

(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2019-17104  
(P2019-17104A)

(43) 公開日 平成31年1月31日(2019.1.31)

(51) Int.Cl.		F I	テーマコード (参考)
<b>HO4N 19/70</b>	<b>(2014.01)</b>	HO4N 19/70	5C159
<b>HO4N 19/91</b>	<b>(2014.01)</b>	HO4N 19/91	

審査請求 有 請求項の数 9 O L (全 24 頁)

<p>(21) 出願番号 特願2018-186688 (P2018-186688)</p> <p>(22) 出願日 平成30年10月1日 (2018.10.1)</p> <p>(62) 分割の表示 特願2016-230306 (P2016-230306) の分割</p> <p>原出願日 平成24年7月10日 (2012.7.10)</p> <p>(31) 優先権主張番号 13/180,419</p> <p>(32) 優先日 平成23年7月11日 (2011.7.11)</p> <p>(33) 優先権主張国 米国 (US)</p>	<p>(71) 出願人 517189537 ヴェロス メディア インターナショナル リミテッド Velos Media International Limited アイルランド国 ダブリン 18, キャ リックマイنز, ザ パーク, ザ ハ イド ビルディング, ユニット 32 Unit 32, the Hyde B uilding, The Park, Carrickmines, Dubli n 18 Ireland</p> <p>(74) 代理人 100105050 弁理士 鷲田 公一</p>
---	--

最終頁に続く

(54) 【発明の名称】 ビデオを復号化する方法、ビデオを符号化する方法、デコーダ、エンコーダ、復号プログラムを記録したコンピュータ読み取り可能な記録媒体、および符号化プログラムを記録したコンピュー

## (57) 【要約】

【課題】復号処理の速度を向上させ、メモリ要件を低減することができる、並列復号に適しているビデオ復号方法を提供する。

【解決手段】ビデオを復号する方法は、少なくとも1つのスライスと少なくとも1つのタイルとを含むビデオのフレームを受信するステップと、少なくとも1つのタイルの位置情報がスライス内で送信されているかどうかを示す第1の情報を受信するステップと、少なくとも1つのタイルの位置を含む第2の情報を受信するステップと、を含む。タイルの終端を示すフラグを受信した場合、ビットストリーム内のビットは、バイト・アラインされた位置まで読み込まれ、エン트로ピーデコーダにおける復号処理が初期化されるステップを備える。

【選択図】なし

## 【特許請求の範囲】

## 【請求項 1】

ビデオを復号化する方法であって、

タイルが有効かどうかを示す第 1 フラグを復号し、タイルが有効であると示された場合

、

( a ) 少なくとも 1 つのスライスと少なくとも 1 つのタイルとを含む前記ビデオのフレームを復号するステップであって、

前記少なくとも 1 つのタイルの特徴は、前記フレームの矩形領域であり、且つ複数の符号化単位を有することである、ステップと、

( b ) 前記少なくとも 1 つのスライスのスライスヘッダーから、前記少なくとも 1 つのタイルの位置情報が前記スライスヘッダーにおいて送信されているかどうかを示す第 1 情報を復号するステップと、

( c ) 前記第 1 情報が、前記少なくとも 1 つのタイルの位置情報が前記スライスヘッダーにおいて送信されていることを示す場合に、前記スライスヘッダーから、前記少なくとも 1 つのタイルの開始のバイト・アラインされた位置を示す第 2 情報を復号するステップと、

( d ) 前記少なくとも 1 つのタイルの終端において、前記タイルの終端を示す第 2 フラグを復号し、かつ、前記タイルの開始のバイト・アラインされた位置に到達するまでパディングビットを復号するステップと、

( e ) 前記少なくとも 1 つのタイルの各々の開始のバイト・アラインされた位置において、前記各ステップの復号におけるエン트로ピ復号処理を初期化するステップと、

( f ) 前記少なくとも 1 つのタイルに含まれる前記複数の符号化単位を、ラスタ走査の順序で復号するステップと、

を備える、方法。

## 【請求項 2】

前記スライスは、少なくとも 1 つのタイルを含む、

請求項 1 に記載の方法。

## 【請求項 3】

エン트로ピ初期化インジケータ値が前記スライスヘッダーから復号される、

請求項 1 に記載の方法。

## 【請求項 4】

前記復号処理を初期化するステップは、前記エン트로ピ初期化インジケータ値に基づく

請求項 3 に記載の方法。

## 【請求項 5】

ビデオを符号化する方法であって、

( a ) 少なくとも 1 つのスライスと少なくとも 1 つのタイルとを含む前記ビデオのフレームのビットストリームを符号化するステップであって、

前記少なくとも 1 つのタイルの特徴は、前記フレームの矩形領域であり、且つ複数の符号化単位を有することである、ステップと、

( b ) 前記タイルが有効かどうかを示す第 1 フラグを符号化し、前記タイルが有効であると示された場合、前記少なくとも 1 つの前記タイルの位置情報が前記少なくとも 1 つのスライスのスライスヘッダーにおいて送信されているかどうかを示す第 1 情報であって前記スライスヘッダーにおいて送信される前記第 1 情報を符号化するステップと、

( c ) 前記第 1 情報が、前記少なくとも 1 つのタイルの位置情報が前記スライスヘッダーにおいて送信されていることを示す場合に、前記少なくとも 1 つのタイルの開始のバイト・アラインされた位置を示す第 2 情報であって前記スライスヘッダーにおいて送信される前記第 2 情報を符号化するステップと、

( d ) 前記少なくとも 1 つのタイルの終端において、前記タイルの終端を示す第 2 フラグを符号化し、かつ、前記タイルの開始のバイト・アラインされた位置に到達するまでパディングビットを符号化するステップと、

10

20

30

40

50

(e) 前記少なくとも1つのタイルの各々の開始のバイト・アラインされた位置において、前記各ステップの符号化におけるエントロピ符号化処理を初期化するステップと、

(f) 前記少なくとも1つのタイルに含まれる前記複数の符号化単位を、ラスタ走査の順序で符号化するステップと、  
を備える方法。

【請求項6】

ビデオを復号化するデコーダであって、回路及びメモリを備え、前記メモリは、前記回路と協働して前記デコーダに少なくとも、

タイルが有効かどうかを示す第1フラグを復号し、タイルが有効であると示された場合、少なくとも1つのスライスと少なくとも1つのタイルとを含む前記ビデオのフレームを復号するステップであって、前記少なくとも1つのタイルの特徴は、前記フレームの矩形領域であり、且つ複数の符号化単位を有することである、ステップと、

前記少なくとも1つのタイルの各々の開始のバイト・アラインされた位置において、前記デコーダにおけるエントロピ復号処理を初期化するステップと、

を実行させ、

前記デコーダは、

前記少なくとも1つのスライスのスライスヘッダーから、前記少なくとも1つのタイルの位置情報が前記スライスヘッダーにおいて送信されているかどうかを示す第1情報を復号し、

前記第1情報が、前記少なくとも1つのタイルの位置情報が前記スライスヘッダーにおいて送信されていることを示す場合に、前記スライスヘッダーから、前記少なくとも1つのタイルの開始のバイト・アラインされた位置を示す第2情報を復号し、

前記少なくとも1つのタイルの終端において、前記タイルの終端を示す第2フラグを復号し、かつ、前記タイルの開始のバイト・アラインされた位置に到達するまでパディングビットを復号し、

前記少なくとも1つのタイルに含まれる前記複数の符号化単位を、ラスタ走査の順序で復号する、

デコーダ。

【請求項7】

ビデオを符号化するエンコーダであって、回路及びメモリを備え、前記メモリは、前記回路と協働して前記エンコーダに、

少なくとも1つのスライスと少なくとも1つのタイルとを含む前記ビデオのフレームのビットストリームを符号化するステップであって、前記少なくとも1つのタイルの特徴は、前記フレームの矩形領域であり、且つ複数の符号化単位を有することである、ステップと、

前記少なくとも1つのタイルの各々の開始のバイト・アラインされた位置において、前記エンコーダにおけるエントロピ符号化処理を初期化するステップと、

を少なくとも実行させ、

前記エンコーダは、

前記タイルが有効かどうかを示す第1フラグを符号化し、前記タイルが有効であると示された場合、前記少なくとも1つの前記タイルの位置情報が前記少なくとも1つのスライスのスライスヘッダーにおいて送信されているかどうかを示す第1情報であって前記スライスヘッダーにおいて送信される前記第1情報を符号化し、

前記第1情報が、前記少なくとも1つのタイルの位置情報が前記スライスヘッダーにおいて送信されていることを示す場合に、前記少なくとも1つのタイルの開始のバイト・アラインされた位置を示す第2情報であって前記スライスヘッダーにおいて送信される前記第2情報を符号化し、

前記少なくとも1つのタイルの終端において、前記タイルの終端を示す第2フラグを符号化し、かつ、前記タイルの開始のバイト・アラインされた位置までパディングビットを符号化し、

10

20

30

40

50

前記少なくとも1つのタイルに含まれる前記複数の符号化単位を、ラスタ走査の順序で符号化する、  
エンコーダ。

【請求項8】

請求項6に記載のデコーダとしてコンピュータを機能させるための復号プログラムを記録したコンピュータ読み取り可能な記録媒体。

【請求項9】

請求項7に記載のエンコーダとしてコンピュータを機能させるための符号化プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願の相互参照は無い。

【背景技術】

【0002】

本発明は、ビデオの符号化及び復号に関する。

【0003】

デジタル・ビデオは、典型的には、一連の画像またはフレームとして表され、これらの各々が画素の配列を含む。各画素は、輝度及び/または色情報などの情報を含む。多くの場合、各画素は3色のセットとして表され、その各々は8ビットのカラー値によって定義される。

【0004】

ビデオ符号化方式、例えば、H.264/MPEG-4 AVC (H.264/AVC) は、一般的に複雑性の増大と引き換えに、より高い符号化効率を提供する。また、ビデオ符号化方式に対しての、ビデオ品質の要求の増大と、ビデオ解像度の要求の増大も、符号化の複雑さを増大させることになる。並列復号に適しているビデオ・デコーダは、復号処理の速度を向上させ、メモリ要件を低減することができ、そして並列符号化に適しているビデオ・エンコーダは、符号化処理の速度を向上させ、メモリ要件を低減することができる。

【0005】

H.264/MPEG-4 AVC [ITU-T VCEG及びISO/IEC MPEGの合同ビデオ・チーム、「H.264:オーディオ・ビジュアル・サービス全般のための先進ビデオ符号化方式、(H.264:Advanced video coding for generic audiovisual services)」ITU-T 勧告 H.264及びISO/IEC 14496-10 (MPEG4-第10部)、2007年11月]及び同様にJCT-VC, [「検討中のテスト・モデル案(Draft Test Model Under Consideration)」JCTVC-A205、JCT-VC会議、ドレスデン、2010年4月(JCT-VC)]は、圧縮効率のために、ビデオ・シーケンスにおいて、マクロブロック予測及びそれに続く残差符号化を使用し、時間的、空間的冗長性を低減するビデオ・コーデック(エンコーダ/デ

【0006】

本発明の前述の及び他の目的、特徴、及び利点は、添付の図面と併せて、本発明の以下の詳細の説明を考慮して、より容易に理解されるであろう。

【発明の概要】

【課題を解決するための手段】

【0007】

本発明の1つの実施形態は、ビデオを復号する方法であって、

(a) 少なくとも1つのスライスと少なくとも1つのタイルとを含む前記ビデオのフレームを受信するステップであって、

10

20

30

40

50

前記少なくとも1つのスライスの各々は、他の前記少なくとも1つのスライスとは独立に復号され、且つ前記少なくとも1つのタイルから構成され、

前記少なくとも1つのタイルの特徴は、前記フレームの矩形領域であり、且つラスタ走査の順序に配置されている前記復号するための符号化単位を有することである、ステップと、

(b) 前記少なくとも1つのタイルの位置情報が前記スライス内で送信されているかどうかを示す第1の情報を受信するステップと、

(c) 前記少なくとも1つのタイルを復号することに適する第2の情報を受信するステップであって、

前記第2の情報は前記少なくとも1つのタイルの位置を含む、ステップとを備える、ビデオを復号する方法を開示する。

10

#### 【0008】

本発明の別の実施形態は、ビデオを復号する方法であって、

(a) 少なくとも1つのスライスと少なくとも1つのタイルとを含む前記ビデオのフレームを受信するステップであって、前記少なくとも1つのスライスの各々の特徴は、他の前記少なくとも1つのスライスとは独立に復号されることであり、前記少なくとも1つのタイルの各々の特徴は、前記フレームの矩形領域であり、且つラスタ走査の順序に配置されている、前記復号するための符号化単位を有することであり、前記フレームの前記少なくとも1つのタイルは、前記フレームのラスタ走査の順序に集合的に配置されている、ステップと、

20

(b) 前記タイルの少なくとも1つの最初の最大符号化単位からデータのセットを受信するステップであって、前記データのセットは、(1) 最大符号化単位アドレスを表す値と、(2) スライス指標を表す値と、(3) 量子化パラメータ情報を表す値とを含む、ステップと

を備える、ビデオを復号する方法を開示する。

#### 【図面の簡単な説明】

#### 【0009】

【図1】 H.264 / AVC ビデオ・エンコーダを示す。

【図2】 H.264 / AVC ビデオ・デコーダを示す。

【図3】 例示的なスライス構造を示す。

【図4】 別の例示的なスライス構造を示す。

【図5】 エントロピ・スライスの再構築を示す。

【図6】 エントロピ・スライスの並列再構築を示す。

【図7】 1つのスライスと9つのタイルを有するフレームを示す。

【図8】 3つのスライスと3つのタイルを有するフレームを示す。

【図9A】 タイルに対してのエントロピの選択を示す。

【図9B】 タイルに対してのエントロピの選択を示す。

【図10A】 タイルに対しての別のエントロピの選択を示す。

【図10B】 タイルに対しての別のエントロピの選択を示す。

【図11】 タイルに対してのさらに別のエントロピの選択を示す。

【図12A】 例示的なシンタックスを示す。

【図12B】 例示的なシンタックスを示す。

【図13】 タイルの開始に関する明示的な信号を示す。

【図14】 タイルの開始に関する黙示的な信号を示す。

【図15】 ライト・ウエイト・タイル・ヘッダーのセットを示す。

【図16】 スライス・パラメータ指標セットのセットを示す。

【図17】 タイル終了フラグを示す。

#### 【発明を実施するための形態】

#### 【0010】

エントロピ符号化 / 復号を使用する任意のビデオ・コーダ / デコーダ (コーデック) が

30

40

50

本明細書に記載の実施形態に適合可能であるが、例示的な実施形態を、単に例示の目的のために H. 264 / AVC エンコーダと H. 264 / AVC デコーダに関連させて説明する。多数のビデオ符号化方式は、ブロック・ベースのハイブリッド・ビデオ符号化のアプローチに基づいており、情報源符号化方式は、フレーム間予測とも称されるピクチャ間予測、フレーム内予測とも称されるピクチャ内予測、及び予測残差の変換符号化のハイブリッドである。フレーム間予測は、時間的冗長性を利用するものであり、フレーム内と予測残差の変換符号化は空間的冗長性を利用するものである。

#### 【0011】

図1は、例示的な H. 264 / AVC ビデオ・エンコーダ 2 を示す。フレームとも称される入力ピクチャ 4 を、符号化のために供給することができる。予測信号 6 及び残差信号 8 が生成され、予測信号 6 は、フレーム間予測 10 またはフレーム内予測 12 のいずれかに基づくものとすることができる。フレーム間予測 10 は、1つ以上の記憶された、参照フレームとも称される参照ピクチャ 16、及び入力フレーム 4 と参照フレーム 16 との間の動き推定セクション 18 の処理によって決定される動き情報 19 を使用する動き補償セクション 14 によって決定することができる。フレーム内予測 12 は、復号信号 22 を使用してフレーム内予測セクション 20 によって決定することができる。残差信号 8 は、予測信号 6 から入力フレーム 4 を減算することによって決定することができる。残差信号 8 は、変換 / スケーリング / 量子化セクション 24 によって、変換、スケーリング、量子化され、量子化変換係数 26 が生成される。復号信号 22 は、予測信号 6 を、量子化変換係数 26 を使用して逆（変換 / スケーリング / 量子化）セクション 30 によって生成された信号 28 に加算して生成することができる。動き情報 19 と量子化変換係数 26 を、エントロピ符号化セクション 32 によって、エントロピ符号化し、圧縮ビデオ・ビットストリーム 34 に書き込むことができる。例えば参照フレームの一部である出力画像領域 38 は、エンコーダ 2 において、再構築されたフィルタ前の信号 22 を使用してデブロッキング・フィルタ 36 によって生成されてもよい。この出力フレームは、後続の入力ピクチャを符号化するための参照フレームとして使用することができる。

#### 【0012】

図2は、例示的な H. 264 / AVC ビデオ・デコーダ 50 を示す。ビットストリームとも称される入力信号 52 を、復号のために供給することができる。受信したシンボルを、エントロピ復号セクション 54 によってエントロピ復号し、動き情報 56 と、イントラ予測情報 57 と、量子化され、スケーリングされた変換係数 58 とを生成することができる。動き情報 56 は、動き補償セクション 60 によってフレーム・メモリ 64 に存在する1つ以上の参照フレーム 84 の一部と組み合わせることができ、そしてフレーム間予測 68 を生成することができる。量子化され、スケーリングされた変換係数 58 を、逆量子化し、スケーリングし、逆変換し、復号された残差信号 70 を生成することができる。残差信号 70 を、予測信号、すなわちフレーム間予測信号 68 またはフレーム内予測信号 76 のいずれかに加算することができる。フレーム内予測情報は、フレーム内予測セクション 74 によって、現フレームの既に復号された情報 72 と組み合わせられ、加算された信号 72 を生成することができる。加算された信号 72 は、デブロッキング・フィルタ 80 でフィルタ処理し、フィルタ処理された信号 82 は、フレーム・メモリ 64 に書き込むことができる。

#### 【0013】

H. 264 / AVC において、入力ピクチャは、固定サイズのマクロブロックに区分することができる。各マクロブロックは、輝度成分の  $16 \times 16$  サンプルと、2つの色差成分の各々  $8 \times 8$  サンプルとの矩形のピクチャ領域をカバーする。H. 264 / AVC 標準の復号処理は、マクロブロックの処理単位を規定している。エントロピ・デコーダ 54 は、圧縮ビデオ・ビットストリーム 52 のシンタックス要素を構文解析し、それらを多重分離する。H. 264 / AVC は、2つの異なるエントロピ復号方法、すなわち、CAVLC と呼ばれ、コンテキスト適応切り替え可変長符号のセットの使用に基づく複雑性の低い方式と、CABAC と呼ばれ、計算量がより厳しく要求される、コンテキスト・ベースの適

10

20

30

40

50

応型 2 値算術符号化の方式と、を指定する。そのようなエントロピ復号方式の両方において、現在のシンボルの復号は、前に正しく復号されたシンボルと、適応的に更新されたコンテキスト・モデルとに依存することができる。更に、例えば、予測データ情報、残差データ情報、異なる色平面などの異なるデータ情報は、一緒に多重化することができる。多重分離は、要素がエントロピ復号されるまで待つことが可能である。

**【 0 0 1 4 】**

エントロピ復号した後に、マクロブロックを、逆量子化及び逆変換を経た残差信号と、フレーム内予測信号またはフレーム間予測信号のいずれかの予測信号とを取得して再構築することができる。ブロック歪みは、復号されたマクロブロックにデブロッキング・フィルタを適用することにより低減できる。典型的には、そのような後続の処理は、入力信号がエントロピ復号された後に開始されるので、エントロピ復号が復号において潜在的ボトルネックとなってしまう。同様に、例えば、H. 264 / AVCにおけるレイヤー間予測または他のスケーリング可能なコーデックにおけるレイヤー間予測など、代替の予測メカニズムが使用されるコーデックにおいて、エントロピ復号は、デコーダにおける処理の前に行う必要がある場合があるので、エントロピ復号が潜在的ボトルネックとなる。

10

**【 0 0 1 5 】**

複数のマクロブロックを含む入力ピクチャは、1つまたは数個のスライスに区分することができる。1つのスライスが表すピクチャの領域におけるサンプルの値は、エンコーダとデコーダとで使用した参照ピクチャが同一であり、デブロッキング・フィルタリングがスライスの境界を越えた情報を使用しない場合、他のスライスからのデータを使わずに、適切に復号され得る。それ故に、1つのスライスに対するエントロピ復号とマクロブロックの再構築は、他のスライスに依存しない。特に、エントロピ復号状態は、各スライスの開始時にリセットしてもよい。エントロピ復号及び再構築の両方について、近傍の利用可能性を定義する時に、他のスライスのデータは、利用不可とマークすることができる。スライスは、並列に、エントロピ復号され、再構築することが可能である。イントラ予測及び動きベクトル予測は、好ましくは、スライスの境界を越えることは許可されない。これに対し、デブロッキング・フィルタリングは、スライス境界をまたがる情報を使用することができる。

20

**【 0 0 1 6 】**

図 3 は、水平方向に 11 個のマクロブロックと垂直方向に 9 個のマクロブロックとを含む例示的なビデオ・ピクチャ 90 を示す ( 9 個の例示的なマクロブロックには 91 ~ 99 とラベルが付けられている )。図 3 には、例示的な 3 つのスライスが示されており、第 1 のスライスは「スライス # 0」100 と示され、第 2 のスライスは「スライス # 1」101 と示され、第 3 のスライスは「スライス # 2」102 と示されている。H. 264 / AVC デコーダは、3 つのスライス 100、101、102 を並列に復号し、再構築することが可能である。各スライドは、逐次方式で走査線順に送信することができる。各スライスの復号 / 再構築の処理の開始時に、コンテキスト・モデルは、初期化またはリセットされ、他のスライスのマクロブロックは、エントロピ復号及びマクロブロック再構築の両方に対して利用不可とマークされる。従って、あるマクロブロック、例えば、「スライス # 1」内の 93 とラベルが付いたマクロブロックに対して、「スライス # 0」内のマクロブロック ( 例えば、91 及び 92 とラベルが付けられたマクロブロック ) は、コンテキスト・モデルの選択または再構築に使用できない。一方、あるマクロブロック、例えば、「スライス # 1」内の 95 とラベルが付いたマクロブロックに対して、「スライス # 1」内の他のマクロブロック ( 例えば、93 及び 94 とラベルが付けられたマクロブロック ) は、コンテキスト・モデルの選択または再構築に使用できる。従って、エントロピ復号及びマクロブロック再構築は、1 つのスライス内において、逐次進行する。スライスがフレキシブル・マクロブロック順序 ( FMO ) を使用して定義されていない限り、スライス内のマクロブロックは、ラスタ走査の順序で処理される。

30

40

**【 0 0 1 7 】**

フレキシブル・マクロブロック順序は、ピクチャがスライスに区分される方法を変更す

50

るためにスライス・グループを定義する。スライス・グループ内のマクロブロックは、マクロブロック対スライス・グループのマッピングで定義され、それは、スライス・ヘッダー内のピクチャ・パラメータ・セットの内容及び追加情報によって通知される。マクロブロック対スライス・グループのマッピングは、ピクチャ内の各マクロブロックに対するスライス・グループ識別番号から構成される。スライス・グループ識別番号は、関連するマクロブロックが属するスライス・グループを規定する。各スライス・グループは、1つ以上のスライスに区分することができ、1つのスライスは、同じスライス・グループ内のマクロブロックのシーケンスであり、該シーケンスは、特定のスライス・グループのマクロブロックのセット内でラスタ走査の順序で処理される。エントロピ復号及びマクロブロック再構築は、1つのスライス・グループ内で逐次進行する。

10

#### 【0018】

図4は、3つのスライス・グループへのマクロブロックの割り当ての例示を示し、第1のスライス・グループは、「スライス・グループ#0」103と表され、第2のスライス・グループは、「スライス・グループ#1」104と表され、第3のスライス・グループは、「スライス・グループ#2」105と表されている。これらのスライス・グループ103、104、105は、ピクチャ90において、それぞれ、2つの前景領域と1つの背景領域とに関連するようにしてもよい。

#### 【0019】

ピクチャは、1つ以上の再構築スライスに区分することが可能で、再構築スライスは、エンコーダとデコーダとで使用された参照ピクチャが同一である場合、その再構築スライスが表すピクチャの領域のサンプルの値が他の再構築スライスからのデータを使わずに正しく再構築できるという意味で、自己完結型であり得る。再構築スライス内の全ての再構築マクロブロックは、再構築のための近傍の定義付けに利用可能であってもよい。

20

#### 【0020】

再構築スライスは、2つ以上のエントロピ・スライスに区分することが可能で、エントロピ・スライスは、エントロピ・スライスが表すピクチャの領域におけるシンボル値が他のエントロピ・スライスからのデータを使わずに、正しくエントロピ復号できるという意味で、自己完結型であり得る。エントロピ符号化状態は、各エントロピ・スライスの復号開始時にリセットすることができる。他のエントロピ・スライスのデータは、エントロピ復号のために近傍の利用可能性を定義する際に、利用不可とマークすることが可能である。他のエントロピ・スライスのマクロブロックは、現在のブロックのコンテキスト・モデルの選択に使用できない。コンテキスト・モデルは、エントロピ・スライス内においてのみ更新可能である。従って、エントロピ・スライスに関連した各エントロピ・デコーダは、それ自体のコンテキスト・モデルのセットを維持するとよい。

30

#### 【0021】

エンコーダは、再構築スライスをエントロピ・スライスに区分するか否かを決定し、その決定をビットストリーム中で通知することができる。その信号は、「entropy\_slice\_flag」と示すことができるエントロピ・スライス・フラグを含むことができる。図5を参照して、エントロピ・スライス・フラグを調べることができ(130)、エントロピ・スライス・フラグが、ピクチャまたは再構築スライスに関連したエントロピ・スライスが無い(132)ことを示す場合、ヘッダーは、通常のスライス・ヘッダーとして構文解析することができる(134)。エントロピ・デコーダ状態は、リセットすることが可能で(136)、エントロピ復号及び再構築のための近傍情報を定義することができる(138)。次いで、スライス・データをエントロピ復号することができ(140)、スライスを再構築することが可能である(142)。エントロピ・スライス・フラグが、ピクチャと関連したエントロピ・スライスまたは再構築スライスがある(146)ことを示す場合、ヘッダーは、エントロピ・スライス・ヘッダーとして構文解析することができる(148)。エントロピ・デコーダ状態は、リセットすることが可能で(150)、エントロピ復号のための近傍情報を定義することができ(152)、エントロピ・スライス・データを、エントロピ復号することができる(154)。次いで、再構築のため

40

50

の近傍情報を定義することが可能で(156)、スライスを再構築することができる(142)。スライスの再構築(142)後、次のスライスまたはピクチャが分析されてもよい(158)。

#### 【0022】

図6を参照して、デコーダは、並列復号が可能であり、それ自体の並列性の度合いを定義することができ、例えば、並列にN個のエントロピ・スライスを復号する能力を備えるデコーダを考慮する。このデコーダは、N個のエントロピ・スライスを識別することができる(170)。現在のピクチャまたは再構築スライスにおいて、N個より少ないエントロピ・スライスが利用可能である場合、デコーダは、後続のピクチャまたは再構築スライスからのエントロピ・スライスを、それらが利用可能であれば、復号することが可能である。あるいは、デコーダは、現在のピクチャまたは再構築スライスが完全に処理されるまで待って、その後、後続のピクチャまたは再構築スライスの部分を復号してもよい。最大N個のエントロピ・スライスの識別(170)後、識別されたエントロピ・スライスの各々を、独立にエントロピ復号してもよい。第1のエントロピ・スライスを復号することができる(172~176)。第1のエントロピ・スライスの復号(172~176)は、デコーダ状態のリセット(172)を含むことが可能である。CABACエントロピ復号が使用される場合、CABAC状態をリセットすることができる。第1のエントロピ・スライスのエントロピ復号のための近傍情報を定義することができ(174)、第1のエントロピ・スライス・データを復号することが可能である(176)。最大N個までのエントロピ・スライスの各々に対して、これらのステップを実行することができる(N番目のエントロピ・スライスに対して178~182)。エントロピ・スライスの全部または一部がエントロピ復号されると、デコーダは、エントロピ・スライスを再構築することができる(184)。

10

20

#### 【0023】

エントロピ・スライスがN個より多い場合、復号スレッドは、1個のエントロピ・スライスのエントロピ復号完了後に、次のエントロピ・スライスのエントロピ復号を開始することができる。従って、スレッドが低複雑性のエントロピ・スライスのエントロピ復号を終了すると、そのスレッドは他のスレッドの復号の終了を待たずに、別のエントロピ・スライスの復号を開始することができる。

#### 【0024】

図3に示すように、スライスの配置は、ラスタ走査またはラスタ走査順序としても知られている画像の走査順序に並んだ一対のマクロブロックの間に、各スライスを定義することに限定することができる。この走査順のスライスの配置は、計算効率は良いが、高効率の並列符号化と並列復号には適さない傾向がある。更に、このスライスの走査順序の定義は、符号化効率に非常に適した共通の特性を持っている可能性が高い、画像のより小さな局所領域をグループにしない傾向がある。図4に示したようなスライスの配置は、その配置において柔軟性は高いが、高効率の並列符号化と並列復号には適さない傾向がある。更に、この非常に柔軟なスライスの定義は、デコーダに実装するためには、計算上複雑である。

30

#### 【0025】

図7を参照すると、あるタイル方式では、画像が矩形(正方形を含む)領域のセットに分割されている。タイルの各々の中のマクロブロック(例えば、最大符号化単位)は、ラスタ走査順序に符号化され、復号される。タイルの配置も同様に、ラスタ走査順序に符号化され、復号される。従って、任意の適切な数(例えば、0以上)の列の境界があってもよいし、任意の適切な数(例えば、0以上)の行の境界があってもよい。従って、図7で示されている1つのスライスのように、フレームは、1つ以上のスライスを定義することが可能である。幾つかの実施形態において、異なるタイルに位置するマクロブロックは、近傍マクロブロック情報に依存する、イントラ予測、動き補償、エントロピ符号化コンテキスト選択または他の処理に利用可能ではない。

40

#### 【0026】

50

図 8 を参照すると、このタイル方式では、画像が 3 個の矩形列のセットに分割されて示されている。タイルの各々の中のマクロブロック（例えば、最大符号化単位）は、ラスタ走査順序に符号化され、復号される。タイルも、同様にラスタ走査順序で符号化され、復号される。1 つ以上のスライスをタイルの走査順序で定義することができる。スライスの各々は独立に復号可能である。例えば、スライス 1 は、マクロブロック 1 ~ 9 を含むように定義することができ、スライス 2 は、マクロブロック 10 ~ 28 を含むように定義することができ、スライス 3 は、3 個のタイルに跨るマクロブロック 29 ~ 126 を含むように定義することができる。タイルの使用により、フレームの、より局所的な領域のデータを処理することによって、符号化効率が促進される。

#### 【 0 0 2 7 】

1 つの実施形態において、エントロピ符号化・復号処理は、各タイルの開始時に初期化される。エンコーダにおいて、この初期化には、エントロピ・エンコーダ内の残りの情報をビットストリームに書き込む、フラッシングとして知られる処理と、あらかじめ定義され設定されたビットストリーム位置の 1 つに到達するために、ビットストリームに追加のデータをパディングする処理と、エントロピ・エンコーダを、所定の、またはエンコーダ及びデコーダの両方に知られている既知の状態に設定する処理とを含めることができる。多くの場合、既知の状態は、値のマトリックスの形である。更に、所定のビットストリームの位置は、ビットの倍数でアラインされた、すなわち、バイト・アラインされた位置とすることができる。デコーダにおいて、この初期化処理は、エントロピ・デコーダを、エンコーダ及びデコーダの両方に知られている既知の状態に設定し、ビットストリーム内の

10

20

#### 【 0 0 2 8 】

幾つかの実施形態において、複数の既知の状態が、エンコーダとデコーダに対して利用可能であり、エントロピ符号化処理及び/またはエントロピ復号処理の初期化に使用することができる。従来から、初期化に使われる既知の状態は、エントロピ初期化インジケータ値を有するスライス・ヘッダーで通知されている。図 7 及び図 8 で示したタイル方式では、タイルとスライスとは互いにアラインされていない。従って、タイルとスライスとがアラインされていないので、従来は、スライス内の最初のマクロブロックと同一の位置にある、ラスタ走査順序で最初のマクロブロックを含まないタイルに対して、エントロピ

30

#### 【 0 0 2 9 】

図 8 を参照すると、エントロピ初期化インジケータ値は、3 つのスライスに対して同様に、スライス 1 のマクロブロック 1 に対するスライス・ヘッダーの中に提供され、スライス 2 のマクロブロック 10 に対するスライス・ヘッダーの中に提供され、スライス 3 のマクロブロック 29 に対するスライス・ヘッダーの中に提供されている。しかしながら、図 7 と同様の形で、エントロピ初期化インジケータ値が、中央のタイル（マクロブロック 37 から始まっている）及び右側のタイル（マクロブロック 100 から始まっている）には欠如している。中央及び右側のタイルに対してエントロピ初期化インジケータ値が無ければ、タイルのマクロブロックを、並列方式で高い符号化効率により効果的に符号化及び復号するのに問題がある。

40

#### 【 0 0 3 0 】

1 つのフレーム内に 1 つ以上のタイル及び 1 つ以上のスライスを使用しているシステムでは、エントロピ初期化インジケータ値をタイルの最初のマクロブロック（例えば、最大符号化単位）と共に提供するのが好ましい。図 7 のマクロブロック 16 と共に、エントロ

50

ピ初期化インジケータ値がエントロピ初期化情報を明示的に選択するために提供されている。その明示的決定には、いかなる適切な方式を使ってもよく、例えば、前のスライス・ヘッダー等の中にある前のエントロピ初期化インジケータ値を使用すべきであるということを示すこと、または、そうでなければ、それぞれのマクロブロック/タイルに関連付けられたエントロピ初期化インジケータ値を送信することが可能である。このように、スライスにエントロピ指標値を含むヘッダーを含めることを可能にしながら、タイルの最初のマクロブロックにも同様にエントロピ初期化インジケータ値を含めることができる。

【0031】

図9Aを参照して、この追加情報の符号化は次のようであってもよい。

```

    If ( num__column__minus1 > 0 && num__rows__m
    in1 > 0 )
    then
        tile__cabac__init__idc__present__flag
  
```

10

【0032】

num\_\_column\_\_minus1 > 0 は、タイル内の列の数が0でないかを決定し、num\_\_rows\_\_minus1 > 0 は、タイル内の行の数が0でないかを決定し、両者は実質的に、タイルが符号化/復号で使用されているかを決定する。タイルが使用されている場合、tile\_\_cabac\_\_init\_\_idc\_\_present\_\_flag は、エントロピ初期化インジケータ値がエンコーダからデコーダに伝えられる方法を示すフラグである。例えば、フラグが第1の値に設定されている場合、前に送られたエントロピ初期化インジケータ値を使用するというような第1のオプションを選択することができる。具体的な例として、この前に送られたエントロピ初期化インジケータ値は、タイルの最初のマクロブロックを含んでいるスライスに対応するスライス・ヘッダーで送信されたエントロピ初期化インジケータ値に等しくすることが可能である。例えば、フラグが第2の値に設定されている場合、エントロピ初期化インジケータ値が、対応するタイルのビットストリームの中に提供されるというような第2のオプションが選択されてもよい。具体的な例として、エントロピ初期化インジケータ値は、タイルの最初のマクロブロックに対応するデータの中に提供される。

20

【0033】

エントロピ初期化インジケータ値がエンコーダからデコーダに伝えられる方法を示すフラグを通知するシンタックスは、次のようにすることができる。

30

【0034】

【数1】

```

num_columns_minus1
num_rows_minus1
if (num_column_minus1 > 0 && num_rows_minus1 > 0 {
    tile_boundary_dependence_idr
    uniform_spacing_idr
    if( uniform_spacing_idr != 1) {
        for (i=0; i<num_columns_minus1; i++)
            columnWidth[i]
        for (i=0; i<num_rows_minus1; i++)
            rowHeight[i]
    }
    if( entropy_coding_mode == 1)
        tile_cabac_init_idc_present_flag
}
  
```

40

【0035】

図9Bを参照して、シーケンス・パラメータ・セット（例えば、フレームのシーケンス

50

に関する情報)及び/またはピクチャ・パラメータ・セット(例えば、特定のフレームに関する情報)に含まれるフラグのような他の方式が、タイルが使用されているかを決定するために使われてもよい。

【0036】

シンタックスは、次のようにすることができる。

【0037】

【数2】

```
tile_enable_flag
```

```
if (tile_enable_flag) {
```

```
    num_columns_minus1
```

```
    num_rows_minus1
```

```
    tile_boundary_dependence_idr
```

```
    uniform_spacing_idr
```

```
    if( uniform_spacing_idr !=1) {
```

```
        for (i=0; i<num_columns_minus1; i++)
```

```
            columnWidth[i]
```

```
        for (i=0; i<num_rows_minus1; i++)
```

```
            rowHeight[i]
```

```
    }
```

```
    if( entropy_coding_mode==1)
```

```
        tile_cabac_init_idc_present_flag
```

```
}
```

10

20

30

【0038】

*tile\_enable\_flag*は、タイルが現在のピクチャで使用されているかを決定する。

40

【0039】

図10Aと10Bを参照して、タイルに対する適切なエントロピ初期化インジケータ値情報を提供する方式は次のようにすることができる。

【0040】

第1に、マクロブロック(例えば、符号化単位)が、タイルにおける最初のマクロブロックであるかどうかを確認する。それ故、この方式は、エントロピ初期化インジケータ値を含む可能性があるタイルの最初のマクロブロックを決定する。図7を参照すると、これは、マクロブロック1、16、34、43、63、87、99、109、及び121を指している。図8を参照すると、これは、マクロブロック1、37及び100を指している。

50

## 【0041】

第2に、タイルの最初のマクロブロック（例えば、符号化単位）が、スライスの最初のマクロブロック（例えば、符号化単位）でないことを確認する。それ故、この方式は、スライス内のさらなるタイルを識別する。図7を参照すると、これは、マクロブロック16、34、43、3、87、99、109、及び121を指している。図8を参照すると、これは、マクロブロック37及び100を指している。

## 【0042】

第3に、`tile_cabac_init_idc_flag`が第1の値に等しいかどうか、且つタイルが有効かどうかを確認する。1つの特定の実施形態では、この値は0に等しい。第2の実施形態では、この値は1に等しい。更なる実施形態では、 $(num\_column\_min1 > 0 \ \&\& \ num\_rows\_min1 > 0)$ の場合、タイルは有効化される。別の実施形態では、`tile_enable_flag`が1に等しい場合に、タイルは有効化される。

10

## 【0043】

そのように識別されたマクロブロックに対して、`tile_cabac_init_idc_present_flag`を設定することができる。

## 【0044】

次いで、システムは、`tile_cabac_init_idc_flag`が存在し、且つ $(num\_column\_minus1 > 0 \ \&\& \ num\_rows\_min1 > 0)$ である場合にのみ、`tile_cabac_init_idc_present_flag`の信号を送ればよい。それ故、システムは、タイルが使用されていて、フラグが、エントロピ情報が送信されていること（すなわち、`cabac_init_idc_flag`）を示す場合のみ、エントロピ情報を送信する。

20

## 【0045】

符号化シンタックスは、次のようにすることができる。

## 【0046】

## 【数3】

```

coding_unit (x0, y0, currCodingUnitSize) {
    If          (x0==tile_row_start_location          &&
y0==tile_col_start_location          &&
currCodingUnitSize==MaxCodingUnitSize          &&
tile_cabac_init_idc_flag==true && mb_id!=first_mb_in_slice {
        tile_cabac_init_idc_present_flag
        if (tile_cabac_init_idc_present_flag)
            cabac_init_idc
    }
    a regular coding unit...
}

```

30

40

## 【0047】

一般的に、スライスの最初のマクロブロックに関連していないタイルの最初のマクロブロック（例えば、符号化単位）に関連する1つ以上のフラグが、エントロピ初期化インジケータ値を定義することができる。フラグは、エントロピ初期化インジケータ値が前に提供された情報であるか、デフォルト値であるか、または提供されることになっているエントロピ初期化インジケータ値であることを示すことができる。

50

## 【 0 0 4 8 】

再び図7を参照する。デコーダは、ピクチャ・フレーム内のマクロブロック16の位置を知っているが、エントロピ復号が原因で、マクロブロック15がエントロピ復号されるまで、ビットストリームの中の、マクロブロック16を記述するビットの位置が分からない。復号を行って次のマクロブロックを識別するこのやり方は、ビットのオーバーヘッドが小さく保たれ、好ましい。しかしながら、タイルを並列に復号することには役立たない。フレームの中の特定のタイルに対してビットストリームの中の特定の位置を識別する能力を高めて、エントロピ復号の完了を待たずに、異なるタイルをデコーダで同時に並列に復号できるようにするために、ビットストリームの中でのタイルの位置を識別する信号をビットストリームの中を含めることが可能である。図11を参照する。ビットストリーム 10  
 の中でのタイルの位置の通知は、スライスのヘッダーで提供されるのが好ましい。フラグが、ビットストリームの中でのタイルの位置がスライス内で送信されていることを示す場合、スライス内のタイル(群)それぞれの最初のマクロブロックの、スライス内での位置に加えて、フレーム内のそのようなタイルの数を含むことも好ましい。更に、所望であれば、選択されたタイルのセットに対してのみの位置情報が含まれてもよい。

## 【 0 0 4 9 】

符号化シンタックスは、次のようにすることができる。

## 【 0 0 5 0 】

## 【 数 4 】

```
tile_locations_flag
```

```
if (tile_location_flag) {
```

```
    tile_locations()
```

```
}
```

```
tile_locations()
```

```
{
```

```
    for (i=0; i<num_of_tiles_minus1; i++) {
```

```
        tile_offset[i]
```

```
    }
```

```
}
```

## 【 0 0 5 1 】

タイル位置がビットストリームの中に送信されているかどうかを、`tile_location_flag`で通知する。`tile_offset[i]`は、絶対位置値または差分サイズ値(前に符号化されたタイルに対してのタイル・サイズの相違)または任意の適切な方式を使って、通知することができる。 40

## 【 0 0 5 2 】

この方式は、オーバーヘッドは小さいが、エンコーダは、一般的には、全てのタイルが符号化されるまでビットストリームを送信できない。

## 【 0 0 5 3 】

幾つかの実施形態では、連続するタイルの、最大値とも称される最大絶対位置値または最大差分サイズ値に関するデータを含むのが好ましい。そのような情報により、エンコー 50

ダは、識別された最大値をサポートするために必要なビット数だけを送信でき、デコーダは、識別された最大値をサポートするために必要なビット数だけを受信できる。例えば、比較的小さい最大値に対しては、タイル位置情報には小さいビット深度のみが必要となる。例えば、比較的大きい最大値に対しては、タイル位置情報には大きいビット深度が必要となる。

#### 【0054】

エントロピ復号を待たずに、デコーダにおいて異なるタイルを並列に処理できるように、異なるタイルを識別する能力を高める別の方式として、各タイルの開始に関連付けられた、ビットストリーム内のマーカーを使用してよい。これらのタイル・マーカーは、ビットストリームのその特定の部分のエントロピ復号を待たないで、識別されるように、ビットストリーム内に含まれる。一実施形態では、マーカーは、 $0 \times 000001$ で始まり、別の実施形態では、マーカーは、 $0 \times 000002$ で始まり、別の実施形態では、マーカーは、 $0 \times 000004$ または任意の他の適切なビットのシーケンスで始まる。更にマーカーは、タイル及び/またはタイルの最初のマクロブロックに関連付けられた追加のヘッダーを含むことができる。この方法で、結果としてビット・レートは増加するが、エンコーダは、全てのタイルが符号化されるのを待たずに、符号化後に各タイルをビットストリームに書き込むことができる。更に、バッファリングと併せて使用される場合は特に、デコーダは、より効率的な方法でビットストリームを構文解析し、異なるタイルを識別できる。

10

#### 【0055】

タイル・ヘッダーは、スライス・ヘッダーと同様であるが、典型的には、より少ない情報が含まれている。必要となる主要な情報は、次のブロックのマクロブロック番号とエントロピ初期化データとスライス指標（タイル内の最初の符号化単位（CU: coding unit）が属するスライスを示す）である。そのようなタイル・ヘッダーの符号化シンタックスは、図12Aに示したようなものでもよい。あるいは、主要な情報には、初期量子化パラメータも含めることができる。そのようなタイル・ヘッダーの符号化シンタックスは、図12Bに示したようなものでもよい。スライス・ヘッダーで送信されず、且つタイル・ヘッダーにない値は、スライス・ヘッダーに送信された値にリセットすることができる。

20

#### 【0056】

幾つかの実施形態では、マーカーは、ビットストリームに含まれ、タイルの開始部に関連付けられる。しかしながら、マーカーは、ビットストリーム内のすべてのタイルに含まれていなくてもよい。これは、エンコーダ及びデコーダが異なるレベルの並列処理で動作することを容易にする。例えば、エンコーダは、ビットストリームには4つのマーカーしか含まなくても、64個のタイルを使用できる。これにより、64の処理を並列に符号化することが可能になり、4つの処理を並列に復号することが可能になる。幾つかの実施形態では、ビットストリーム内のマーカーの数は、エンコーダとデコーダの両方に知られている方法で規定される。例えば、マーカーの数は、ビットストリーム内で通知することも、プロファイルまたはレベルで定義することもできる。

30

#### 【0057】

幾つかの実施形態では、位置データがビットストリームの中にもめられ、タイルの開始部に関連付けられている。しかしながら、位置データは、ビットストリーム内のすべてのタイルに含まれていなくてもよい。これは、エンコーダ及びデコーダが異なるレベルの並列処理で動作することを容易にする。例えば、エンコーダは、ビットストリームには4つの位置しか含まなくても、64個のタイルを使用できる。これは、64の処理を並列に符号化することを可能にし、4つの処理を並列に復号することを可能にする。幾つかの実施形態では、ビットストリーム内の位置の数は、エンコーダ及びデコーダの両方に知られている方法で指定される。例えば、位置の数は、ビットストリーム内で通知することも、プロファイルまたはレベルで定義することもできる。

40

#### 【0058】

50

図13を参照して、タイルの開始を明示的に識別する1つの例示的方式を示す。タイル指標の値が初期化され ( $\text{TileIdx} = 0$ )、前のタイル位置の値が初期化され ( $\text{PreviousLocation} = 0$ )、前のタイル・サイズの値が初期化され ( $\text{PreviousTileSize} = 0$ )、タイルの数の値が初期化される ( $\text{TileCount} = (\text{num\_columns\_minus}1 + 1) * (\text{num\_rows\_minus}1 + 1)$ ) (300)。タイル・データは、タイルの最初の最大符号化単位 (LCU) のビットストリームに書き込まれ (310)、それは、典型的には、ピクチャの先頭において、左上側の角のLCUである。タイル・データは、タイル内の画素値を再構築するために必要なデータである。書き込まれるべき別のタイルがある場合 (320)、値のセットを修正することができる。タイルの内の最大符号化単位の数を示す現在のタイル・サイズを更新することができる ( $\text{CurrentTileSize} = \text{CurrentLocation} - \text{PreviousLocation}$ ) (330)。ここで、 $\text{CurrentLocation}$  は、ビットストリーム内の現在のタイルの開始の位置である。別の実施形態では、現在のタイル・サイズは、タイル内の符号化単位の数を表すように更新することができる。現在のタイル・サイズと前のタイル・サイズの間のおffsetを表すタイル差分offset (340) は、更新することができる。前の位置は、現在の位置に設定することができる ( $\text{PreviousLocation} = \text{CurrentLocation}$ )、前のタイル・サイズは、現在のタイル・サイズに設定することができる ( $\text{PreviousTileSize} = \text{CurrentTileSize}$ )、タイル指標は、インクリメントすることができる ( $\text{TileIdx} = \text{TileIdx} + 1$ ) (350)。次のタイルの最初のLCUは、データと共に書き込むことができる (310)。この再帰的な処理が、スライス内に書き込まれるべき追加のタイルがなくなる (335) まで、繰り返される。

#### 【0059】

タイル数が1より大きい場合 ( $\text{TileCount} > 1$ ) (360)、タイル位置情報フラグの存在が設定される ( $\text{TileLocationInformationPresentFlag} = \text{.T.}$ ) (370)。更に、例えば、 $\text{TileLocationInformationPresentFlag}$ 、 $\text{TileDifferentialOffset}[0]$ 、 $\text{TileDifferentialOffset}[1]$ 、・・・、 $\text{TileDifferentialOffset}[\text{NumberOfTilesMinus}1]$ を含む値のセットが書き込まれてもよい (380)。ここに、 $\text{NumberOfTilesMinus}1$  は、 $\text{TileCount} - 1$  に等しい。幾つかの実施形態では、値のセットは、スライス・ヘッダーに書き込まれる (380)。タイルが1つだけ存在する場合 (360)、タイルに関係する追加のデータの格納はスキップされる (385)。幾つかの実施形態では、 $\text{TileDifferentialOffset}$  は、現在のスライス内のタイル・サイズを表す。

#### 【0060】

図14を参照して、タイルの開始を黙示的に識別する例示的な方式を示す。タイル指標の値が初期化され ( $\text{TileIdx} = 0$ )、前のタイル位置が初期化され ( $\text{PreviousLocation} = 0$ )、前のタイル・サイズが初期化される ( $\text{PreviousTileSize} = 0$ ) (400)。タイル・データは、タイルの最初の最大符号化単位 (LCU) のビットストリームに書き込まれ (410)、それは、典型的には、ピクチャの先頭において、左上側の角のLCUである。書き込まれるべき別のタイルが存在する場合 (420)、値のセットがタイルの符号化単位に書き込まれる (425)。書き込まれる値 (425) は、マーカー及び/またはタイル・ヘッダー情報を含んでもよく、タイル・ヘッダーは、 $\text{lightweight\_slice\_flag}$  または等価なものを有するスライス・ヘッダーであってもよい。タイル・ヘッダー情報は、例えば、最大符号化単位アドレスと、スライス指標と、量子化パラメータ情報とを含んでもよい。次のタイルの最初のLCUは、データと共に書き込まれる (410)。この再帰的な処理が、スライス内に書き込まれるべき追加のタイルがなくなる (430) まで、繰り返される。タイル開始

位置の黙示的及び/または明示的な通知は、ビットストリームの中に含まれていると理解されたい。

【0061】

図15を参照して、タイル・ヘッダーは、LCUアドレス、スライス指標及び/または量子化パラメータ情報に加えて、1つ以上のタイル指標値を含むことができる。LCUアドレスは、タイル指標値TileIdxで置き換えられる場合がある。スライス指標は、ビットストリーム内で前に送信されたスライス・パラメータ・セットを指すスライス・パラメータ・セット指標で置き換えられる場合がある。スライス・パラメータ・セットは、量子化パラメータ、エントロピ・コード初期化パラメータ、参照リスト・パラメータ、またはループ・フィルタ・パラメータに対する値を含むことができるが、それに限定されない。タイル指標値は、例えば、タイル指標マイナス1のような、任意の適切な値とすることができる。この方法で、最初のタイルに番号0を付け、順次タイルに番号を付ける。タイル指標値を含むことにより、タイルの復号は、計算上の複雑性がより低下する。本発明の幾つかの実施形態では、タイル指標値は、ビットストリームまたはスライスの中の最初のタイルに対応しないタイル・ヘッダーにのみ存在する。

10

【0062】

更に、特定のスライスに対して、スライス・ヘッダー内の現在の情報またはスライス・ヘッダー内の前に送信された情報のどちらを使うかを表すフラグがタイル・ヘッダーに含まれていてもよい。これは、システムの計算上の複雑性を更に低減すると共にビットストリームのビット・レートも低減する。

20

【0063】

本発明の1つの実施形態では、フラグの存在は、ビットストリーム内においてslice\_parameter\_set\_present\_flagで表されてもよい。slice\_parameter\_set\_present\_flagのシンタックスは、次のようであってもよい。

【0064】

【表1】

Pic_parameter_set_rbsp() {	Descriptor
pic_parameter_set_id	ue(v)
...	
slice_parameter_set_present_flag	u(1)
rbsp_trailing_bits0	
}	

30

【0065】

slice\_parameter\_set\_present\_flagは、スライス・パラメータ・セットが送信されているかをビットストリーム内で通知する。1つの例示的シンタックスは、ピクチャ・パラメータ・セットに対してである。スライス・パラメータ・セット指標は、slice\_parameter\_set\_present\_flagが、スライス・パラメータ・セットがビットストリーム内で送信されていることを表す場合にのみ、ビットストリーム内で送信される。「slice\_parameter\_set\_present\_flag」が存在し、且つTRUE値に設定されている場合、slice\_parameter\_set\_present\_flagが通知される。「slice\_parameter\_set\_present\_flag」が存在し、且つFALSE値に設定されている場合、slice\_parameter\_set\_present\_flagが通知されない。幾つかの実施形態では、「slice\_parameter\_set\_present\_flag」がFALSE値に設定されている時、前に送信されたスライス・ヘッダーからの

40

50

情報が使用される。幾つかの実施形態では、TRUE値は1に等しい場合がある。幾つかの実施形態では、FALSE値は0に等しい場合がある。以下に説明するように、この処理は、フラグ `slice__parameter__set__flag` によって通知されたときに、幾つかのタイル・ヘッダーに対して、スキップしてもよい。本発明の幾つかの実施形態では、フラグ `slice__parameter__set__flag` は、真値であると推測される。

【0066】

図16を参照して、タイル・ヘッダーは、スライス・パラメータ・セット指標値を含んでもよい。スライス・パラメータ・セット指標値は、前に送信されたスライス・ヘッダー内の情報である、スライス・パラメータ・セットの中の1つ以上のセットを指してもよい。スライス・パラメータ・セットの各々は、例えば、量子化パラメータ(QP)、コンテキスト・ベースの適応型2値算術符号化情報(`cabac__init__idc`)、参照ピクチャ・リスト修正、またはループ・フィルタ情報を含んでもよい。ループ・フィルタ情報は、典型的に、例えば、適応型ループ・フィルタ情報、サンプル適応型オフセット情報、デブロッキング・フィルタ情報を含む。

10

【0067】

例示的なタイル・ヘッダーのシンタックスは、次のようである。

【0068】

【表2】

tile_header0{	Descriptor
tile_idx-1	u(v)
If(slice_parameter_set_present_flag)	
{	
slice_parameter_set_flag	u(1)
if(slice_parameter_set_flag)	
slice_parameter_set_id	ue(v)
}	
...	

20

30

【0069】

`slice__parameter__set__id` は、スライス・パラメータ・セット指標値を表し、前に送信されたビットストリーム内の特定のスライス・パラメータ・セットを指す。図16は、2つの異なるタイル・ヘッダーが、2つの異なる `slice__parameter__set__id` をどのように含み、前に送信されたビットストリーム内の2つの異なるスライス・パラメータ・セットをどのように参照するか、を示す。

【0070】

`tile__idx-1` は、タイル指標マイナス1を指定することができる。`tile__idx-1` は、符号なし数とすることができる。`tile__idx-1` は、 $v$  ビットを使うことができ、 $v$  はピクチャ内のタイルの最大数に基づいて決められる。例えば、 $v$  は、 $\text{RoundAbove}(\log_2(\text{TileCount}-1))$  として計算することができる。ここで、 $\text{RoundAbove}(x)$  は、 $x$  に等しいかまたは  $x$  より大きい最初の整数値に丸めることを表す。

40

【0071】

前に述べたように、`slice__parameter__set__flag` は、`slice__parameter__set__id` がビットストリーム内で送信されているかどうかを指定する。`slice__parameter__set__flag` が第1の値である場合、`slice__parameter__set__id` は、ビットストリームで送信されておらず、既存のアクティブなスライス・パラメータ・セットがタイルに対して使用される。

50

1つの実施形態では、アクティブなスライス・パラメータ・セットは、前のタイルで使われたスライス・パラメータ・セットである。第2の実施形態では、アクティブなスライス・パラメータ・セットは、前のタイルの中の最後の符号化単位によって使用されたスライス・パラメータ・セットである。slice\_parameter\_set\_flagが、第2の値の場合、slice\_parameter\_set\_idがビットストリーム内で送信される。1つの実施形態では、第1と第2の値は、それぞれ、0及び1である。

【0072】

slice\_parameter\_set\_idは、タイルによって参照されるスライス・パラメータ・セット識別子を指定する。

【0073】

図17を参照して、フラグを、タイルの終端を示すためにタイルの終端に含めてもよい。この方法で、タイルの終端を決定することに関する計算上の複雑さを簡単にできる。

【0074】

幾つかのケースで、本記載は、スライス及び/またはタイルに適したデータを有するビットストリームを生成する方式を例示した。この情報が適切な方法でビットストリーム内に含まれていることによって、デコーダは、このデータを、同様に受信し、ビットストリームをより効果的に復号するために使用できる。

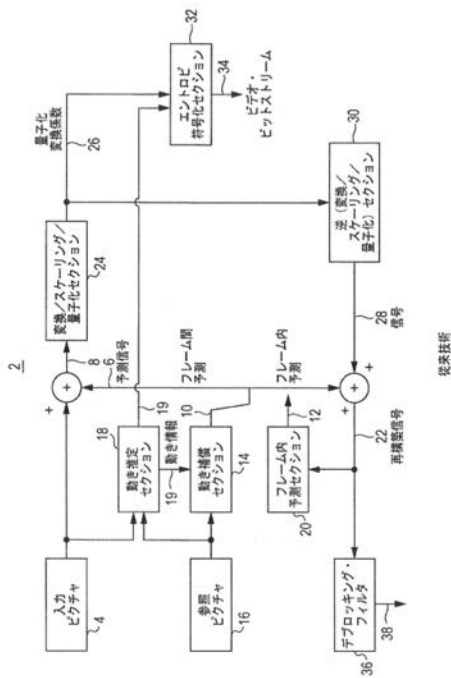
【0075】

前述の明細書において用いた用語および表現は、限定ではなく説明のための用語としてその中で使用されており、そのような用語および表現の使用において、図示及び説明した特徴またはその一部の等価物を排除する意図はなく、本発明の範囲は以下の特許請求の範囲によってのみ規定され限定されることが分かる。

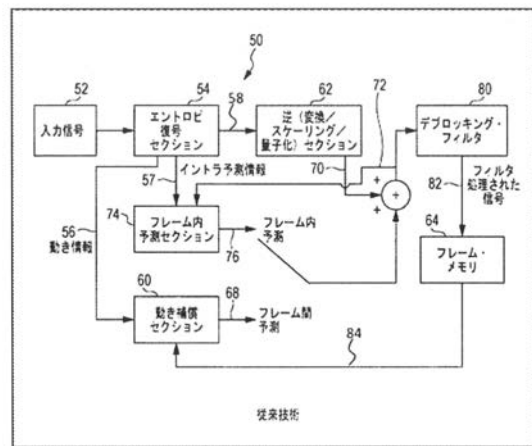
10

20

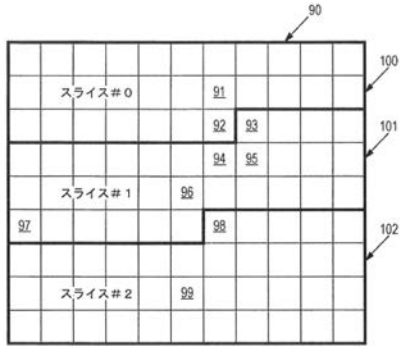
【図1】



【図2】

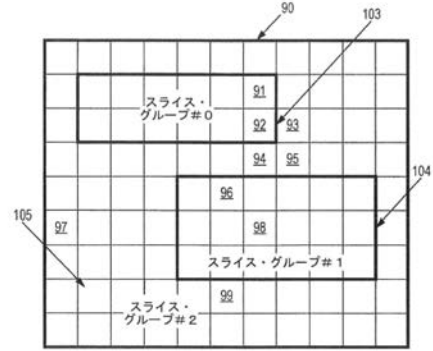


【 図 3 】



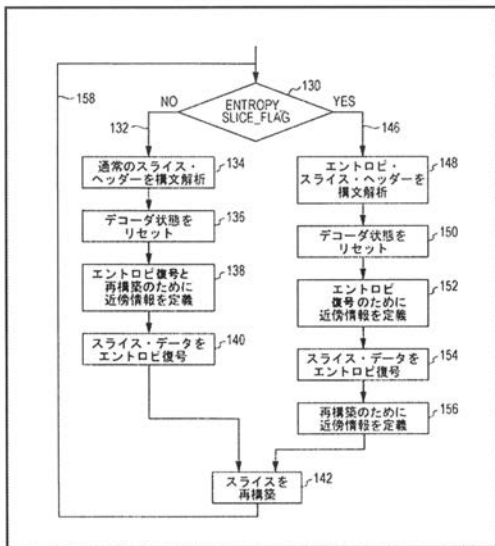
従来技術

【 図 4 】

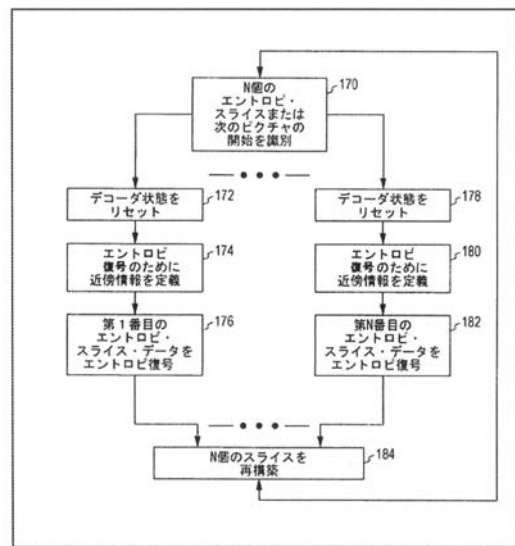


従来技術

【 図 5 】



【 図 6 】



【 図 7 】

1	2	3	4	5	16	17	18	19	20	21	34	35	36
6	7	8	9	10	22	23	24	25	26	27	37	38	39
11	12	13	14	15	28	29	30	31	32	33	40	41	42
43	44	45	46	47	63	64	65	66	67	68	87	88	89
48	49	50	51	52	69	70	71	72	73	74	90	91	92
53	54	55	56	57	75	76	77	78	79	80	93	94	95
58	59	60	61	62	81	82	83	84	85	86	96	97	98
99	100	101	102	103	109	110	111	112	113	114	121	122	123
104	105	106	107	108	115	116	117	118	119	120	124	125	126

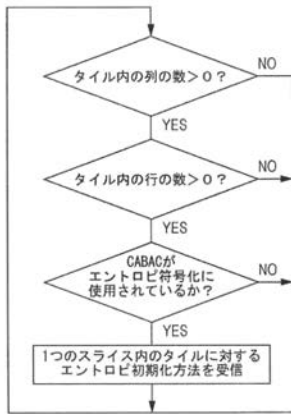
列の境界 (left side), 行の境界 (top)

【 図 8 】

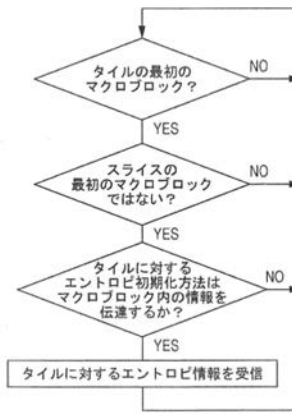
1	2	3	4	43	100	101	102
5	6	7	8	42	103	104	105
9	10	11	12	41	106	107	108
13	14	15	16	40	109	110	111
17	18	19	20	39	112	113	114
21	22	23	24	38	115	116	117
25	26	27	28	37	118	119	120
29	30	31	32	36	121	122	123
33	34	35	36	35	124	125	126

スライス1, スライス2, スライス3 (bottom), 列の境界 (left side), スライス (right side)

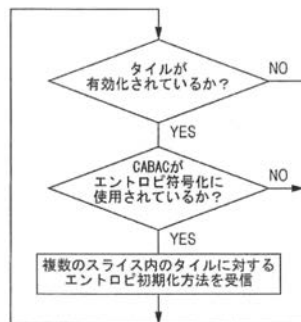
【 図 9 A 】



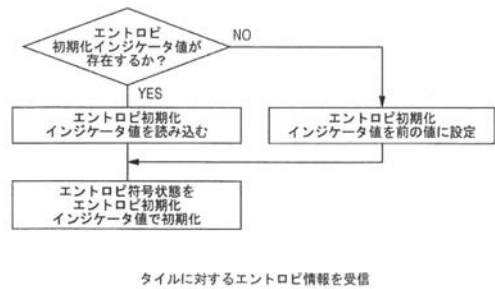
【 図 10 A 】



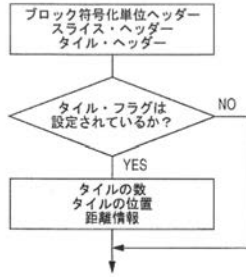
【 図 9 B 】



【 図 10 B 】



【 図 1 1 】



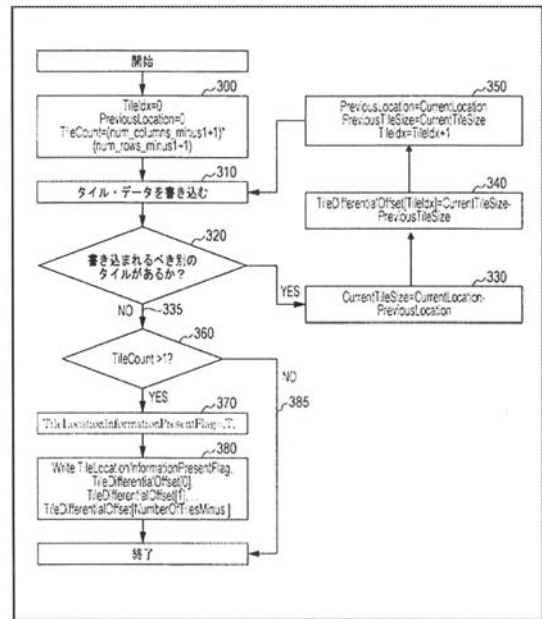
【 図 1 2 A 】

	C	Descriptor
slice header() {		
first lsb in slice	2	ue(v)
lightweight slice flag	2	u(1)
if( lightweight slice flag ) {		
slice_type	2	ue(v)
pic parameter set id	2	ue(v)
Frame num	2	u(v)
if( ldrPicFlag )		
idr pic id	2	ue(v)
pic order cnt lsb	2	u(v)
if( slice_type == P    slice_type == B ) {		
Num ref idx active override flag	2	u(1)
if( num ref idx active override flag ) {		
num ref idx 10 active minus1	2	ue(v)
if( slice_type == B )		
num ref idx 11 active minus1	2	ue(v)
}		
}		
}		
if( nal_ref_idc != 0 )		
Dec ref pic marking()	2	
if( entropy_coding_mode_flag && slice_type != 1 )		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
alf_param()		
if( slice_type == P    slice_type == B ) {		
mc_interpolation_idc	2	ue(v)
mv_competition_flag	2	u(1)
if( mv_competition_flag ) {		
mv_competition_temporal_flag	2	u(1)
}		
}		
if( slice_type == B && mv_competition_flag )		
collocated_from_10_flag	2	u(1)
}		
else {		
if( entropy_coding_mode_flag && slice_type != 1 )		
cabac_init_idc	2	ue(v)
}		
}		

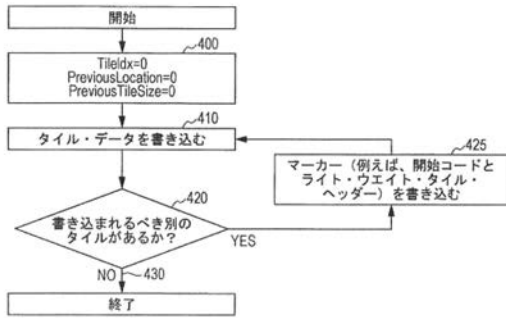
【 図 1 2 B 】

	C	Descriptor
slice header() {		
first lsb in slice	2	ue(v)
lightweight slice flag	2	u(1)
if( lightweight slice flag ) {		
slice_type	2	ue(v)
pic parameter set id	2	ue(v)
Frame num	2	u(v)
if( ldrPicFlag )		
idr pic id	2	ue(v)
pic order cnt lsb	2	u(v)
if( slice_type == P    slice_type == B ) {		
Num ref idx active override flag	2	u(1)
if( num ref idx active override flag ) {		
num ref idx 10 active minus1	2	ue(v)
if( slice_type == B )		
num ref idx 11 active minus1	2	ue(v)
}		
}		
}		
if( nal_ref_idc != 0 )		
Dec ref pic marking()	2	
if( entropy_coding_mode_flag && slice_type != 1 )		
cabac_init_idc	2	ue(v)
slice_qp_delta	2	se(v)
alf_param()		
if( slice_type == P    slice_type == B ) {		
mc_interpolation_idc	2	ue(v)
mv_competition_flag	2	u(1)
if( mv_competition_flag ) {		
mv_competition_temporal_flag	2	u(1)
}		
}		
if( slice_type == B && mv_competition_flag )		
collocated_from_10_flag	2	u(1)
}		
else {		
if( entropy_coding_mode_flag && slice_type != 1 )		
cabac_init_idc	2	ue(v)
slice_qp_delta		
}		
}		

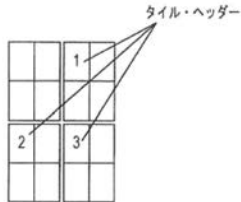
【 図 1 3 】



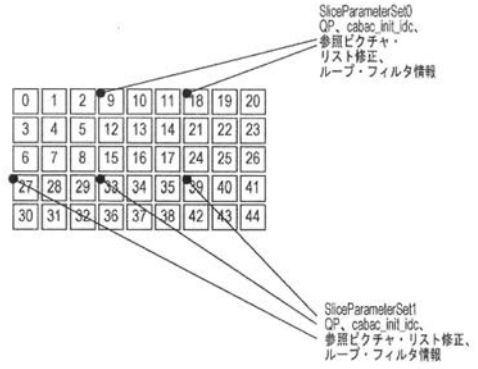
【 図 1 4 】



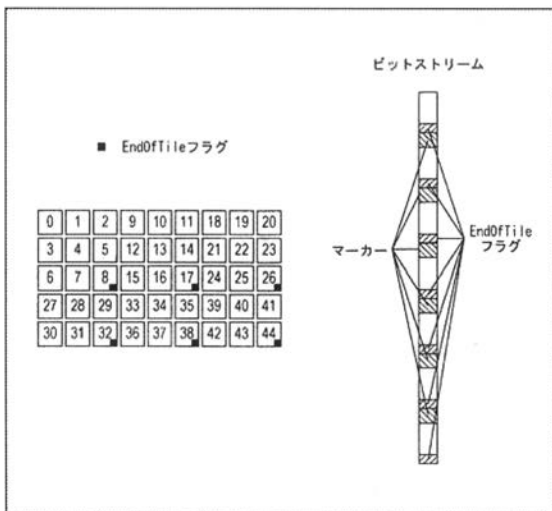
【 図 1 5 】



【 図 1 6 】



【 図 1 7 】



---

フロントページの続き

(72)発明者 ミスラ キラン

アメリカ合衆国 ワシントン州 98607, カマス, ノースウェスト パシフィック リム ブ  
ールバード 5750 シャープ ラボラトリーズ オブ アメリカ インコーポレイテッド内

(72)発明者 セガール クリストファー エー.

アメリカ合衆国 ワシントン州 98607, カマス, ノースウェスト パシフィック リム ブ  
ールバード 5750 シャープ ラボラトリーズ オブ アメリカ インコーポレイテッド内

Fターム(参考) 5C159 KK08 KK13 MA00 ME01 RB09 RC07 RC11 TB06 UA02 UA05

(54)【発明の名称】ビデオを復号化する方法、ビデオを符号化する方法、デコーダ、エンコーダ、復号プログラムを記録したコンピュータ読み取り可能な記録媒体、および符号化プログラムを記録したコンピュータ読み取り可能な記録媒体