単純なハードウェアは、より低コストであり、エネルギー消費がより少ないと考えられる。

【００８５】
ビデオデコーダ３０が複数の符号化画素区画を並列して復号するように構成されていない場合、ビデオデコーダ３０は入口点オフセットも入口点マーカも使用しない。したがって、ビデオデコーダ３０が複数の符号化画素区画を並列して復号するように構成されていない場合、ビデオデコーダ３０が受信したビットストリームへの入口点オフセットおよび／または入口点マーカを含めるとビットストリーム中のビットの数を無用に増やすことになる可能性がある。
【００８６】
本開示の１つまたは複数の技術によると、ビデオエンコーダ２０は、ビデオエンコーダ２０がコード化スライスＮＡＬユニットに入口点オフセットおよび／または入口点マーカを含めないように構成してもよい。その代わりに、ビデオエンコーダ２０は、コード化スライスＮＡＬユニット内の符号化画素区画の入口点を示すＳＥＩメッセージを生成してもよい。例えば、ビデオエンコーダ２０は、コード化スライスＮＡＬユニット内の符号化ピクチャ区画のための入口点オフセットを含むＳＥＩメッセージを生成してもよい。言い換えると、ＳＥＩメッセージはコード化ＮＡＬユニット内の符号化ピクチャ区画のバイトオフセットを指定することによって入口点を示すことができる。
【００８７】
図２は、本開示の技術を実装するように構成された例示のビデオエンコーダ２０を示すブロック図である。図２は説明のために示すものであり、本開示で大まかに例示し、説明する技術を限定するものとみなすべきではない。説明のために、本開示ではビデオエンコーダ２０について、ＨＥＶＣコーディングの文脈で説明する。しかし、本開示の技術は他のコーディング標準または方法にも適用可能である。
【００８８】
図２の例では、ビデオエンコーダ２０は複数の機能構成要素を含む。ビデオエンコーダ２０の機能構成要素は、予測処理ユニット１００と、残差生成ユニット１０２と、変換処理ユニット１０４と、量子化ユニット１０６と、逆量子化ユニット１０８と、逆変換処理ユニット１１０と、再構築ユニット１１２と、フィルターユニット１１３と、復号ピクチャバッファ１１４と、エントロピー符号化ユニット１１６とを含む。予測処理ユニット１００は、インター予測処理ユニット１２１とイントラ予測処理ユニット１２６とを含む。インター予測処理ユニット１２１は、動き推定ユニット１２２と動き補償ユニット１２４とを含む。他の例では、ビデオエンコーダ２０の機能構成要素の数はこれより多くても少なくてもよく、異なる機能構成要素を含んでもよい。さらに、動き推定ユニット１２２と動き補償ユニット１２４とは高度に統合されていてもよいが、図２の例では説明のために別々に表されている。
【００８９】
ビデオエンコーダ２０はビデオデータを受信することができる。ビデオエンコーダ２０はビデオデータを様々な発信源から受信することができる。例えば、ビデオエンコーダ２０はビデオ源１８（図１）またはその他の発信源からビデオデータを受信してもよい。ビデオデータは一連のピクチャを表す。ビデオデータを符号化するために、ビデオエンコーダ２０はピクチャの各スライスを符号化してもよい。スライスの符号化の一部として、ビデオエンコーダ２０はスライス内の各ツリーブロックを符号化してもよい。
【００９０】
ビデオエンコーダ２０は、ラスタースキャン順に従ってスライス内のツリーブロックを符号化することができる。言い換えると、ビデオエンコーダ２０は、スライス内のツリーブロックの最上行を左から右に進み、次にツリーブロックの次の行を左から右に進み、このようにしてビデオエンコーダ２０がスライス内の各ツリーブロックを符号化し終わるまで進む順序で、スライスのツリーブロックを符号化することができる。
【００９１】
ラスタースキャン順に従ってツリーブロックを符号化した結果、特定の１つのツリーブロックの上と左のツリーブロックが符号化されているが、その特定のツリーブロックの下

と右のツリーブロックはまだ符号化されていないと考えられる。したがって、ビデオエンコーダ２０は、特定のツリーブロックを符号化するときに、特定のツリーブロックの上と左のツリーブロックを符号化することによって生成された情報にアクセスすることができると考えられる。しかし、ビデオエンコーダ２０は特定のツリーブロックを符号化するときに特定のツリーブロックの下と右のツリーブロックを符号化することによって生成される情報にアクセスすることができないと考えられる。

【００９２】

ツリーブロックの符号化の一部として、予測処理ユニット１００は、ツリーブロックに関連付けられた画素ブロックに対して四分木区画化を行って画素ブロックを漸進的に縮小する画素ブロックに分割することができる。より小さい画素ブロックをＣＵに関連付けてもよい。例えば、予測処理ユニット１００は、ツリーブロックの各画素ブロックを等しい大きさの４つのサブブロックに区画化し、サブブロックのうちの１つまたは複数のサブブロックを大きさの等しい４つのサブブロックに分割するという具合に分割してもよい。

【００９３】

ＣＵに関連付けられる画素ブロックの大きさは、８×８画素から最大６４×６４サンプル以上のサンプルを含むツリーブロックに関連付けられた画素ブロックのサイズまでの範囲とすることができる。本開示では、「Ｎ×Ｎ」と「Ｎ掛けるＮ」とを区別なく使用して例えば、１６×１６画素または１６掛ける１６画素のように画素ブロックの画素寸法を垂直方向および水平方向の寸法で表す。一般に、１６×１６画素ブロックは、垂直方向に１６画素（ｙ＝１６）と水平方向に１６画素（ｘ＝１６）とを有する。同様に、Ｎ×Ｎブロックは一般に、垂直方向にＮ画素と水平方向にＮ画素を有し、Ｎは負でない整数値を表す。

【００９４】

ビデオエンコーダ２０は、ツリーブロックのＣＵを符号化してＣＵの符号化表現（すなわちコード化ＣＵ）を生成することができる。ビデオエンコーダ２０は、ｚスキャン順序に従ってツリーブロックのＣＵを符号化してもよい。言い換えると、ビデオエンコーダ２０は、左上のＣＵと、右上のＣＵと、左下のＣＵと、右下のＣＵとをこの順序で符号化してもよい。ビデオエンコーダ２０が区画化されたＣＵを符号化するとき、ビデオエンコーダ２０は特定のＣＵの画素ブロックのサブブロックに関連付けられたＣＵをｚスキャン順に従って符号化することができる。言い換えると、ビデオエンコーダ２０は左上のサブブロックに関連付けられたＣＵと、右上のサブブロックに関連付けられたＣＵと、左下のサブブロックに関連付けられたＣＵと、右下のサブブロックに関連付けられたＣＵとをこの順序で符号化することができる。

【００９５】

ｚスキャン順に従ってツリーブロックのＣＵを符号化した結果、特定のＣＵの上、左上、右上、左、および左下の各ＣＵが符号化されている可能性がある。その特定のＣＵの下または右のＣＵはまだ符号化されていない可能性がある。したがって、ビデオエンコーダ２０は、その特定のＣＵを符号化するときにその特定のＣＵに隣接する他のいくつかのＣＵを符号化することによって生成される情報にアクセスすることができる可能性がある。しかし、その特定のＣＵを符号化するときにその特定のＣＵに隣接する他のＣＵを符号化することによって生成される情報にはアクセスすることができない可能性がある。

【００９６】

ＣＵの符号化の一部として、予測処理ユニット１００は、ＣＵの１つまたは複数のＰＵのうちのＣＵの画素ブロックを区画化してもよい。ビデオエンコーダ２０とビデオデコーダ３０とは様々なＰＵサイズに対応することができる。特定のＣＵのサイズを２Ｎ×２Ｎと仮定すると、ビデオエンコーダ２０とビデオデコーダ３０とは、イントラ予測の場合は２Ｎ×２ＮまたはＮ×ＮのＰＵサイズに対応することができ、インター予測の場合は２Ｎ×２Ｎ、２×Ｎ、Ｎ×２Ｎ、Ｎ×Ｎ、または同様のＰＵサイズに対応することができる。ビデオエンコーダ２０とビデオデコーダ３０とは、インター予測について２Ｎ×ｎＵ、２Ｎ×ｎＤ、ｎＬ×２Ｎ、およびｎＲ×２ＮのＰＵサイズの非対称区画化にも対応可能であ

る。
【００９７】
インター予測処理ユニット１２１は、ＣＵの各ＰＵに対してインター予測を行うことができる。インター予測により時間的圧縮を実現することができる。インター予測処理ユニット１２１は、ＰＵのために予測データを生成してもよい。ＰＵの予測データは、そのＰＵに対応する予測サンプルブロックと、そのＰＵのための動き情報とを含み得る。動き推定ユニット１２２は、ＰＵのための動き情報を生成することができる。場合よっては、動き推定ユニット１２２はマージモードまたは高度動き予測ベクトル（ａｄｖａｎｃｅｄ ｍｏｔｉｏｎ ｖｅｃｔｏｒ ｐｒｅｄｉｃｔｉｏｎ：ＡＭＶＰ）モードを使用してＰＵの動き情報を通知してもよい。動き補償ユニット１２４は、ＰＵに関連付けられたピクチャ以外の１つまたは複数のピクチャ（すなわち参照ピクチャ）の画素に基づいてＰＵの予測画素ブロックを生成することができる。
【００９８】
スライスは、Ｉスライス、Ｐスライス、またはＢスライスとすることができる。動き推定ユニット１２２と動き補償ユニット１２４とは、ＰＵがＩスライス、Ｐスライス、またはＢスライスのいずれにあるかに応じて、ＣＵのＰＵについて異なる操作を行ってもよい。Ｉスライスでは、すべてのＰＵがイントラ予測される。したがって、ＰＵがＩスライス内にある場合、動き推定ユニット１２２と動き補償ユニット１２４とはそのＰＵに対してはインター予測を行わない。
【００９９】
ＰＵがＰスライス内にある場合、そのＰＵを含むピクチャは、「リスト０」と呼ぶ参照ピクチャのリストに関連付けられる。ある例では、リスト０にリストされている各参照ピクチャは表示順でピクチャの前に現れる。リスト０内の参照ピクチャのそれぞれを、他のピクチャのインター予測に使用することができる。動き推定ユニット１２２は、リスト０内の参照ピクチャでＰスライス内のＰＵのための参照ブロックを検索することができる。ＰＵの参照ブロックは、ＰＵの画素ブロックに最も密接に対応する画素ブロックであってよい。動き推定ユニット１２２は様々な測定基準を使用して、参照ピクチャ内の画素ブロックがＰＵの画素ブロックにどれだけ密接に対応しているかを判断することができる。例えば、動き推定ユニット１２２は、絶対差の和（ＳＡＤ）、平方差の和（ＳＳＤ）、またはその他の差測定基準によって、参照ピクチャ内の画素ブロックがＰＵの画素オブロックにどれだけ密接に対応しているかを判断することができる。
【０１００】
動き推定ユニット１２２は、Ｐスライス内のＰＵの参照ブロックと、そのＰＵと参照ブロックとの間の空間的変位を示す動きベクトルとを含むリスト０内の参照ピクチャを示す参照ピクチャインデックスを生成してもよい。動き推定ユニット１２２は、様々な精度の動きベクトルを生成することができる。例えば、動き推定ユニット１２２は４分の１画素精度、８分の１画素精度、またはその他の分数画素精度の動きベクトルを生成することができる。分数画素精度の場合、参照ピクチャ内の整数位置画素値から参照ブロック値が補間され得る。動き推定ユニット１２２は、参照ピクチャインデックスと動きベクトルとをＰＵの動き情報として出力してもよい。動き補償ユニット１２４は、ＰＵの動き情報に関連付けられた参照ブロックに基づいてＰＵの予測サンプルブロックを生成することができる。
【０１０１】
ＰＵがＢスライス内にある場合、ＰＵを含むピクチャを「リスト０」および「リスト１」と呼ぶ参照ピクチャの２つのリストに関連付けることができる。ある例では、リスト１にリストされる各参照ピクチャは、表示順で現在のピクチャの後に現れる。さらに、ＰＵがＢスライス内にある場合、動き推定ユニット１２２はＰＵのために一方向インター予測または双方向インター予測を行ってもよい。ＰＵのために一方向インター予測を行うには、動き推定ユニット１２２はリスト０またはリスト１の参照ピクチャでＰＵの参照ブロックを検索してもよい。動き推定ユニット１２２は、参照ブロックを含む参照ピクチャのリ

スト０またはリスト１内での位置を示す参照ピクチャインデックスと、ＰＵと参照ブロックとの間の空間的変位を示す動きベクトルとを生成することができる。

【０１０２】

ＰＵの双方向インター予測を行うために、動き推定ユニット１２２はリスト０内の参照ピクチャでもＰＵの参照ブロックを検索することができ、リスト１内の参照ピクチャでＰＵの他の参照ブロックを検索することができる。動き推定ユニット１２２は、これらの参照ブロックを含む参照ピクチャの、リスト０およびリスト１内での位置を示す参照ピクチャインデックスを生成してもよい。さらに、動き推定ユニット１２２は、参照ブロックとＰＵとの間の空間的変位を示す動きベクトルを生成することができる。ＰＵの動き情報は、ＰＵの参照ピクチャインデックスと動きベクトルとを含み得る。動き補償ユニット１２４は、ＰＵの動き情報によって示される参照ブロックに基づいて、ＰＵの予測サンプルブロックを生成することができる。

【０１０３】

ＣＵに対する符号化演算の実行の一部として、イントラ予測処理ユニット１２６はＣＵのＰＵについてイントラ予測を行ってもよい。イントラ予測により空間的圧縮を実現し得る。イントラ予測処理ユニット１２６は、ＰＵと同じピクチャ内の復号サンプルに基づいてＰＵのための予測データを生成することができる。ＰＵのための予測データは、ＰＵのための予測サンプルブロックと様々な構文要素とを含み得る。イントラ予測処理ユニット１２６は、Ｉスライス内と、Ｐスライス内と、Ｂスライス内のＰＵについてイントラ予測を行ってもよい。

【０１０４】

ＰＵについてイントラ予測を行うために、イントラ予測処理ユニット１２６は複数のイントラ予測モードを使用して、ＰＵの複数組の予測データを生成することができる。イントラ予測モードを使用してＰＵの１組の予測データを生成するために、イントラ予測処理ユニット１２６は、そのイントラ予測モードに関連付けられた方向および／または傾きで、隣接ＰＵのサンプルブロックからサンプルをＰＵのサンプルブロック全体にわたって拡張して適用することができる。ＰＵ、ＣＵおよびツリーブロックの左から右、上から下の符号化順序を仮定すると、隣接ＰＵはそのＰＵの上、右上、左上、または左にあり得る。イントラ予測処理ユニット１２６は様々な数のイントラ予測モード、例えば３３方向のイントラ予測モードを使用することができる。ある例では、イントラ予測モードの数はＰＵのサイズに依存し得る。

【０１０５】

予測処理ユニット１００は、ＰＵのためにインター予測処理ユニット１２１によって生成された予測データ、またはＰＵのためにイントラ予測処理ユニット１２６によって生成された予測データの中からＣＵのＰＵのための予測データを選択してもよい。ある例では、予測処理ユニット１００は、これらの組の予測データのレート／歪み測定基準に基づいてＣＵのＰＵのために予測データを選択する。

【０１０６】

予測処理ユニット１００は、四分木区画化を行ってＣＵの残差画素ブロックをサブブロックに区画化してもよい。分割されない各残差画素ブロックをＣＵの異なるＴＵに関連付けてもよい。ＣＵのＴＵに関連付けられた残差画素ブロックのサイズと位置は、ＣＵのＰＵの画素ブロックのサイズと位置とに基づいていてもいなくてもよい。「残差四分木」（ｒｅｓｉｄｕａｌ　ｑｕａｄ－ｔｒｅｅ：ＲＱＴ）と呼ばれる四分木構造は、各残差画素ブロックに関連付けられたノードを含み得る。ＣＵのＴＵはＲＱＴのリーフノードに対応し得る。

【０１０７】

ＴＵの残差画素ブロックの画素はＹサンプルと、Ｃｂサンプルと、Ｃｒサンプルとを含むため、各ＴＵはＹサンプルのサンプルブロックと、Ｃｂサンプルのサンプルブロックと、Ｃｒサンプルのサンプルブロックとに関連付けられ得る。残差生成ユニット１０２は、ＣＵのサンプルブロックの対応するサンプルからＣＵのＰＵの予測サンプルブロックを減

算することによって、ＣＵのための残差サンプルブロックを生成することができる。
【０１０８】
変換処理ユニット１０４は、ＴＵに関連付けられた残差サンプルブロックに１つまたは複数の変換を適用することにより、ＣＵの各ＴＵについて係数ブロックを生成することができる。各係数ブロックは係数の２次元行列とすることができる。変換処理ユニット１０４は、ＴＵに関連付けられた残差サンプルブロックに様々な変換を適用し得る。例えば、変換処理ユニット１０４は、離散コサイン変換（ＤＣＴ）、方向性変換、または概念的に類似した変換をＴＵに関連付けられた残差サンプルブロックに適用し得る。
【０１０９】
量子化ユニット１０６は、ＴＵに関連付けられた係数ブロック内の係数を量子化する。この量子化処理によって、係数のうちの一部または全部に付随するビット深度を低減することができる。例えば、量子化時にｎビットの係数はｍビットの係数に端数切り捨てされ得る。ここで、ｎはｍより大きい。量子化ユニット１０６はＣＵに関連付けられた量子化パラメータ（ＱＰ）値に基づいてＣＵのＴＵに関連付けられた係数ブロックを量子化してもよい。ビデオエンコーダ２０は、ＣＵに関連付けられるＱＰ値を調整することにより、ＣＵに関連付けられた係数ブロックに適用される量子化の程度を調整することができる。
【０１１０】
ビデオエンコーダ２０は、様々な方法でＣＵにＱＰ値を関連付けることができる。例えば、ビデオエンコーダ２０は、ＣＵに関連付けられたツリーブロックについてレート歪み分析を行ってもよい。レート歪み分析では、ビデオエンコーダ２０は、異なるＱＰ値を使用してツリーブロックを複数回符号化することによって、ツリーブロックの複数のコード化表現を生成してもよい。ビデオエンコーダ２０は、ビットレートと歪み測度とが最も低いツリーブロックのコード化表現のＣＵに特定のＱＰ値が関連付けられるとき、その特定のＱＰ値がＣＵに関連付けられることを通知してもよい。
【０１１１】
逆量子化ユニット１０８と逆変換処理ユニット１１０とは、係数ブロックにそれぞれ逆量子化と逆変換とを適用して係数ブロックから残差サンプルブロックを再構築することができる。再構築ユニット１１２は、再構築された残差サンプルブロックを、予測処理ユニット１００によって生成された１つまたは複数の予測サンプルブロックからの対応するサンプルに加えて、ＴＵに関連付けられた再構築残差サンプルブロックを生成する。ビデオエンコーダ２０は、このようにしてＣＵの各ＴＵについてサンプルブロックを再構築することによりＣＵのサンプルブロックを再構築することができる。
【０１１２】
フィルターユニット１１３が、ＣＵに関連付けられたサンプルブロック内のブロッキングアーチファクトを低減するデブロック操作を行ってもよい。フィルターユニット１１３が再構築されたサンプルブロックに対して１つまたは複数のデブロック操作を行った後、復号ピクチャバッファ１１４が再構築サンプルブロックを格納してもよい。動き推定ユニット１２２と動き補償ユニット１２４とが、再構築サンプルブロックを含む参照ピクチャを使用して後続のピクチャのＰＵについてインター予測を行ってもよい。さらに、イントラ予測処理ユニット１２６が復号ピクチャバッファ１１４内の再構築サンプルブロックを使用してＣＵと同じピクチャ内の他のＰＵについてイントラ予測を行ってもよい。
【０１１３】
エントロピー符号化ユニット１１６は、ビデオエンコーダ２０の他の機能構成要素からデータを受け取ることができる。例えば、エントロピー符号化ユニット１１６は、量子化ユニット１０６から係数ブロックを受け取り、予測処理ユニット１００から構文要素を受け取ることができる。エントロピー符号化ユニット１１６は、このデータに１つまたは複数のエントロピー符号化演算を行ってエントロピーコード化データを生成することができる。例えば、ビデオエンコーダ２０は、データに対して、コンテキスト適応型可変長コーディング（ｃｏｎｔｅｘｔ－ａｄａｐｔｉｖｅ ｖａｒｉａｂｌｅ ｌｅｎｇｔｈ ｃｏｄｉｎｇ：ＣＡＶＬＣ）演算、ＣＡＢＡＣ演算、可変－可変（ｖａｒｉａｂｌｅ－ｔｏ－

ｖａｒｉａｂｌｅ：Ｖ２Ｖ）長コーディング演算、構文ベースのコンテキスト適応型２値算術コーディング（ｓｙｎｔａｘ－ｂａｓｅｄ ｃｏｎｔｅｘｔ－ａｄａｐｔｉｖｅ ｂｉｎａｒｙ ａｒｉｔｈｍｅｔｉｃ ｃｏｄｉｎｇ：ＳＢＡＣ）演算、確率区間区画化エントロピー（Ｐｒｏｂａｂｉｌｉｔｙ Ｉｎｔｅｒｖａｌ Ｐａｒｔｉｔｉｏｎｉｎｇ Ｅｎｔｒｏｐｙ：ＰＩＰＥ）コーディング演算、指数ゴロムコーディング、またはその他の種類のエントロピー符号化演算を行ってもよい。エントロピー符号化ユニット１１６は、このエントロピー符号化データを含むビットストリームを出力することができる。

【０１１４】

データに対するエントロピー符号化演算の実行の一部として、エントロピー符号化ユニット１１６はコンテキストモデルを選択してもよい。エントロピー符号化ユニット１１６がＣＡＢＡＣ演算を行う場合、コンテキストモデルは特定のビンが特定の値を有する確率の推定値を示すことができる。ＣＡＢＡＣの文脈では、「ビン」という用語を使用して構文要素の２値化表現の１ビットを指すことがある。

【０１１５】

前述のように、ビデオエンコーダ２０はビデオデータの符号化表現を含むビットストリームを生成することができる。ビデオデータの符号化表現は、複数のＮＡＬユニットを含み得る。ＮＡＬユニットは、スライスのコード化表現を含むコード化スライスＮＡＬユニットを含み得る。スライスは複数のピクチャ区画（例えばエントロピースライス、タイル、またはＷＷＰ波）に関連付けられたツリーブロックを含み得るため、コード化スライスＮＡＬユニットは複数の符号化ピクチャ区画を含み得る。前述のように、符号化ピクチャ区画は単一のピクチャ区画に関連付けられた、コード化スライスＮＡＬユニット内のコード化ツリーブロックのグループとすることができる。

【０１１６】

各コード化スライスＮＡＬユニットは、スライスヘッダーとスライスデータとを含み得る。コード化スライスＮＡＬユニットのスライスデータは、スライス内の各ツリーブロックのコード化表現を含み得る。場合によっては、スライスのツリーブロックは異なるピクチャ区画に関連付けられ得る。したがって、単一のコード化スライスＮＡＬユニットのスライスデータが、複数のピクチャ区画に関連付けられた符号化ツリーブロックを含み得る。前述のように、「符号化画素区画」および「コード化サブスライス」という用語を使用して、単一ピクチャ区画に関連付けられたコード化スライスＮＡＬユニット内のコード化ツリーブロックのグループを指すことがある。

【０１１７】

ビデオデコーダ３０を、複数の符号化ピクチャ区画を並列して構文解析または復号するように構成してもよい。複数の符号化ピクチャ区画の同時構文解析または同時復号を容易にするため、各符号化ピクチャ区画はバイト整列され得る。すなわち、符号化ピクチャ区画の先頭のビットがメモリ内のアドレス指定可能バイトの先頭のビットになり、符号化ピクチャ区画の最後のビットがメモリ内のアドレス指定可能バイトの最後のビットになる。

【０１１８】

コード化スライスＮＡＬユニットのスライスデータ内の符号化ピクチャ区画が確実にバイト整列されるようにするため、本開示の技術により、コード化スライスＮＡＬユニットのスライスヘッダーはバイト境界で終結してもよい。場合によっては、スライスヘッダーの先頭と最後の両方がバイト整列されてもよい。例えば、ビデオエンコーダ２０は、複数の符号化バイト整列エントロピースライス、タイル、またはＷＷＰ波（すなわち符号化ピクチャ区画）内のツリーブロックを含むスライスのバイト整列スライスヘッダーを持つコード化スライスＮＡＬを生成してもよい。

【０１１９】

ある例では、ビデオエンコーダ２０は、ＳＰＳ、ＰＰＳ、またはＡＰＳなどのパラメータセットにフラグ（例えばｓｌｉｃｅ＿ｈｅａｄｅｒ＿ｂｙｔｅ＿ａｌｉｇｎｅｄ＿ｆｌａｇ）を含めてもよい。フラグが１の場合、パラメータセットを参照する各コード化スライスＮＡＬユニットのスライスヘッダーがバイト整列される。このような例では、フラグ

が０の場合、パラメータセットを参照する各コード化スライスＮＡＬユニットのスライスヘッダーはバイト整列される場合とされない場合がある。他の例では、このフラグの意味を逆にしてもよい。他の例では、コード化スライスＮＡＬユニットのスライスヘッダーは常にバイト整列される。これは、上記のフラグが存在しないのと等しく、フラグの値が常に１であると推定されるのと等しい。

【０１２０】

このようにして、ビデオエンコーダ２０はパラメータセットを含むＮＡＬユニットを生成することができ、ビデオデコーダ３０はそれを受信することができる。異なる例では、パラメータセットはＳＰＳ、ＰＰＳ、またはＡＰＳであってもよい。パラメータセットは、そのパラメータセットを参照するスライスヘッダーがバイト整列されているか否を示す構文要素を含み得る。

【０１２１】

ビデオエンコーダ２０がバイト整列スライスヘッダーを生成する例では、スライスヘッダーは様々な構文規則に準拠し得る。例えば、ビデオエンコーダ２０は、以下の表１に示す例示の構文規則に準拠したスライスヘッダーを生成してもよい。

【表１】

表1:スライスヘッダー構文規則

slice_header( ) {	記述子
...	
if(slice_header_byte_aligned_flag)	
while(!byte_aligned( ))	
bit_equal_to_one	f(1)
}	

【０１２２】

説明の便宜上、表１の例示の構文規則では既存の構文要素を省略している。表１の例示の構文規則で、ｂｉｔ＿ｅｑｕａｌ＿ｔｏ＿ｏｎｅ構文要素は、値１を有するビットである。ｂｙｔｅ＿ａｌｉｇｎｅｄ（）関数は、スライスヘッダーがバイト整列されているか否かを示す値を返す。

【０１２３】

さらに、本開示の技術によると、ビットストリームは、コード化スライスＮＡＬユニット内のコード化ピクチャ区画の入口点を示す１つまたは複数のＳＥＩメッセージを含み得る。説明の便宜上、本開示ではコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージを「入口点ＳＥＩメッセージ」と呼ぶことがある。

【０１２４】

ある例では、ＳＥＩメッセージは、ＳＥＩメッセージのペイロードのタイプを指定するｐａｙｌｏａｄＴｙｐｅ構文要素を含む。そのようなある例では、入口点ＳＥＩメッセージのｐａｙｌｏａｄＴｙｐｅ構文要素は、他のタイプのＳＥＩメッセージのｐａｙｌｏａｄＴｙｐｅ構文要素とは異なり得る。したがって、入口点ＳＥＩメッセージは、入口点ＳＥＩメッセージが、コード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含み得る。

【０１２５】

ビットストリームは、入口点ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを含み得る。ある例では、ＳＥＩ　ＮＡＬユニット内の入口点ＳＥＩメッセージは、そのＳＥＩ　ＮＡＬユニットの直後のコード化スライスＮＡＬユニットに適用され得る。したがって、

ビデオデコーダ３０は、ビットストリーム内で複数の符号化ピクチャ区画を含むＮＡＬユニットの直前のＳＥＩ　ＮＡＬユニットに少なくとも部分的に基づいて、ＳＥＩメッセージが複数の符号化ピクチャ区画を含むＮＡＬユニットに適用可能であると判断することができる。説明の便宜上、ＳＥＩメッセージが適用され得るコード化スライスＮＡＬユニットを、本開示では入口点ＳＥＩメッセージのコード化スライスＮＡＬユニットと呼ぶ。

【０１２６】

ある例では、ＳＥＩメッセージはＳＥＩメッセージが適用可能なユニットを示すレベルがＳＥＩメッセージに関連付けられる。入口点ＳＥＩメッセージは、入口点ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットに続くコード化スライスＮＡＬユニットに適用可能としてもよい。あるいは、入口点ＳＥＩメッセージは、入口点ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットに続くコード化スライスＮＡＬユニットに含まれるコード化スライスに適用可能としてもよい。したがって、入口点ＳＥＩメッセージは、ＮＡＬユニットレベルＳＥＩメッセージまたはスライスレベルＳＥＩメッセージであり得る。

【０１２７】

さらに、ある例では、コード化スライスＮＡＬユニットがピクチャの最初のビデオコーディングレイヤ（ＶＣＬ）ＮＡＬユニットである場合、コード化スライスＮＡＬユニットに適用可能な入口点ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットは、１０であるＮＡＬユニットタイプを有してもよい。一方、コード化スライスＮＡＬユニットがピクチャの最初のＶＣＬ　ＮＡＬユニットでない場合、コード化スライスＮＡＬユニットに適用可能な入口点ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットは、１１であるＮＡＬユニットタイプを有してもよい。「ＶＣＬ　ＮＡＬユニット」は、ＶＣＬ　ＮＡＬユニットに分類されるｎａｌ＿ｕｎｉｔ＿ｔｙｐｅの予約値を有するコード化スライスＮＡＬユニットおよびＮＡＬユニットのサブセットを指す総称である。ＶＣＬ　ＮＡＬユニットではないタイプのＮＡＬユニットを、非ＶＣＬ　ＮＡＬユニットと呼ぶ場合がある。非ＶＣＬ　ＮＡＬユニットのタイプの例としては、ＳＰＳ　ＮＡＬユニット、ＰＰＳ　ＮＡＬユニット、ＳＥＩ　ＮＡＬユニットなどがある。ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅは、ＮＡＬユニットのＮＡＬユニットタイプを指定する、ＮＡＬユニットのヘッダー内の構文要素である。

【０１２８】

ビデオエンコーダ２０は様々な構文規則を有する入口点ＳＥＩメッセージを生成することができる。例えば、ビデオエンコーダ２０は、以下の表２に示す構文規則を有する入口点ＳＥＩメッセージを生成してもよい。

【表２】

表2:入口点SEIメッセージの構文規則

entry_offset(payloadSize) {	**記述子**
offset_len_minus10	ue(v)
num_offsets_minus1	ue(v)
for(i = 0; i <= num_substreams_minus1; i++)	
substream_offset[i]	u(v)
}	

【０１２９】

表２の例示の構文規則において、ｏｆｆｓｅｔ＿ｌｅｎ＿ｍｉｎｕｓ１０構文要素に１０を加えた値によって、ｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｉ］構文要素の長さをビット数で指定する。このようにして、オフセット構文要素のそれぞれの長さは、ｏｆｆｓｅｔ＿ｌｅｎ＿ｍｉｎｕｓ１０構文要素の値に１０を加えた値に等しい。さらに、表２の例

示の構文規則において、ｎｕｍ＿ｏｆｆｓｅｔｓ＿ｍｉｎｕｓ１構文要素に１を加えた値によって、入口点ＳＥＩメッセージ内のｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｉ］構文要素の数を指定する。

【０１３０】

表２の例示の構文規則において、ｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｉ］構文要素は、ｉ番目のオフセットをバイト数で指定する。入口点ＳＥＩメッセージのコード化スライスＮＡＬユニットは、符号化ピクチャ区画インデックス値０、１、２．．．、ｎｕｍ＿ｏｆｆｓｅｔｓ＿ｍｉｎｕｓ１＋１を有する、ｎｕｍ＿ｏｆｆｓｅｔｓ＿ｍｉｎｕｓ１プラス２個の符号化ピクチャ区画からなり得る。符号化ピクチャ区画０は、入口点ＳＥＩメッセージのコード化スライスＮＡＬユニットのバイト０からｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［０］－１までからなり得る。ｋが１からｎｕｍ＿ｏｆｆｓｅｔｓ＿ｍｉｎｕｓ１＋１までの範囲である符号化ピクチャ区画ｋは、入口点ＳＥＩメッセージのコード化スライスＮＡＬユニットのバイトｓｕｂｓｔｒｅａｍ＿ｏｆｆｅｔ［ｋ－１］からｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｋ］－１までからなり得る。

【０１３１】

このようにして、ビデオエンコーダ２０は、それに基づいてオフセット構文要素のビット数で表された長さを判断可能な構文要素を通知することができる。さらに、ビデオエンコーダ２０は、それに基づいて一連のオフセット構文要素内のオフセット構文要素の数を判断可能な構文要素を通知することができる。さらに、ビデオエンコーダ２０は、一連のオフセット構文要素内のオフセット構文要素が、ＮＡＬユニット内の符号化画素区画のバイトオフセットを示す、一連のオフセット構文要素を通知することができる。同様に、ビデオデコーダ３０は、一連のオフセット構文要素のビット数で表した長さを指定する構文要素をビットストリームで受信することができる。ビデオデコーダ３０は一連のオフセット構文要素内のオフセット構文要素の数を指定する構文要素もビットストリームで受信することができる。さらに、ビデオデコーダ３０は、一連のオフセット構文要素内のオフセット構文要素がＮＡＬユニット内の符号化ピクチャ区画のバイトオフセットを示す、一連のオフセット構文要素をビットストリームで受信することができる。

【０１３２】

表２の例示の構文規則とは別の例示の構文規則では、ｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｉ］の長さは、１バイトと２バイトと３バイトと４バイトとのうちの１つである。ｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｉ］の長さは、１バイトと、２バイトと、３バイトと、４バイトとにそれぞれ対応する値０、１、２または３を有する構文要素によって示すことができる。このような例では、表２のｏｆｆｓｅｔ＿ｌｅｎ＿ｍｉｎｕｓ１０構文要素をこの構文要素に置き換えることができる。さらに、入口点ＳＥＩメッセージのその他の様々な例示の構文規則において、表２のｏｆｆｓｅｔ＿ｌｅｎ＿ｍｉｎｕｓ１０構文要素がｏｆｆｓｅｔ＿ｌｅｎ＿ｍｉｎｕｓ８構文要素に置き換えられｓｕｂｓｔｒｅａｍ＿ｏｆｆｓｅｔ［ｉ］構文要素の最小の長さは８ビットである。

【０１３３】

前述のように、複数のピクチャ区画（例えば、エントロピースライス、コード化タイル、または波面波）にわたるピクチャ内予測が許容され得る。したがって、１つの符号化ピクチャ区画の復号は、他の符号化ピクチャ区画の復号に依存し得る。ある例では、符号化ピクチャ区画が異なるＮＡＬユニットにカプセル化され、それらのＮＡＬユニットは異なるパケット、例えばリアルタイムトランスポートプロトコル（ＲＴＰ）パケットにカプセル化される。異なるＮＡＬユニットが依存符号化ピクチャ区画を含み得るため、これらのＮＡＬユニットうちの１つの損失によって、ビデオデコーダ３０は他のＮＡＬユニット内の符号化区画を復号することができなくなる可能性がある。したがって、１つのパケットに複数のＮＡＬユニットを含めると有利になり得る。ある種の従来のＣＯＤＥＣでは、パケタイザーが、コード化スライスＮＡＬユニットのＮＡＬユニットヘッダー内の情報から、コード化スライスＮＡＬユニットが符号化ピクチャ区画を含むことを判断することができない場合がある。そのようなＣＯＤＥＣでは、コード化スライスＮＡＬユニットが符号

化ピクチャ区画を含むことを示す情報がコード化スライスＮＡＬユニットのスライスヘッダー内にのみ存在することがある。コード化スライスＮＡＬユニットのスライスヘッダーはエントロピー符号化される。したがって、コード化スライスＮＡＬユニットがコード化ピクチャ区画を含むことを判断するには、パケタイザーがコード化スライスＮＡＬユニットのスライスヘッダーをエントロピー復号する必要がある場合があり、それによってパケタイザーが複雑化する。

【０１３４】

パケタイザーの複雑さを低減するために、ピクチャ区画境界を越えたピクチャ内予測が可能な単一の符号化ピクチャ区画を含むＮＡＬユニットに、異なるＮＡＬユニットタイプを割り振ってもよい。さらに、異なる適用範囲を有する情報を含むＳＥＩ　ＮＡＬユニットに異なるユニットタイプを割り振ってもよい。

【０１３５】

ビデオエンコーダ２０は様々な方法でＮＡＬユニットのタイプを示すことができる。例えば、ビデオエンコーダ２０は、ＮＡＬユニットのＮＡＬユニットヘッダーに「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素を含めてそのＮＡＬユニットのタイプを示してもよい。ある例では、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素は６ビットとすることができる。このような例では、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素によって指定される値は０から６３までの範囲であってもよい。本開示では、範囲は別に記載のない限り範囲の両端を含むものとする。「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が０、または４８から６３までの範囲であるＮＡＬユニットは、復号処理に影響を与えないものとしてもよい。「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が０、または４８から６３までの範囲であるＮＡＬユニットは、アプリケーションによって決まる使い方をしてもよい。「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が０、または４８から６３までの範囲であるＮＡＬユニットには復号処理が指定されない。異なるアプリケーションが「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が０、または４８から６３までの範囲であるＮＡＬユニットを異なる目的に使用してもよい。したがって、異なる適用範囲を有する情報を含むＳＥＩ　ｎａｌユニットに異なるＮＡＬユニットタイプが割り当てられる。他の例では、ＨＥＶＣ　ＷＤ４の場合のように、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素は５ビットであってもよい。「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が５ビットである例では、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の値の範囲は０から３１までであってもよい。

【０１３６】

以下に示す表３に、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の値と、ＮＡＬユニットの内容およびＮＡＬユニットのローバイトシーケンスペイロード（Ｒａｗ　Ｂｙｔｅ　Ｓｅｑｕｅｎｃｅ　Ｐａｙｌｏａｄ：ＲＢＳＰ）構文構造と、ＮＡＬユニットタイプクラスとの関係の一例を示す。

【表３】

表3

nal_unit_type	NALユニットの内容およびRBSP構文構造	NALユニットタイプクラス
0	指定なし	非VCL
1	非IDRおよび非CRAピクチャのコード化スライス	VCL
2	非IDRおよび非CRAピクチャのコード化サブスライス	VCL
3	CRAピクチャのコード化スライス	VCL
4	CRAピクチャのコード化サブスライス	VCL
5	IDRピクチャのコード化スライス	VCL
6	IDRピクチャのコード化サブスライス	VCL
7	シーケンスパラメータセット	non-VCL
8	ピクチャパラメータセット	non-VCL
9	アダプテーションパラメータセット	non-VCL
10	補助強化情報(SEI)	non-VCL
11	サブピクチャ補助強化情報(SEI)	non-VCL
12	アクセスユニットデリミタ	non-VCL
13	フィラーデータ	non-VCL
13-47	予約	n/a
48..63	該当なし	non-VCL

【０１３７】

表３の例において、１から６までの範囲の「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素を有するＮＡＬユニットはコード化スライスＮＡＬユニットである。ＮＡＬユニットが、特定のピクチャのスライスまたはサブスライスに関連付けられたデータを含み、そのＮＡＬユニットの「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が５または６である場合、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が１から４であるＮＡＬユニットはその特定のピクチャに関連付けられたデータを含み得ない。ＮＡＬユニットが特定のピクチャのスライスまた

はサブスライスに関連付けられたデータを含み、そのＮＡＬユニットの「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が５または６である場合、ピクチャは、インスタンテイニアスデコーダリフレッシュ（Ｉｎｓｔａｎｔａｎｅｏｕｓ　Ｄｅｃｏｄｅｒ　Ｒｅｆｒｅｓｈ：ＩＤＲ）ピクチャであり得る。表３に示すように、コード化サブスライスを含むＮＡＬユニットと、コード化サブスライスを含まないコード化スライスＮＡＬユニットは、同じ構文に準拠するＲＢＳＰを含み得る。

【０１３８】

ＮＡＬユニットが「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素２、４、または６を有する場合、そのＮＡＬユニットはサブスライスを含む。例えば、ＮＡＬユニットが「ｌｉｇｈｔｗｅｉｇｈｔ＿ｓｌｉｃｅ＿ｆｌａｇ」１（すなわち軽量スライス）を有するスライスに関連付けられたデータを含む場合、そのＮＡＬユニットは値２、４、または６の「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素を有することができる。説明の便宜上、本開示では「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」が２、４、または６のＮＡＬユニットをサブスライスＮＡＬユニットと呼ぶ。したがって、コード化サブスライスを含むＮＡＬユニットは、コード化サブスライスを含まないコード化スライスＮＡＬユニットとは異なるＮＡＬユニットタイプに属し得る。例えば、コード化サブスライスを含むＮＡＬユニットは、コード化サブスライスを含まないコード化スライスＮＡＬユニットのタイプ構文要素によって指定される数値とは異なる数値を指定するタイプ構文要素を含み得る。復号時、ビデオデコーダ３０は、サブスライス内のピクチャ内予測を行うために他のスライスまたはサブスライスからのデータを必要とする場合がある。

【０１３９】

ＮＡＬユニットが、１０または１１の「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素を有する場合、そのＮＡＬユニットはＳＥＩ　ＮＡＬユニットである。「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が１０のＳＥＩ　ＮＡＬユニットは、１つまたは複数のコード化ピクチャ全体に適用可能なＳＥＩメッセージを含み得る。「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が１０のＳＥＩ　ＮＡＬユニットは、コード化ピクチャの最初のＶＣＬ　ＮＡＬユニットに先行することができる。

【０１４０】

「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が１１のＳＥＩ　ＮＡＬユニットを本開示ではサブピクチャＳＥＩ　ＮＡＬユニットと呼ぶことがある。サブピクチャＳＥＩ　ＮＡＬｕｎｉｔは、コード化ピクチャの１つまたは複数の連続するＶＣＬ　ＮＡＬユニットのリストに適用可能なＳＥＩメッセージを含むことができる。このリストは、コード化ピクチャのすべてのＶＣＬ　ＮＡＬユニットの真のサブセットを含むことができ、コード化ピクチャの最初のＶＣＬ　ＮＡＬユニットを含まない。サブピクチャＳＥＩ　ＮＡＬユニットは、サブピクチャＳＥＩ　ＮＡＬユニットに含まれるＳＥＩメッセージが適用されるＶＣＬ　ＮＡＬユニットのリストの最初のＶＣＬ　ＮＡＬユニットに先行することができる。

【０１４１】

他の例では、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の値は表３に示すものとは異なる意味を持っていてもよい。例えば、他の例では、イントラスライス（すなわちＩスライス）とインタースライス（すなわちＰスライスおよびＢスライス）は、異なるＮＡＬユニットタイプを有し得る。言い換えると、Ｉスライスについては「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の第１の値があり、ＰスライスとＢスライスについては「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の第２の値があり得る。他の例では、Ｉスライスについては「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の第１の値があり、Ｐスライスについては「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の第２の値があり、Ｂスライスについては「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の第３の値があってもよい。さらに他の例では、短いスライス（すなわち「ｌｉｇｈｔｗｅｉｇｈｔ＿ｓｌｉｃｅ＿ｈｅａｄｅｒ」構文要素が１のスライス）を表すデータを含むＮＡＬユニットとフルスライス（すなわち「ｌｉｇｈｔｗｅｉｇｈｔ＿ｓｌｉｃｅ＿ｈｅａｄｅｒ」構文要素が０であるスライス）を表すデータを含むＮＡＬユニットについて、「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の異なる値があっ

てもよい。

【０１４２】

表３に代わる他の例では、ＳＥＩメッセージがいくつかのレベルに分類されていてもよい。各レベルは「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の異なる値に関連付けることができる。ＳＥＩメッセージのレベルには次のものが含まれ得る。

【０１４３】

１）シーケンスレベル：１つまたは複数のコード化ビデオシーケンス全体に適用可能。

【０１４４】

２）サブシーケンスレベル：複数のコード化ピクチャだが１未満のコード化ビデオシーケンスに適用可能。

【０１４５】

３）ピクチャレベル：１つのアクセスユニット内のすべてのＶＣＬ　ＮＡＬユニットに適用可能。

【０１４６】

４）サブピクチャレベル：アクセスユニット内のすべてのＶＣＬ　ＮＡＬユニットの真のサブセットに適用可能。例えば、サブピクチャレベルのＳＥＩメッセージは、ある対象領域をカバーするＶＣＬ　ＮＡＬユニット、Ｈ．２６４／ＡＶＣ標準のＡｎｎｅｘ　Ｇで定義されているレイヤ表現のＶＣＬ　ＮＡＬユニット、Ｈ．２６４／ＡＶＣ標準のＡｎｎｅｘ　Ｈで定義されているビューコンポーネントのＶＣＬ　ＮＡＬユニット、またはＨ．２６４／ＡＶＣ標準によって定義されている冗長コード化ピクチャのＶＣＬ　ＮＡＬユニットに適用可能とすることができる。

【０１４７】

ビットストリームを一連のアクセスユニットに分割することができる。各アクセスユニットは、１つのコード化ピクチャと、０以上の非ＶＣＬ　ＮＡＬユニットとを含み得る。ビットストリームの先頭のアクセスユニットは、ビットストリームの先頭のＮＡＬユニットから始まる。コード化ピクチャの最後のＶＣＬ　ＮＡＬユニットの後にある以下に示すＮＡＬユニットのうちのいずれかの最初のＮＡＬユニットは、新しいアクセスユニットの開始を指定することができる。

【０１４８】

・アクセスユニットデリミタＮＡＬユニット（ある場合）
・シーケンスパラメータセットＮＡＬユニット（ある場合）
・ピクチャパラメータセットＮＡＬユニット（ある場合）
・アダプテーションパラメータセットＮＡＬユニット（ある場合）
・「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」が１０のＳＥＩ　ＮＡＬユニット（ある場合）
・「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」が１４から２９までの範囲のＮＡＬユニット（ある場合）
・コード化ピクチャの最初のＶＣＬ　ＮＡＬユニット（常に存在する）

ビデオエンコーダ２０とビデオデコーダ３０とは、アクセスユニット内のコード化ピクチャと非ＶＣＬ　ＮＡＬユニットの順序に関する以下の制約に従い得る。

【０１４９】

・アクセスユニットデリミタＮＡＬユニットがある場合、アクセスユニットデリミタＮＡＬは先頭ＮＡＬユニットである。いずれのアクセスユニット内でもアクセスユニットデリミタＮＡＬは１つしかあり得ない。

【０１５０】

・ＳＥＩ　ＮＡＬユニットの「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が１０の場合、そのＳＥＩ　ＮＡＬユニットはコード化ピクチャの先頭ＶＣＬ　ＮＡＬユニットに先立つ。

【０１５１】

・ＳＥＩ　ＮＡＬユニットの「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が１１の場合、そのＳＥＩ　ＮＡＬユニットはコード化ピクチャの先頭ＶＣＬ　ＮＡＬユニットの後に続

き、ＳＥＩ　ＮＡＬユニットが適用される先頭ＶＣＬ　ＮＡＬユニットに先立つ。したがってそのＳＥＩ　ＮＡＬユニットはコード化ピクチャの最後のＶＣＬ　ＮＡＬユニットに先立つ。

【０１５２】

・ＳＥＩ　ＮＡＬユニットがバッファリング期間ＳＥＩメッセージを含む場合、そのＳＥＩ　ＮＡＬユニットは「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素１０を有することができ、バッファリング期間ＳＥＩメッセージはアクセスユニット内の先頭ＳＥＩ　ＮＡＬユニットの先頭ＳＥＩメッセージペイロードである。

【０１５３】

・「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素が０、１１、１３、または、３０から６３までの範囲にあるＮＡＬユニットは、コード化ピクチャの先頭ＶＣＬ　ＮＡＬユニットに先立たない。ＳＰＳ　ＮＡＬユニット、ＰＰＳ　ＮＡＬユニット、またはＡＰＳ　ＮＡＬユニットは、アクセスユニット内に存在してもよいが、そのアクセスユニット内の一次符号化ピクチャの最後のＶＣＬ　ＮＡＬユニットの後に続き得ない。

【０１５４】

このようにして、ビデオエンコーダ２０はビデオデータを符号化する方法を実装することができる。この方法は、複数のＮＡＬユニットを生成することを備え得る。この方法は、ＮＡＬユニットのＮＡＬユニットタイプに基づいてコード化サブスライスを含むＮＡＬユニットのサブセットを識別することも含み得る。さらに、この方法は、ＮＡＬユニットのサブセットを含むパケットを生成することを含み得る。同様に、ビデオデコーダ３０は、各ＮＡＬユニットがコード化サブスライスを含み、各ＮＡＬユニットが、そのＮＡＬユニットがコード化サブスライス（すなわち符号化ピクチャ区画）を含むＮＡＬユニットのタイプに属することを示すタイプ標識を含む、複数のＮＡＬユニットを含むパケットを受信することを備えるビデオデータの復号方法を実装することができる。このビデオデータ復号方法は、サブスライスを復号することも含み得る。

【０１５５】

他の例では、ビデオエンコーダ２０は、複数のサブスライスに関連付けられたデータを含む単一のＮＡＬユニットを生成してもよい。例えば、ビデオエンコーダ２０は複数の軽量スライスに関連付けられたデータを含む単一のコード化スライスＮＡＬユニットを生成してもよい。このような例では、ＮＡＬユニット内の軽量スライスはバイト整列されてよく、ビデオエンコーダ２０はサブスライスのうちの先頭のサブスライスに関連付けられたデータのほか、各サブスライスに関連付けられたデータのバイトオフセットを通知してもよい。ビデオデコーダ３０は、バイトオフセットを使用して、復号のために異なる処理コアに対して異なるサブスライスを割り振ることができる。このようにして、ビデオデコーダ３０はサブスライスに関連付けられたデータを並列して復号することができる。しかし、ビデオデコーダ３０がサブスライスを並列して復号しない場合、ビデオデコーダ３０はバイトオフセットを無視してもよい。したがって、ビデオデコーダ３０がサブスライスを並列して復号しない場合は、バイトオフセットを送るのは非効率的な場合がある。

【０１５６】

ある例では、タイルを使用して段階的復号リフレッシュ（ｇｒａｄｕａｌ　ｄｅｃｏｄｉｎｇ　ｒｅｆｒｅｓｈ：ＧＤＲ）を実現してもよい。例えば、ビデオエンコーダ２０はピクチャシーケンス内の各ピクチャを同じ方法でタイルに区画化してもよい。例えば、ビデオエンコーダ２０は同じ９本の垂直タイル境界を使用して各ピクチャを８個のタイルに区画化してもよい。この例では、左端のタイルがタイル０、左から２番目のタイルがタイル１という具合になってもよい。各ピクチャは少なくとも１つのＰスライスを含む。したがって、シーケンス内の各ピクチャの少なくとも一部が少なくとも１つの他のピクチャに依存する。シーケンスの先頭ピクチャ（すなわちピクチャ０）内では、タイル０内のツリーブロックがすべてイントラコーディングされ、先頭ピクチャの残りのすべてのタイル内のツリーブロックがインターコーディングされる。シーケンスの２番目のピクチャ（すなわちピクチャ１）内では、タイル１内のツリーブロックがすべてイントラコーディングさ

れ、２番目のピクチャの残りのすべてのタイル内のツリーブロックがインターコーディングされる。このパターンがシーケンス内の全ピクチャに対して続いてもよい。言い換えると、ピクチャのシーケンスのピクチャＮ内で、０からシーケンス内のピクチャの数から１を引いた数までの範囲の任意の値Ｎについて、タイルＮ／８内のすべてのツリーブロックがイントラコーディングされ、他のすべてのタイル内のツリーブロックがインターコーディングされる。前文中の「／」は剰余除算（すなわちモジュロ演算子）を示す。復号がそのピクチャから開始する場合、次の７個のピクチャのほかすべてのピクチャが完全に復号されるという点で、Ｎ／８が０であるインデックス値を持つ各ピクチャをランダムアクセス点として使用することができる。他の例では、他の数のタイルを使用してもよい。例えば、ビデオエンコーダ２０は１つのピクチャにつき２つのタイルのみを使用するＧＤＲを実装してもよい。

【０１５７】

ランダムアクセスとは、コード化ビットストリームを、ビデオストリーム内の任意のピクチャから、特に先頭のコード化ピクチャではないコード化ピクチャから復号することを指す。復号がｐｉｃＰから開始するときに、復号順と出力順の両方においてコード化ピクチャｐｉｃＱの後続のすべてのコード化ピクチャを正しく復号することができる場合、コード化ピクチャｐｉｃＰはランダムアクセス点（ＲＡＰ）である。ｐｉｃＱはｐｉｃＰと同じであってもなくてもよい。ＲＡＰは、ＩＤＲピクチャでも非ＩＤＲピクチャでもよい。復号順でＩＤＲピクチャの後続のすべてのコード化ピクチャを、復号順でＩＤＲピクチャに先行する任意のピクチャからインター予測なしに復号することができる。非ＩＤＲピクチャはＩＤＲピクチャではない任意のピクチャである。

【０１５８】

非ＩＤＲランダムアクセス点は、以下の条件が真であるコード化ピクチャである。

【０１５９】

・非ＩＤＲランダムアクセス点（ｐｉｃＲ）はＩＤＲピクチャではない。

【０１６０】

・ｐｉｃＲのピクチャ順カウント（ＰＯＲ）をｒＰｏｃとし、ｐｉｃＡを復号順と出力順の両方において同じコード化ビデオシーケンス内のｐｉｃＲの後続ピクチャとし、ｐｉｃＡのＰＯＣをａＰｏｃとする。ｐｉｃＲでランダムアクセスを行った場合（すなわちｐｉｃＲから復号を開始した場合）、同じコード化ビデオシーケンス内にあり、出力順でｐｉｃＡの後にあるすべてのピクチャを正しく復号することができる。

【０１６１】

非ＩＤＲ　ＲＡＰは、ランダムアクセスをその非ＩＤＲ　ＲＡＰで行ったときに、非ＩＤＲ　ＲＡＰと同じコード化ビデオシーケンス内にあって出力順でその非ＩＤＲ　ＲＡＰの後にあるすべてのピクチャを正しく復号することができる場合に、クリーンなランダムアクセス（ＣＲＡ）ピクチャである。これが真でない場合、その非ＩＤＲ　ＲＡＰを段階的復号リフレッシュ（ＧＤＲ）ピクチャと呼ぶことができる。

【０１６２】

ＧＤＲシーケンス内の各ピクチャはそれ自体のＰＰＳを有することができる。各ＰＰＳはタイル境界を指定する。言い換えると、タイル境界を示す構文要素がＧＤＲシーケンス上の各ピクチャのＰＰＳにおいて複製され得る。これはビットの浪費となる可能性があり、インバンドＰＰＳ更新が必要になり得、それによって誤り耐性問題が生じる可能性がある。

【０１６３】

ある例では、ビデオエンコーダ２０は、ピクチャが２つのタイルしか含まないタイル境界の変更を可能にしてもよい。例えば、ビデオエンコーダ２０はピクチャのシーケンス内の各ピクチャを１対のタイルに区画化してもよい。この場合、ビデオエンコーダ２０は、ピクチャ境界にはない単一の垂直方向または水平方向のタイル境界を持つピクチャのみのタイル境界の変更を可能にし得る。タイル境界をピクチャからピクチャに移動することができる。例えば、タイル境界が垂直の場合、タイル境界は左から右または右から左に移動

可能である。タイル境界が水平の場合、タイル境界は上方または下方に移動可能である。

【０１６４】

タイル境界の移動は様々な方法で通知することができる。例えば、ビデオエンコーダ２０は、シーケンス内のピクチャからピクチャへのタイル境界の移動を示す１つまたは複数の構文要素を生成してもよい。このような構文要素は、様々な構文構造内に配置可能である。例えば、ビデオエンコーダ２０は、境界の移動と位置とを示す構文要素の少なくとも一部を含むパラメータセットを生成してもよい。この例では、パラメータセットはＳＰＳ、ＰＰＳ、ＡＰＳ、またはその他のタイプのパラメータセットとすることができる。この種のある例では、タイル境界の移動と位置とが、ＳＰＳおよび／またはＰＰＳ内の構文要素と組み合わせて示されてもよい。他の例では、タイル境界の移動と位置とは、ＡＰＳまたはスライスヘッダー内の１つまたは複数の構文要素によってのみ示される。他の例では、ビデオエンコーダ２０は、境界の移動と位置とを示す１つまたは複数の構文要素を含むスライスヘッダーを生成してもよい。

【０１６５】

他の例では、ＳＰＳ内の構文要素とスライスヘッダーとで境界の移動と位置とを示す。表４にこの例におけるＳＰＳ　ＲＢＳＰの構文規則を示す。表５にスライスヘッダーの構文規則を示す。表４のＳＰＳ　ＲＢＳＰにおける構文要素と表５のスライスヘッダーとが組み合わさってタイル境界の移動を通知する。

【表４】

表4

seq parameter set rbsp( ) {	記述子
profile idc	u(8)
reserved zero 8bits /* equal to 0 */	u(8)
level idc	u(8)
seq parameter set id	ue(v)
pic width in luma samples	u(16)
pic height in luma samples	u(16)
bit depth luma minus8	ue(v)
bit depth chroma minus8	ue(v)
bit depth luma increment	ue(v)
bit depth chroma increment	ue(v)
log2 max frame num minus4	ue(v)
pic order cnt type	ue(v)
if(pic order cnt type == 0)	
log2 max pic order cnt lsb minus4	ue(v)
else if(pic order cnt type == 1) {	
delta pic order always zero flag	u(1)
offset for non ref pic	se(v)
num ref frames in pic order cnt cycle	ue(v)
for(i = 0; i<num ref frames in pic order cnt cycle; i++)	
offset for ref frame[i]	se(v)
}	
max num ref frames	ue(v)
gaps in frame num value allowed flag	u(1)
log2 min coding block size minus3	ue(v)
log2 diff max min coding block size	ue(v)
log2 min transform block size minus2	ue(v)
log2 diff max min transform block size	ue(v)
max transform hierarchy depth inter	ue(v)
max transform hierarchy depth intra	ue(v)
interpolation filter flag	u(1)
num tile columns minus1	ue(v)
num tile rows minus1	ue(v)
if(num tile columns minus1 != 0 || num tile rows minus1 != 0) {	
tile boundary independence idc	u(1)
if(num tile columns minus1 + num tile rows minus1 == 1) // one is equal to	
dynamic tile boundary flag	u(1)
if(dynamic tile boundary flag) {	
tile boundary change direction flag	u(1)
tile boundary change rate minus1	ue(v)
}	
else {	
uniform spacing idc	u(1)
if(uniform spacing idc != 1) {	
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)
}	
}	
}	
rbsp trailing bits( )	
}	

【表５】

表5

slice_header( ) {	記述子
lightweight_slice_flag	u(1)
if(!lightweight_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	u(v)
if(IdrPicFlag)	
idr_pic_id	ue(v)
if(pic_order_cnt_type == 0)	
pic_order_cnt_lsb /*	u(v)
if(slice_type == P || slice_type == B) {	
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)
}	
}	
ref_pic_list_modification( )	
ref_pic_list_combination( )	
if(nal_ref_idc != 0)	
dec_ref_pic_marking( )	
}	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
first_slice_in_pic_flag	u(1)
if(first_slice_in_pic_flag == 0)	
slice_address	u(v)
if(!lightweight_slice_flag) {	
slice_qp_delta	se(v)
if(sample_adaptive_offset_enabled_flag)	
sao_param()	
if(deblocking_filter_control_present_flag) {	
disable_deblocking_filter_idc	
if(disable_deblocking_filter_idc != 1) {	
slice_alpha_c0_offset_div2	
slice_beta_offset_div2	
}	
}	
if(slice_type = = B)	
collocated_from_l0_flag	u(1)
if(adaptive_loop_filter_enabled_flag) {	
if(!shared_pps_info_enabled_flag)	
alf_param( )	
alf_cu_control_param( )	
}	
if(dynamic_tile_boundary_flag)	
tile_boundary_change_cycle	u(v)
}	
}	

【０１６６】

表４において、「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素は、タイル境界が動的に変化するか否かを示す。例えば、「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素が１の場合、タイル境界は動的に変化し得る。「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」が０の場合、タイル境界は動的に変化しない。「ｎｕｍ＿ｔｉｌｅ＿ｃｏｌｕｍｎｓ＿ｍｉｎｕｓ１」構文要素と「ｎｕｍ＿ｔｉｌｅ＿ｒｏｗｓ＿ｍｉｎｕｓ１」構文要素との和が１でない場合、「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素は１にはなり得ない。「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素が存在しない場合、ビデオデコーダ３０は「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素が０であると推論し得る。１つの垂直方向のタイルまたは２つの水平方向のタイルしかない場合にのみ、「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素は１となり得る。

【０１６７】

さらに、表４の「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素により、タイル境界の移動の方向を指定することができる。例えば、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が０で、タイル境界が垂直の場合、タイル境界は、左から右に１つのピクチャから次のピクチャへと移動し得る。「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が１で、タイル境界が水平の場合、タイル境界は右から左に１つのピクチャから次のピクチャへと移動し得る。「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が０で、タイル境界が水平の場合、タイル境界は上から下に移動し得る。「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が１で、タイル境界が水平の場合、タイル境界は下から上に移動し得る。

【０１６８】

表４の「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素は、タイル境界が１つのピクチャから次のピクチャに移動するツリーブロックの行または列の数を指定する。例えば、タイル境界が垂直の場合、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素は、タイル境界が１つのピクチャから次のピクチャに移動するツリーブロックの列の数を示し得る。タイル境界が水平の場合、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素は、タイル境界が１つのピクチャから次のピクチャに移動するツリーブロックの行の数を示し得る。行数または列数は、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素に１を加えた数とすることができる。例えば、変数「ＴｉｌｅＢｏｕｎｄａｒｙＣｈａｎｇｅＲａｔｅ」の値は、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素に１を加えることによって導き出すことができる。説明の便宜上、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素をレート構文要素と呼ぶことがある。

【０１６９】

ビデオデコーダ３０は、表５の「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」構文要素を使用してタイル境界の位置を導き出すことができる。説明の便宜上、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」構文要素を周期構文要素と呼ぶことがある。周期構文要素は、シーケンス内のピクチャに関連付けられた周期を示し得る。タイル境界が垂直の場合、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」構文要素は以下の式で示されるビット数によって表すことができる。

【数１】

$$\mathrm{Ceil}(\mathrm{Log2}(\mathrm{PicWidthInLcus} \div \mathrm{TileBoundaryChangeRate} + 1))$$

【０１７０】

上式で、Ｃｅｉｌ（．．．）は上限関数、Ｌｏｇ２（．．．）は２を底とした対数関数、ＰｉｃＷｉｄｔｈＩｎＬｃｕｓ」はＬＣＵ（すなわちツリーブロック）内のピクチャの幅である。さらに、タイル境界が垂直の場合、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」構文要素は、０から以下の式で示される値までの範囲の値をとり得る。

【数２】

$$Ceil(PicWidthInLcus \div TileBoundaryChangeRate)$$

【０１７１】

タイル境界が水平の場合、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」要素は、以下の式によって示されるビット数で表すことができる。

【数３】

$$Ceil(Log2(PicHeightInLcus \div TileBoundaryChangeRate + 1))$$

【０１７２】

上式で、「ＰｉｃｈｅｉｇｈｔＩｎＬｃｕｓ」はＬＣＵ（すなわちツリーブロック）内のピクチャの高さである。さらに、タイル境界が水平の場合、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」構文要素は０から以下の式で示される値までの範囲の値をとり得る。

【数４】

$$Ceil(PicHeightInLcus \div TileBoundaryChangeRate)$$

【０１７３】

タイル境界が垂直の場合（すなわち「ｎｕｍ＿ｔｉｌｅ＿ｃｏｌｕｍｎｓ＿ｍｉｎｕｓ１」構文要素が１の場合）、ビデオデコーダ３０は、タイル境界の位置を以下のようにして導き出すことができる。「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が０の場合、左のタイルのツリーブロックにおける幅は「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」＊「ＴｉｌｅＢｏｕｎｄａｒｙＣｈａｎｇｅＲａｔｅ」に等しい。「ＴｉｌｅＢｏｕｎｄａｒｙＣｈａｎｇｅＲａｔｅ」変数は「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素に１を加えた値に等しい。一方、「ｔｉｌｅ＿ｂｏｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が１の場合、右のタイルのツリーブロックにおける幅は「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」＊「ＴｉｌｅＢｏｕｎｄａｒｙＣｈａｇｅＲａｔｅ」に等しい。

【０１７４】

タイル境界が水平の場合（すなわち「ｎｕｍ＿ｔｉｌｅ＿ｒｏｗｓ＿ｍｉｎｕｓ１」構文要素が１の場合）、ビデオデコーダ３０は、タイル境界の位置を以下のようにして導き出すことができる。「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が０の場合、上のタイルのツリーブロックにおける高さは「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」＊「ＴｉｌｅＢｏｕｎｄａｒｙＣｈａｎｇｅＲａｔｅ」に等しい。「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素が１の場合、下のタイルのツリーブロックにおける高さは「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｃｙｃｌｅ」＊「ＴｉｌｅＢｏｕｎｄａｒｙＣｈａｎｇｅＲａｔｅ」に等しい。

【０１７５】

表５で、変数「ＩｄｒＰｉｃＦｌａｇ」は以下のように指定される。

【数５】

$$IdrPicFlag = ((nal_unit_type == 5 \ \| \ nal_unit_type == 6) \ ? \ 1 \ : \ 0)$$

【０１７６】

言い換えると、スライスヘッダーを含むＮＡＬユニットの「ｎａｌ＿ｕｎｉｔ＿ｔｙｐｅ」構文要素の値が５または６の場合、「ＩｄｒＰｉｃＦｌａｇ」変数の値は１である。

【０１７７】

他の例において、表６にピクチャパラメータセットＲＢＳＰの例示の構文規則を示す。ＰＰＳ　ＲＢＳＰの構文要素と表５のスライスヘッダーとが組み合わさって、タイル境界の移動と位置とを通知することができる。

【表６】

表6

pic_parameter_set_rbsp( ) {	記述子
pic_parameter_set_id	ue(v)
seq_parameter_set_id	ue(v)
entropy_coding_mode_flag	u(1)
num_ref_idx_l0_default_active_minus1	ue(v)
num_ref_idx_l1_default_active_minus1	ue(v)
pic_init_qp_minus26/* relative to 26 ***/**	se(v)
constrained_intra_pred_flag	u(1)
tile_info_present_flag	u(1)
if(tile_info_present_flag = = 1) {	
num_tile_columns_minus1	ue(v)
num_tile_rows_minus1	ue(v)
if(num_tile_columns_minus1 != 0 || num_tile_rows_minus1 != 0) {	
tile_boundary_independence_idc	u(1)
if(num_tile_columns_minus1 + num_tile_rows_minus1 = = 1) // one is	
dynamic_tile_boundary_flag	u(1)
if(dynamic_tile_boundary_flag) {	
tile_boundary_change_direction_flag	u(1)
tile_boundary_change_rate_minus1	ue(v)
}	
else {	
uniform_spacing_idc	u(1)
if(uniform_spacing_idc != 1) {	
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)
}	
}	
}	
}	
rbsp_trailing_bits( )	
}	

【０１７８】

表６の「ｄｙｎａｍｉｃ＿ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｆｌａｇ」構文要素と、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｄｉｒｅｃｔｉｏｎ＿ｆｌａｇ」構文要素と、「ｔｉｌｅ＿ｂｏｕｎｄａｒｙ＿ｃｈａｎｇｅ＿ｒａｔｅ＿ｍｉｎｕｓ１」構文要素とは、表４の対応する要素と同じ意味を有し得る。表４、表５、および表６のその他の構文要素は、前掲のＪＣＴＶＣ文書ＪＣＴＶＣ－Ｆ３３５またはＨＥＶＣ　ＷＤ４で定義されている意味を有する。ある例では、ビデオエンコーダ２０とビデオデコーダ３０とは、表４の構文規則に加えて、またはその代わりに、表６の構文規則を使用してもよい。

【０１７９】

このようにして、ビデオエンコーダ２０はビデオデータを符号化する方法を実行することができる。ビデオデータの符号化方法は、シーケンス内の各ピクチャをタイル境界によって分けられた１対のタイルに区画化することを備え得る。この符号化方法は、シーケンス内のピクチャからピクチャへのタイル境界の移動を示す１つまたは複数の構文要素を生成することも備え得る。同様に、ビデオデコーダ３０は、ビデオデータを復号する方法を実行することができる。この復号方法は、ピクチャのシーケンス内のピクチャからピクチャへのタイル境界の移動を示す１つまたは複数の構文要素を、ビデオデータのビットストリームで受信することを備え得る。この復号方法は、ピクチャのそれぞれについて、構文要素に基づいてタイル境界の位置を判断することも備え得る。

【０１８０】

図３は、本開示の技術を実装するように構成された例示のビデオデコーダ３０を示すブロック図である。図３は、説明のために示すものであって、本技術を本開示で大まかに例示し、説明する通りに限定するものではない。説明のために、本開示ではビデオデコーダ３０についてＨＥＶＣコーディングの文脈で説明する。しかし、本開示の技術は他のコーディング標準または方法にも適用可能である。

【０１８１】

図３の例では、ビデオデコーダ３０は複数の機能構成要素を含む。ビデオデコーダ３０の機能構成要素は、エントロピー復号ユニット１５０と、予測処理ユニット１５２と、逆量子化ユニット１５４と、逆変換処理ユニット１５６と、再構築ユニット１５８と、フィルターユニット１５９と、復号ピクチャバッファ１６０とを含む。予測処理ユニット１５２は、動き補償ユニット１６２とイントラ予測処理ユニット１６４とを含む。他の例では、ビデオデコーダ３０の機能構成要素の数はこれより多くても少なくてもよく、異なる機能構成要素を含んでもよい。

【０１８２】

ビデオデコーダ３０は、複数の構文要素を備えるビットストリームを受信することができる。エントロピー復号ユニット１５０は、ビットストリームを構文解析してビットストリームから構文要素を抽出することができる。ビットストリームの構文解析の一部として、エントロピー復号ユニット１５０はビットストリーム内のエントロピー符号化された構文要素をエントロピー復号することができる。予測処理ユニット１５２と、逆量子化ユニット１５４と、逆変換処理ユニット１５６と、再構築ユニット１５８と、フィルターユニット１５９とは、ビットストリームから抽出された構文要素に基づいて復号ビデオデータを生成することができる。

【０１８３】

ビットストリームは、一連のＮＡＬユニットを備え得る。ビットストリームのＮＡＬユニットは、ＳＰＳ　ＮＡＬユニット、ＰＰＳ　ＮＡＬユニット、ＳＥＩ　ＮＡＬユニットなどを含み得る。ビットストリームの構文解析の一部として、エントロピー復号ユニット１５０は、ＳＰＳ　Ｎａｌユニットから ＳＰＳ、ＰＰＳ　ＮＡＬユニットからＰＰＳ、ＳＥＩユニットからＳＥＩメッセージ、というように抽出して復号することができる。

【０１８４】

さらに、ビットストリームのＮＡＬユニットは、コード化スライスＮＡＬユニットを含み得る。ビットストリームの構文解析の一部として、エントロピー復号ユニット１５０はコード化スライスＮＡＬユニットからコード化スライスを抽出してエントロピー復号する

ことができる。コード化スライスのそれぞれはスライスヘッダーとスライスデータとを含み得る。スライスヘッダーはスライスに関する構文要素を含み得る。スライスヘッダー内の構文要素は、スライスを含むピクチャに関連付けられたＰＰＳを識別する構文要素を含み得る。エントロピー復号ユニット１５０は、コード化スライスヘッダー内の構文要素についてＣＡＢＡＣ復号演算などのエントロピー復号演算を行ってスライスヘッダーを復元することができる。

【０１８５】

コード化スライスＮＡＬユニットからのスライスデータの抽出の一部として、エントロピー復号ユニット１５０はスライスデータ内のコード化ＣＵから構文要素を抽出することができる。抽出される構文要素は、係数ブロックに関連付けられた構文要素を含み得る。エントロピー復号ユニット１５０は、これらの構文要素の一部に対してＣＡＢＡＣ復号演算を行ってもよい。

【０１８６】

構文要素に対するＣＡＢＡＣ復号の実行の一部として、エントロピー復号ユニット１５０はコーディングコンテキストを識別することができる。エントロピー復号ユニット１５０は、１つの区間を上部部分区間と下部部分区間とに分割することができる。部分区間の一方には値０が関連付けられてもよく、他方の部分区間には値１が関連付けられてもよい。部分区間の幅は、関連付けられた値について識別されたコーディングコンテキストによって示される確率に比例し得る。符号化値が下部部分区間内にある場合、エントロピー復号ユニット１５０は下部部分区間に関連付けられた値を有するビンを復号することができる。符号化値が上部部分区間内にある場合、エントロピー復号ユニット１５０は上部部分区間に関連付けられた値を有するビンを復号することができる。構文要素の次のビンを復号するために、エントロピー復号ユニット１５０はこれらのステップを、符号化値を含む部分区間である間隔で繰り返すことができる。エントロピー復号ユニット１５０が次のビンについてこれらのステップを繰り返すとき、エントロピー復号ユニット１５０は、識別されたコーディングコンテキストと復号ビンとによって示される確率に基づいて修正された確率を使用してもよい。エントロピー復号ユニット１５０はこれらのビンを逆２値化して構文要素を復元してもよい。

【０１８７】

さらに、ビデオデコーダ３０は非区画化ＣＵに対して再構築演算を行うことができる。非区画化ＣＵに対して再構築演算を行うために、ビデオデコーダ３０はＣＵの各ＴＵに対して再構築演算を行うことができる。ＣＵの各ＴＵについて再構築演算を行うことによって、ビデオデコーダ３０はＣＵに関連付けられた残差画素ブロックを再構築することができる。

【０１８８】

ＣＵのＴＵに対する再構築操作の実行の一部として、逆量子化ユニット１５４がＴＵに関連付けられた係数ブロックを逆量子化、すなわち逆２値化することができる。逆量子化ユニット１５４は、ＴＵのＣＵに関連付けられたＱＰ値を使用して量子化の程度を判断することができ、同様に、逆量子化ユニット１５４が適用する逆量子化の程度を判断することができる。

【０１８９】

逆量子化ユニット１５４が係数ブロックを逆量子化した後、逆変換処理ユニット１５６が、ＴＵに関連付けられた残差サンプルブロックを生成するために係数ブロックに１つまたは複数の逆変換を適用することができる。例えば、逆変換処理ユニット１５６は、係数ブロックに対して、逆ＤＣＴ、逆整数変換、逆カルーネンレーベ変換（ＫＬＴ）、逆回転変換、逆方向性変換、またはその他の逆変換を適用してもよい。

【０１９０】

ＰＵがスキップモードで符号化されている場合、またはＰＵの動き情報がマージモードを使用して符号化されている場合、動き補償ユニット１６２はそのＰＵのマージ候補リストを生成してもよい。動き補償ユニットはマージ候補リスト内の選択されたマージ候補を

識別することができる。動き補償ユニット１６２は、選択されたマージ候補によって示される動き情報に関連付けられた１つまたは複数の参照ブロックに基づいてＰＵのための予測サンプルブロックを生成することができる。

【０１９１】

ＰＵの動き情報がＡＭＶＰモードを使用して符号化されている場合、動き補償ユニット１６２はリスト０ＭＶ予測子候補リストおよび／またはリスト１ＭＶ予測子候補リストを生成してもよい。動き補償ユニット１６２は、選択されたリスト０ＭＶ予測子候補および／または選択されたリスト１ＭＶ予測子候補を判断することができる。次に、動き補償ユニット１６２は、リスト０動きベクトル差（ＭＶＤ）、リスト１ＭＶＤ、選択されたリスト０ＭＶ予測子候補によって指定されるリスト０動きベクトル、および／または選択されたリスト１ＭＶ予測子候補によって指定されるリスト１動きベクトルに基づいて、ＰＵのためのリスト１動きベクトル、および／または、ＰＵのためのリスト０動きベクトルを判断することができる。動き補償ユニット１６２は、リスト０動きベクトルとリスト０参照ピクチャインデックスおよび／またはリスト１動きベクトルとリスト１参照ピクチャインデックスに関連付けられた参照ブロックに基づいて、ＰＵの予測サンプルブロックを生成することができる。

【０１９２】

ある例では、動き補償ユニット１６２は補間フィルターに基づいて補間を行うことによってＰＵの予測サンプルブロックを精緻化してもよい。サブピクセル精度での動き補償に使用する補間フィルターの識別子を構文要素に含め得る。動き補償ユニット１６２は、参照ブロックのサブ整数サンプルの補間値を計算するためにＰＵの予測サンプルブロックの生成時にビデオエンコーダ２０が使用したのと同じ補間フィルターを使用することができる。動き補償ユニット１６２は、ビデオエンコーダ２０が使用した補間フィルターを受信構文情報に従って判断し、その補間フィルターを使用して予測サンプルブロックを生成することができる。

【０１９３】

ＰＵがイントラ予測を使用して符号化されている場合、イントラ予測処理ユニット１６４はイントラ予測を行ってＰＵの予測サンプルブロックを生成することができる。例えば、イントラ予測処理ユニット１６４は、ビットストリーム内の構文要素に基づいてＰＵのイントラ予測モードを判断することができる。イントラ予測処理ユニット１６４は、イントラ予測モードを使用して、空間的に隣接するＰＵのサンプルブロックに基づいてＰＵの予測データ（例えば予測サンプル）を生成してもよい。

【０１９４】

再構築ユニット１５８は、ＣＵのＴＵに関連付けられた残差サンプルブロックと、ＣＵのＰＵの予測サンプルブロック、すなわち適宜にイントラ予測データまたはインター予測データ、とを使用して、ＣＵのサンプルブロックを再構築することができる。具体的には、再構築ユニット１５８は残差サンプルブロックのサンプルを、予測サンプルブロックの対応するサンプルに加えてＣＵのサンプルブロックを再構築することができる。

【０１９５】

フィルターユニット１５９は、デブロッキング操作を行ってＣＵに付随するブロッキングアーチファクトを低減することができる。ビデオデコーダ３０はＣＵのサンプルブロックを復号ピクチャバッファ１６０に格納してもよい。復号ピクチャバッファ１６０は、後で行われる動き補償と、イントラ予測と、図１の表示デバイス３２などの表示デバイス上への表示とのために参照ピクチャを提供することができる。例えば、ビデオデコーダ３０は、復号ピクチャバッファ１６０内のサンプルブロックに基づいて、他のＣＵのＰＵについてイントラ予測またはインター予測を行うことができる。

【０１９６】

図４は、本開示の１つまたは複数の技術による、ビデオエンコーダ２０の例示の動作２００を示すフローチャートである。図４の例では、ビデオエンコーダ２０はビデオデータのスライスのピクチャ区画（例えばタイル、波、エントロピースライスなど）を符号化す

ることができる（２０２）。スライスは複数のコーディングユニットを含み得る。ビデオエンコーダ２０はピクチャ区画を符号化することにより符号化ピクチャ区画を生成する。ビデオエンコーダ２０は、スライスの符号化ピクチャ区画を含むコード化スライスＮＡＬユニットを生成することができる（２０４）。

【０１９７】

さらに、ビデオエンコーダ２０はコード化ピクチャ区画の入口点を示すＳＥＩメッセージを生成することができる（２０６）。すなわち、ＳＥＩメッセージはコード化ピクチャ区画のコード化スライスＮＡＬユニット内での位置を示すことができる。ＳＥＩメッセージは前述の形式を有し得る。例えば、ビデオエンコーダ２０はＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを生成してもよい。

【０１９８】

ビデオエンコーダ２０は、コード化スライスＮＡＬユニットとＳＥＩメッセージとを含むビットストリームを出力することができる（２０８）。このビットストリームは、ビデオデータの符号化表現を含む。ビデオエンコーダ２０は様々な方法でビットストリームを出力することができる。例えば、ビデオエンコーダ２０は、ビットストリームをインターネットなどの通信ネットワークで送信してもよい。他の例では、ビデオエンコーダ２０はビットストリームをハードディスクドライブまたはメモリなどのコンピュータ可読記憶媒体に出力してもよい。

【０１９９】

図５は、本開示の１つまたは複数の技術によるビデオデコーダ３０の例示の動作２５０を示すフローチャートである。図５の例では、ビデオデコーダ３０は複数のコード化ピクチャ区画を含むコード化スライスＮＡＬユニットを受信することができる（２５２）。さらに、ビデオデコーダ３０は、コード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージを、例えば非ＶＣＬ　ＮＡＬユニットで受信することができる（２５４）。ビデオデコーダ３０は様々な方法でコード化スライスＮＡＬユニットとＳＥＩメッセージとを受信することができる。例えば、ビデオデコーダ３０は、インターネットなどの通信ネットワークを介してリモートサーバからコード化スライスＮＡＬユニットとＳＥＩメッセージとを受信してもよい。他の例では、ビデオデコーダ３０は、ハードディスクドライブまたはメモリなどのコンピュータ可読記憶媒体からコード化スライスＮＡＬユニットとＳＥＩメッセージとを受信してもよい。

【０２００】

ビデオデコーダ３０は、ＳＥＩメッセージで示される入口点に少なくとも部分的に基づいて、コード化スライスＮＡＬユニット内での符号化ピクチャ区画の位置を判断することができる（２５６）。例えば、ＳＥＩメッセージは、特定の符号化ピクチャ区画がコード化スライスＮＡＬユニットのスライスデータの２５６番目のバイトから始まることを示し得る。ビデオデコーダ３０は符号化ピクチャ区画を復号することができる（２５８）。ある例では、ビデオデコーダ３０は、符号化ピクチャ区画の２つ以上の区画を並列して復号することができる。

【０２０１】

ビデオデコーダ３０は符号化ピクチャ区画の復号画素ブロックを出力することができる（２６０）。ある例では、ビデオデコーダ３０は復号画素ブロックを表示デバイス３２に表示するために出力する。他の例ではビデオデコーダ３０は、復号画素ブロックをビデオエンコーダに出力してもよい。この例では、ビデオエンコーダはその復号画素ブロックをビデオトランスコーディング動作の一部として再符号化してもよい。

【０２０２】

図６は、本開示の１つまたは複数の技術による、バイト整列スライスヘッダーを有するコード化スライスＮＡＬユニットを生成するためのビデオエンコーダ２０の例示の動作２８０を示すフローチャートである。図６の例では、ビデオエンコーダ２０は複数の符号化ピクチャ区画を生成することができる（２８２）。符号化ピクチャ区画のそれぞれはビデオデータ中のピクチャ内の異なる１組のＣＵに関連付けられている。さらに、ビデオエン

コーダ２０はスライスヘッダーおよび複数の符号化ピクチャ区画を含むＮＡＬユニットを生成することができる（２８４）。スライスヘッダーは、スライスヘッダーをバイト境界に整列させる１つまたは複数のパディングビットを含み得る。

【０２０３】

図７は、本開示の１つまたは複数の技術による、バイト整列スライスヘッダーを生成するためのビデオエンコーダ２０の例示の動作３００を示すフローチャートである。図７の例では、ビデオエンコーダ２０はスライスヘッダーを生成することができる（３０２）。さらに、ビデオエンコーダ２０は、パラメータセット（例えばシーケンスパラメータセット、ピクチャパラメータセット、アダプテーションパラメータセットなど）内の構文要素が、スライスヘッダーがバイト境界に整列しなければならないことを示しているか否かを判断する（３０４）。

【０２０４】

構文要素が、スライスヘッダーがバイト境界に整列しなければならないことを示している場合（３０４の「ＹＥＳ」）、ビデオエンコーダ２０はスライスヘッダーがバイト境界に整列しているか否かを判断してもよい（３０６）。スライスヘッダーがバイト境界に整列していない場合（３０６の「ＮＯ」）、ビデオエンコーダ２０はスライスヘッダーの後ろにビットを追加してもよい（３０８）。スライスヘッダーの後ろにビットを追加した後、ビデオエンコーダ２０はスライスヘッダーがバイト境界に整列しているか否かを再度判断することができる（３０６）。このようにして、ビデオエンコーダ２０はスライスヘッダーがバイト整列されるまでスライスヘッダーの後ろにビットを追加し続けることができる。

【０２０５】

スライスヘッダーがバイト境界に整列されている場合（３０６の「ＹＥＳ」）または構文要素が、スライスヘッダーがバイト境界に整列する必要がないことを示している場合（３０４の「ＮＯ」）、動作３００は完了する。

【０２０６】

図８は、本開示の１つまたは複数の技術による、ビデオデコーダ３０がバイト整列スライスヘッダーを有するコード化スライスＮＡＬユニットを受信する、ビデオデコーダ３０の例示の動作３３０を示すフローチャートである。図８の例では、ビデオデコーダ３０はＮＡＬユニットを受信することができる（３３２）。ＮＡＬユニットは、スライスヘッダーとビデオデータの複数の符号化ピクチャ区画とを含み得る。スライスヘッダーは、スライスヘッダーをバイト境界に整列させる１つまたは複数のパディングビットを含み得る。ビデオデコーダ３０は符号化ピクチャ区画の２つ以上を並列して復号してもよい（３３４）。

【０２０７】

図９は、波面並列処理を示す概念図である。前述のように、ピクチャは、各画素ブロックがツリーブロックに関連付けられた画素ブロックに区画化され得る。図９では、ツリーブロックに関連付けられた画素ブロックを白い正方形からなる格子として示している。ピクチャはツリーブロック行３５０Ａ～３５０Ｅ（「ツリーブロック行３５０」と総称）を含む。

【０２０８】

第１のスレッドは、ツリーブロック行３５０Ａ内のツリーブロックをコーディング中であってよい。同時に、他のスレッドはツリーブロック行３５０Ｂと、３５０Ｃと、３５０Ｄとのツリーブロックをコーディング中とすることができる。図９の例では、第１のスレッドは現在ツリーブロック３５２Ａをコーディング中であり、第２のスレッドが現在ツリーブロック３５２Ｂをコーディング中であり、第３のスレッドが現在ツリーブロック３５２Ｃをコーディング中であり、第４のスレッドが現在ツリーブロック３５２Ｄをコーディング中である。本開示では、ツリーブロック３５２Ａと、３５２Ｂと、３５２Ｃと、３５２Ｄとを「現在のツリーブロック３５２」と総称することがある。ビデオコーダはすぐ上の行の３つ以上のツリーブロックがコーディングされた後でツリーブロック行のコーディ

ングを開始することができるため、現在のツリーブロック３５２は水平方向にツリーブロック２つの幅分だけ互いにずれている。

【０２０９】

図９の例では、スレッドは太い灰色の矢印で示すツリーブロックからのデータを使用して現在のツリーブロック３５２内のＣＵのためのイントラ予測またはインター予測を行うことができる。スレッドは、ＣＵのインター予測を行うために１つまたは複数の参照フレームも使用することができる。所与のツリーブロックをコーディングするために、スレッドは前にコーディングされたツリーブロックに関連付けられた情報に基づいて１つまたは複数のＣＡＢＡＣコンテキストを選択することができる。スレッドは、その１つまたは複数のＣＡＢＡＣコンテキストを使用して、その所与のツリーブロックの先頭ＣＵに関連付けられた構文要素に対してＣＡＢＡＣコーディングを行うことができる。その所与のツリーブロックが行の左端のツリーブロックでない場合、スレッドはその所与のツリーブロックの左のツリーブロックの最後のＣＵに関連付けられた情報に基づいて１つまたは複数のＣＡＢＡＣコンテキストを選択してもよい。その所与のツリーブロックが行の左端のツリーブロックである場合、スレッドはその所与のツリーブロックの上方で２ツリーブロック右側にあるツリーブロックの最後のＣＵに関連付けられた情報に基づいて、１つまたは複数のＣＡＢＡＣコンテキストを選択することができる。スレッドは、細い黒の矢印で示されているツリーブロックの最後のＣＵからのデータを使用して、現在のツリーブロック３５２の先頭ＣＵのためのＣＡＢＡＣコンテキストを選択することができる。

【０２１０】

図１０は、複数のタイル４０２Ａと、４０２Ｂと、４０２Ｃと、４０２Ｄと、４０２Ｅと、４０２Ｆ（「タイル４０２」と総称）とに区画化されたピクチャ４００の例示のツリーブロックコーディング順序を示す概念図である。ピクチャ４００内の各正方形ブロックはツリーブロックに関連付けられた画素ブロックを表す。太い破線は例示のタイル境界を示す。異なる種類の網掛けは異なるスライスに対応する。

【０２１１】

画素ブロック内の数字は、ピクチャ４００のタイルコーディング順でのタイル内の対応するツリーブロック（ＬＣＵ）の位置を示す。図１０の例に示すように、タイル４０２Ａ内のツリーブロックが最初にコーディングされ、次にタイル４０２Ｂ内のツリーブロック、その次にタイル４０２Ｃ内のツリーブロック、その次にタイル４０２Ｄ内のツリーブロック、その次にタイル４０２Ｅ内のツリーブロック、その次にタイル４０２Ｆ内のツリーブロックがそれぞれコーディングされる。各タイル４０２内では、ツリーブロックはラスタースキャン順にコーディングされる。

【０２１２】

ビデオエンコーダは、ピクチャ４００のために４つのコード化スライスＮＡＬユニットを生成することができる。第１のコード化スライスＮＡＬユニットはツリーブロック１～１８の符号化表現を含み得る。第１のコード化スライスＮＡＬユニットのスライスデータは、２つの符号化ピクチャ区画を含み得る。最初の符号化ピクチャ区画は、ツリーブロック１～９の符号化表現を含み得る。２番目の符号化ピクチャ区画は、ツリーブロック１０～１８の符号化表現を含み得る。したがって、最初のコード化スライスＮＡＬユニットは、複数のタイルを含むスライスの符号化表現を含み得る。

【０２１３】

第２のコード化スライスＮＡＬユニットは、ツリーブロック１９～２２の符号化表現を含み得る。第２のコード化スライスＮＡＬユニットのスライスデータは、単一の符号化ピクチャ区画を含み得る。第３のコード化スライスＮＡＬユニットは、ツリーブロック２３～２７の符号化表現を含み得る。第３のコード化スライスＮＡＬユニットのスライスデータは、１つの符号化ピクチャ区画のみを含み得る。したがって、タイル４０２Ｃは複数のスライスを含み得る。第４のコード化スライスＮＡＬユニットは、ツリーブロック２８～４５の符号化表現を含み得る。第４のコード化スライスＮＡＬユニットのスライスデータは、タイル４０２Ｄと、４０２Ｅと、４０２Ｆとのそれぞれに１つずつ、３つの符号化ピ

クチャ区画を含み得る。したがって、第４のコード化スライスＮＡＬユニットは、複数のタイルを含むスライスの符号化表現を含み得る。

【０２１４】

図１１は、本開示の１つまたは複数の態様による、ＮＡＬユニット内の符号化ピクチャ区画のオフセットに関連付けられた構文要素を表す可変長値を含むビットストリームを生成するための、ビデオエンコーダ２０の例示の動作４３０を示すフローチャートである。図１１の例では、ビデオエンコーダ２０は、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とをエントロピー符号化することができる（４３２）。オフセット構文要素のそれぞれの長さは、第１の構文要素に基づいて判断し得る。一連のオフセット構文要素内のオフセット構文要素の数は、第２の構文要素に基づいて判断し得る。ＮＡＬユニット内の複数の符号化ピクチャ区画の位置は、オフセット構文要素に基づいて判断し得る。ビデオエンコーダ２０は、エントロピー符号化された第１の構文要素を表す可変長値と、エントロピー符号化された第２の構文要素を表す可変長値と、オフセット構文要素を表す固定長値とを含むビットストリームを生成することができる（４３４）。

【０２１５】

図１２は、本開示の１つまたは複数の態様による、ＮＡＬユニット内の符号化ピクチャ区画のオフセットに関連付けられた構文要素を表す可変長値を含むビットストリームによって表されたビデオデータを復号するための、ビデオデコーダの例示の動作４５０を示すフローチャートである。図１２の例では、ビデオデコーダ３０は、第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連のオフセット構文要素とを含むビットストリームを受信する（４５２）。さらに、ビデオデコーダ３０は、第１の構文要素と第２の構文要素とをエントロピー復号することができる（４５４）。ビデオデコーダ３０は、第１の構文要素に基づいてオフセット構文要素のそれぞれの長さを判断することができる（４５６）。ビデオデコーダ３０は、第２の構文要素に基づいて、一連のオフセット構文要素内のオフセット構文要素の数を判断することができる（４５８）。

【０２１６】

さらに、ビデオデコーダ３０は、オフセット構文要素のそれぞれの長さと一連のオフセット構文要素内のオフセット構文要素の数とに少なくとも部分的に基づいて、オフセット構文要素を構文解析することができる（４６０）。ある例では、ビデオデコーダ３０がオフセット構文要素を構文解析するときに、ビデオデコーダ３０はオフセット構文要素のそれぞれについて固定長復号を実行する。固定長復号は、エントロピー復号の特殊モードとも呼ばれ得る。同様に、固定長符号化はエントロピー符号化の特殊モードとも呼ばれ得る。

【０２１７】

ビデオデコーダ３０は、オフセット構文要素に基づいてＮＡＬユニット内の複数の符号化ピクチャ区画の位置を判断することができる（４６２）。ある例では、ＮＡＬユニットのスライスヘッダーが、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とを含む。他の例では、ＳＥＩメッセージが、第１の構文要素と、第２の構文要素と、一連のオフセット構文要素とを含む。さらに、ビデオデコーダ３０は符号化ピクチャ区画を復号することができる（４６４）。

【０２１８】

１つまたは複数の例では、叙述の各機能はハードウェア、ソフトウェア、ファームウェア、またはこれらの任意の組み合わせで実装され得る。ソフトウェアで実装する場合、これらの機能は１つまたは複数の命令またはコードとしてコンピュータ可読媒体に記憶され、または送信され、ハードウェアベースの処理ユニットによって実行され得る。コンピュータ可読媒体は、データ記憶媒体などの有形の媒体に対応するコンピュータ可読記憶媒体、または、例えば通信プロトコルに従って１つの場所から他の場所へのコンピュータプログラムの転送を容易にする媒体を含む通信媒体を含み得る。このように、コンピュータ可読媒体は、一般に、（１）非一時的な有形のコンピュータ可読記憶媒体、または（２）信

号または搬送波などの通信媒体に対応し得る。データ記憶媒体は、１つまたは複数のコンピュータまたは１つまたは複数のプロセッサが、本開示に記載の技術の実装のために命令、コードおよび／またはデータ構造体を読み出すためにアクセスすることができる任意の入手可能な媒体とすることができる。コンピュータプログラム製品は、コンピュータ可読媒体を含み得る。

【０２１９】

一例として、このようなコンピュータ可読記憶媒体はＲＡＭ、ＲＯＭ、ＥＥＰＲＯＭ（登録商標）、ＣＤ－ＲＯＭまたはその他の光ディスクストレージ、磁気ディスクストレージ、またはその他の磁気ストレージデバイス、フラッシュメモリ、または所望のプログラムコードを命令またはデータ構造体の形で格納するために使用可能かつコンピュータによりアクセス可能なその他のあらゆる媒体を含み得るが、これらには限定されない。また、いかなる接続もコンピュータ可読媒体と正式に称することができる。例えば、ウェブサイト、サーバ、またはその他のリモートソースから同軸ケーブル、光ファイバーケーブル、ツイストペア、デジタル加入者回線（ＤＳＬ）、または赤外線、無線およびマイクロ波などのワイヤレス技術を使用して命令を送信する場合、その同軸ケーブル、光ファイバーケーブル、ツイストペア、ＤＳＬ、または赤外線、無線およびマイクロ波などのワイヤレス技術は媒体の定義に含まれる。ただし、コンピュータ可読媒体およびデータ記憶媒体は、接続、搬送波、信号またはその他の一時的媒体を含まず、非一時的な有形の記憶媒体のことを意図しているものと理解すべきである。本開示では、ディスク（ｄｉｓｋおよびｄｉｓｃ）は、コンパクトディスク（ＣＤ）、レーザディスク、デジタル多用途ディスク（ＤＶＤ）、フロッピー（登録商標）ディスク、およびＢｌｕ－ｒａｙ（登録商標）ディスクを含み、この場合、ｄｉｓｋは通常データを磁気的に再生し、ｄｉｓｃはレーザを使用してデータを光学的に再生する。上記の組み合わせもコンピュータ可読媒体の範囲に含まれるべきである。

【０２２０】

命令は、１つまたは複数のデジタル信号プロセッサ（ＤＳＰ）、汎用マイクロプロセッサ、特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブルロジックアレイ（ＦＰＧＡ）、またはその他の同等の集積または個別論理回路などの、１つまたは複数のプロセッサによって実行され得る。したがって、本開示では「プロセッサ」という用語は、上記の構造体のいずれか、または本開示に記載の技術の実装に適するその他の任意の構造体を指し得る。さらに、態様によっては、本開示に記載の機能を、符号化および復号のために構成された専用のハードウェアおよび／またはソフトウェアモジュール内に設けるか、または複合コーデックに組み込むことができる。

【０２２１】

本開示の技術は、ワイヤレスハンドセット、集積回路（ＩＣ）、または１組のＩＣ（例えばチップセット）を含む様々なデバイスまたは装置で実装可能である。本開示では、本開示の技術を実行するように構成されたデバイスの機能的態様を強調して説明するために様々な構成要素、モジュール、またはユニットについて述べたが、これらは必ずしも異なるハードウェアユニットによる実現を必要としない。上述のように、適切なソフトウェアおよび／またはファームウェアとともに、様々なユニットを１つのコーデックハードウェアユニット内に統合するか、または上述のように１つまたは複数のプロセッサを含む相互作用可能なハードウェアユニットの集合によって提供してもよい。

【０２２２】

以上、様々な例について説明した。これらおよびその他の例は以下の特許請求の範囲に含まれる。

以下に、本願出願の当初の特許請求の範囲に記載された発明を付記する。

［１］ビデオデータを符号化する方法であって、

オフセット構文要素のそれぞれの長さが第１の構文要素に基づいて判断可能であり、一連のオフセット構文要素内のオフセット構文要素の数が第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要

素に基づいて判断可能な、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とをエントロピー符号化することと、
エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含むビットストリームを生成することとを備える方法。
［２］前記符号化ピクチャ区画のそれぞれは、単一のエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのグループを含む［１］に記載の方法。
［３］前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す［１］に記載の方法。
［４］前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含む補助強化情報（ＳＥＩ）メッセージを生成することをさらに備える［１］に記載の方法。
［５］前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む［４］に記載の方法。
［６］前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを生成することをさらに備える［５］に記載の方法。
［７］前記オフセット構文要素のそれぞれの長さは前記第１の構文要素の値に１０を加えた値に等しい［１］に記載の方法。
［８］オフセット構文要素のそれぞれの長さが第１の構文要素に基づいて判断可能であり、一連のオフセット構文要素内のオフセット構文要素の数が第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能な、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とをエントロピー符号化し、
エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含むビットストリームを生成するように構成された１つまたは複数のプロセッサを備えるコンピューティングデバイス。
［９］前記符号化ピクチャ区画のそれぞれは、単一のエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのグループを含む［８］に記載のコンピューティングデバイス。
［１０］前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す［８］に記載のコンピューティングデバイス。
［１１］前記１つまたは複数のプロセッサは、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含む補助強化情報（ＳＥＩ）メッセージを生成するように構成された［８］に記載のコンピューティングデバイス。
［１２］前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む［１１］に記載のコンピューティングデバイス。
［１３］前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを生成することをさらに備える［８］に記載のコンピューティングデバイス。
［１４］前記オフセット構文要素のそれぞれの長さは前記第１の構文要素の値に１０を加えた値に等しい［８］に記載のコンピューティングデバイス。
［１５］前記コンピューティングデバイスはビデオエンコーダを備える［８］に記載のコンピューティングデバイス。
［１６］オフセット構文要素のそれぞれの長さが第１の構文要素に基づいて判断可能であり、一連のオフセット構文要素内のオフセット構文要素の数が第２の構文要素に基づい

て判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能な、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とをエントロピー符号化するための手段と、
エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含むビットストリームを生成するための手段とを備えるコンピューティングデバイス。
［１７］コンピューティングデバイスの１つまたは複数のプロセッサによって実行されると、前記コンピューティングデバイスを、
オフセット構文要素のそれぞれの長さが第１の構文要素に基づいて判断可能であり、一連のオフセット構文要素内のオフセット構文要素の数が第２の構文要素に基づいて判断可能であり、ＮＡＬユニット内の複数の符号化ピクチャ区画の位置が前記オフセット構文要素に基づいて判断可能な、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とをエントロピー符号化し、
エントロピー符号化された前記第１の構文要素を表す可変長値と、エントロピー符号化された前記第２の構文要素を表す可変長値と、前記オフセット構文要素を表す固定長値とを含むビットストリームを生成するように構成する命令を記憶するコンピュータ可読記憶媒体。
［１８］ビデオデータを復号する方法であって、
第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連の固定長オフセット構文要素とを含むビットストリームを受信することと、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号することと、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断することと、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断することと、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析することと、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断することと、
前記符号化ピクチャ区画を復号することとを備える方法。
［１９］前記符号化ピクチャ区画のそれぞれは、単一のエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのグループを含む［１８］に記載の方法。
［２０］前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す［１８］に記載の方法。
［２１］前記ビットストリームは、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含むＳＥＩメッセージを含む［１８］に記載の方法。
［２２］前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む［２１］に記載の方法。
［２３］前記ビットストリームは、前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを含む［１８］に記載の方法。
［２４］前記ＳＥＩ　ＮＡＬユニットは、前記ビットストリーム内で、前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にあり、
前記方法は、ビットストリーム内で前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にあるＳＥＩ　ＮＡＬユニットに少なくとも部分的に基づいて、前記ＳＥＩメッセージが前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットに適用可能である

と判断することをさらに備える［２３］に記載の方法。
［２５］前記符号化ピクチャ区画の並列復号のために前記符号化ピクチャ区画を異なる処理コアに供給することと、前記処理コアにおいてそれぞれの前記符号化ピクチャ区画を復号することとをさらに備える［１８］に記載の方法。
［２６］前記オフセット構文要素のそれぞれの長さは前記第１の構文要素の値に１０を加えた値に等しい［１８］に記載の方法。
［２７］第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連の固定長オフセット構文要素とを含むビットストリームを受信し、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号し、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断し、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断し、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析し、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断し、
前記符号化ピクチャ区画を復号するように構成された１つまたは複数のプロセッサを備えるコンピューティングデバイス。
［２８］前記符号化ピクチャ区画のそれぞれは、単一のエントロピースライス、タイル、または波面並列処理（ＷＰＰ）波に関連付けられた、前記ＮＡＬユニット内のコード化ツリーブロックのグループを含む［２７］に記載のコンピューティングデバイス。
［２９］前記一連のオフセット構文要素は、前記ＮＡＬユニット内の先行する符号化ピクチャ区画を基準にした前記符号化ピクチャ区画のバイトオフセットを示す［２７］に記載のコンピューティングデバイス。
［３０］前記ビットストリームは、前記第１の構文要素と、前記第２の構文要素と、前記一連のオフセット構文要素とを含むＳＥＩメッセージを含む［２７］に記載のコンピューティングデバイス。
［３１］前記ＳＥＩメッセージは、前記ＳＥＩメッセージがコード化スライスＮＡＬユニット内の符号化ピクチャ区画の入口点を示すＳＥＩメッセージのタイプに属することを示す構文要素を含む［３０］に記載のコンピューティングデバイス。
［３２］前記ビットストリームは、前記ＳＥＩメッセージを含むＳＥＩ　ＮＡＬユニットを含む［２７］に記載のコンピューティングデバイス。
［３３］前記ＳＥＩ　ＮＡＬユニットは、前記ビットストリーム内で、前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にあり、
前記１つまたは複数のプロセッサは、ビットストリーム内で前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットの直前にあるＳＥＩ　ＮＡＬユニットに少なくとも部分的に基づいて、前記ＳＥＩメッセージが前記複数の符号化ピクチャ区画を含む前記ＮＡＬユニットに適用可能であると判断するように構成された［３２］に記載のコンピューティングデバイス。
［３４］前記１つまたは複数のプロセッサは、前記符号化ピクチャ区画の並列復号のために前記符号化ピクチャ区画を異なる処理コアに供給するように構成された［２７］に記載のコンピューティングデバイス。
［３５］前記コンピューティングデバイスはビデオデコーダを備える［２７］に記載のコンピューティングデバイス。
［３６］前記第１の構文要素の値に１０を加えた値が前記一連のオフセット構文要素のビット数で表した長さに等しい［２７］に記載のコンピューティングデバイス。
［３７］第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連の固定長オフセット構文要素とを含む

ビットストリームを受信するための手段と、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号するための手段と、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断するための手段と、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断するための手段と、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析するための手段と、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断するための手段と、
前記符号化ピクチャ区画を復号するための手段とを備えるコンピューティングデバイス。
［３８］コンピューティングデバイスの１つまたは複数のプロセッサによって実行されると、前記コンピュータを、
第１の構文要素を表すエントロピー符号化された可変長値と、第２の構文要素を表すエントロピー符号化された可変長値と、一連の固定長オフセット構文要素とを含むビットストリームを受信し、
前記第１の構文要素と前記第２の構文要素とをエントロピー復号し、
前記第１の構文要素に基づいて前記オフセット構文要素のそれぞれの長さを判断し、
前記第２の構文要素に基づいて前記一連のオフセット構文要素内のオフセット構文要素の数を判断し、
前記オフセット構文要素のそれぞれの前記長さと前記一連のオフセット構文要素内のオフセット構文要素の前記数とに少なくとも部分的に基づいて前記オフセット構文要素を構文解析し、
前記オフセット構文要素に基づいてネットワーク抽象化レイヤ（ＮＡＬ）ユニット内の複数の符号化ピクチャ区画の位置を判断し、
前記符号化ピクチャ区画を復号するように構成する命令を記憶するコンピュータ可読記憶媒体。

【図１】

図 1

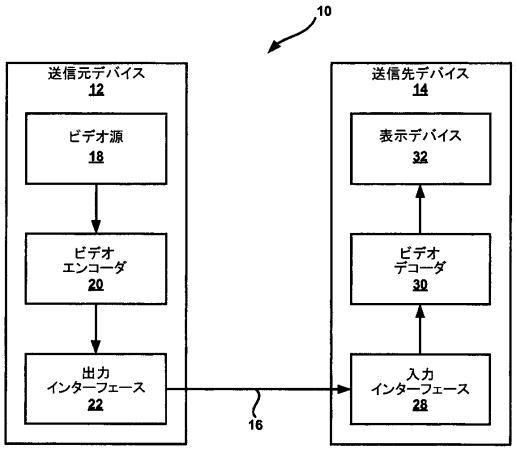


FIG. 1

【図２】

図 2

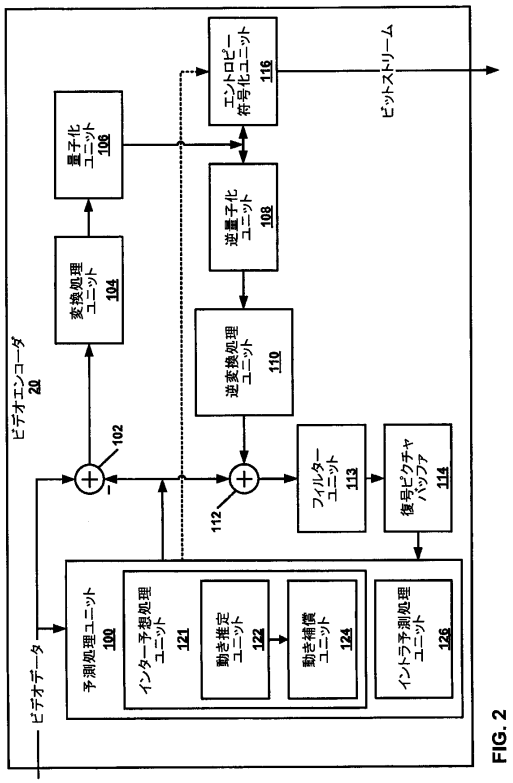


FIG. 2

【図３】

図 3

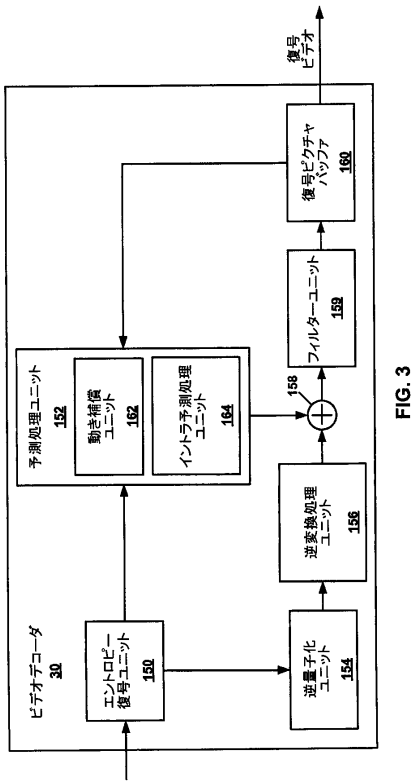


FIG. 3

【図４】

図 4

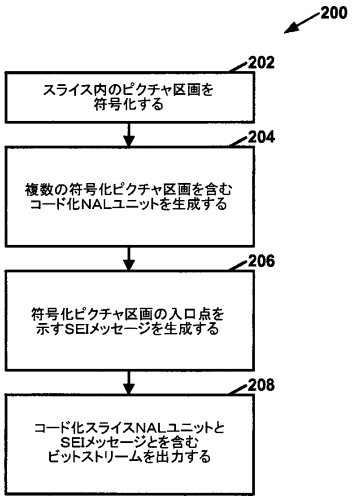


FIG. 4

【図５】

図５

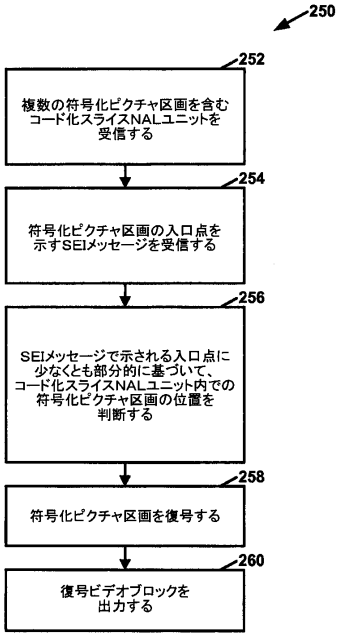


FIG. 5

【図６】

図６

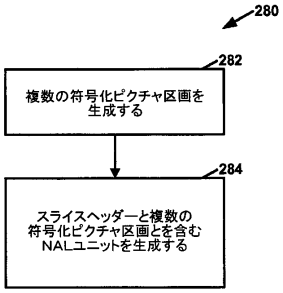


FIG. 6

【図７】

図７

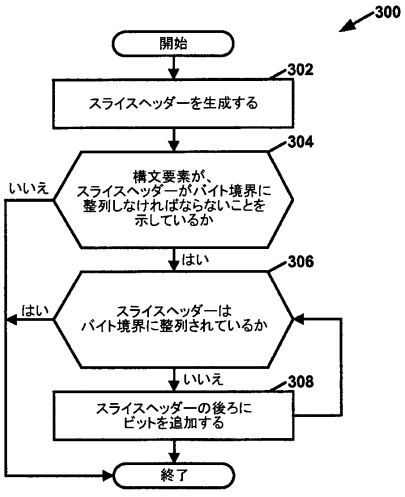


FIG. 7

【図８】

図８

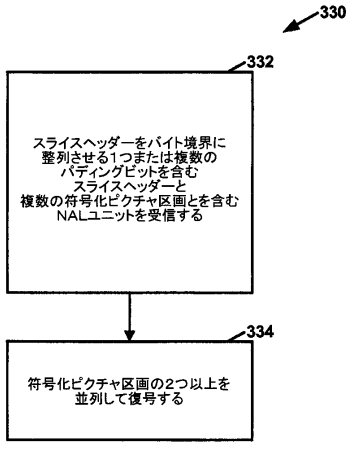


FIG. 8

【図９】

図 9

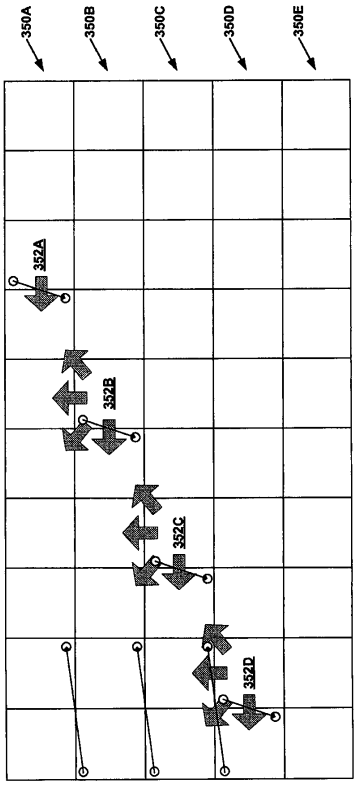


FIG. 9

【図１０】

図 10

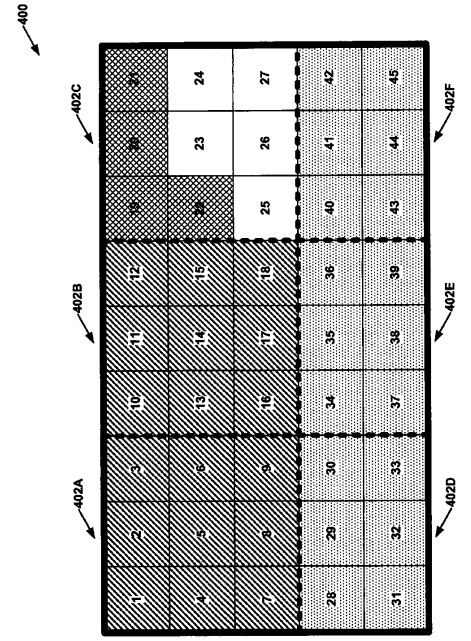


FIG. 10

【図１１】

図 11

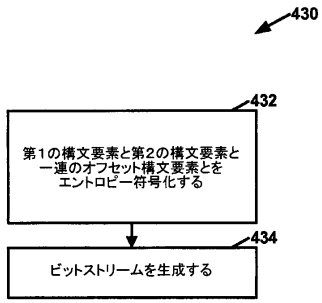


FIG. 11

【図１２】

図 12

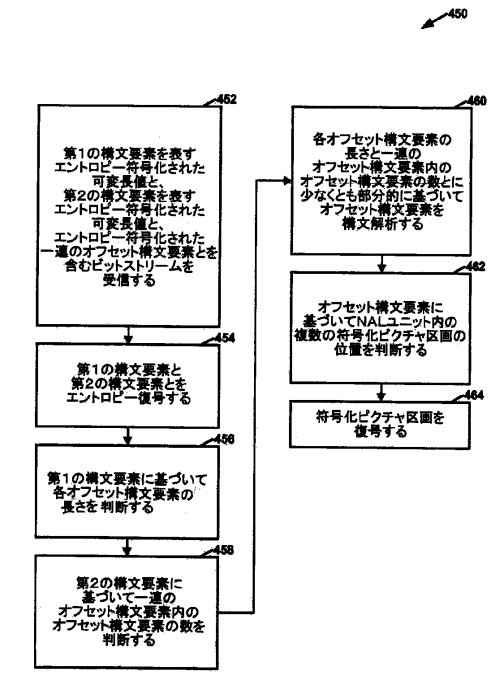


FIG. 12

フロントページの続き

(31)優先権主張番号　13/666,442
(32)優先日　平成24年11月1日(2012.11.1)
(33)優先権主張国　米国(US)

(74)代理人　100153051
弁理士　河野　直樹
(74)代理人　100140176
弁理士　砂川　克
(74)代理人　100158805
弁理士　井関　守三
(74)代理人　100179062
弁理士　井上　正
(74)代理人　100124394
弁理士　佐藤　立志
(74)代理人　100112807
弁理士　岡田　貴志
(74)代理人　100111073
弁理士　堀内　美保子
(72)発明者　ワン、イェークイ
アメリカ合衆国、カリフォルニア州　９２１２１、サン・ディエゴ、モアハウス・ドライブ　５７７５

審査官　堀井　啓明

(56)参考文献　国際公開第２０１０／０３０７５２（WO，Ａ１）
特開２００６－３１９９４３（ＪＰ，Ａ）
米国特許出願公開第２００６／０２３３５２５（ＵＳ，Ａ１）
Kiran Misra,et.al., New results for parallel decoding for Tiles, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting:Torino,IT,14-22 July,2011, 米国, JCTVC, ２０１１年　7月１７日, JCTVC-F594, P.1-P.6, ＵＲＬ, http://phenix.it-sudparis.eu/jct/index.php
Gordon Clare,et.al., Wavefront and Cabac Flush: Different Degrees of Parallelism Without Transcoding, Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting:Torino,IT,14-22 July,2011, 米国, JCTVC, ２０１１年　7月１５日, JCTVC-F275, ＵＲＬ, http://phenix.it-sudparis.eu/jct/index.php

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｈ０４Ｎ１９／００－１９／９８