(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7093420号
(P7093420)

(45)発行日　令和4年6月29日(2022.6.29)

(24)登録日　令和4年6月21日(2022.6.21)

(51)国際特許分類
H04N 19/70 (2014.01)

FI
H04N 19/70

請求項の数　10　（全42頁）

(21)出願番号　特願2020-551918(P2020-551918)
(86)(22)出願日　平成31年3月25日(2019.3.25)
(65)公表番号　特表2021-519019(P2021-519019A)
(43)公表日　令和3年8月5日(2021.8.5)
(86)国際出願番号　PCT/EP2019/057432
(87)国際公開番号　WO2019/197145
(87)国際公開日　令和1年10月17日(2019.10.17)
審査請求日　令和2年11月20日(2020.11.20)
(31)優先権主張番号　1805886.7
(32)優先日　平成30年4月9日(2018.4.9)
(33)優先権主張国・地域又は機関
英国(GB)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　110003281
特許業務法人大塚国際特許事務所
(72)発明者　タケ， ジョナサン
フランス国 レンヌ－アタラント， セデックス セッソン－セヴィニエ ３５５１７， リュ ドゥ ラ トゥッシュランベール キヤノン リサーチ センター フランス エス，エー，エス 内
(72)発明者　ウエドラオゴ， ナエル
フランス国 レンヌ－アタラント， セデックス セッソン－セヴィニエ ３５５１７， リュ ドゥ ラ トゥッシュランベー
最終頁に続く

(54)【発明の名称】　フレーム部分を有するビデオデータを符号化又は復号する方法及び装置

(57)【特許請求の範囲】
【請求項１】
フレームを符号化してビットストリームを生成する方法であって、前記フレームはフレーム部分に空間的に分割され、前記方法は、
前記フレームに含まれるフレーム部分を符号化する工程と、
前記フレームに含まれるフレーム部分の各々の識別子をシグナリングする工程と、
前記フレームに含まれるフレーム部分の位置に関する空間情報をシグナリングする工程と、
を有し、
前記識別子および前記空間情報は、前記ビットストリームの所定のパラメータセットにシグナリングされ、
前記識別子を表すために用いられるビット数の情報は、前記ビットストリームにシグナリングされる
ことを特徴とする方法。
【請求項２】
前記フレーム部分は、独立して符号化されることを特徴とする請求項１に記載の方法。
【請求項３】
前記フレーム部分が独立して符号化されたことを示すフラグを提供することをさらに含むことを特徴とする請求項２に記載の方法。
【請求項４】
ビットストリームから、フレームを含むビデオデータを復号する方法であって、前記フレ

ームはフレーム部分に空間的に分割され、前記方法は、
前記ビットストリームの所定のパラメータセットから、前記フレームに含まれるフレーム部分の各々の識別子を取得する工程と、
前記ビットストリームの前記所定のパラメータセットから、前記フレームに含まれるフレーム部分の位置に関する空間情報を取得する工程と、
前記空間情報に基づいて前記フレーム内のフレーム部分の位置を決定する工程と、
前記フレーム部分を復号する工程と
を有し、
前記識別子を表すために用いられるビット数の情報は、前記ビットストリームにシグナリングされている
ことを特徴とする方法。
【請求項５】
前記フレーム部分は、独立して符号化されることを特徴とする請求項４に記載の方法。
【請求項６】
前記フレーム部分が独立して符号化されたことを示すフラグを取得することをさらに含むことを特徴とする請求項５に記載の方法。
【請求項７】
フレームを符号化してビットストリームを生成する符号化装置であって、前記フレームはフレーム部分に空間的に分割され、前記符号化装置は、
前記フレームに含まれるフレーム部分を符号化する手段と、
前記フレームに含まれるフレーム部分の各々の識別子をシグナリングする手段と、
前記フレームに含まれるフレーム部分の位置に関する空間情報をシグナリングする手段と、
を有し、
前記識別子および前記空間情報は、前記ビットストリームの所定のパラメータセットにシグナリングされ、
前記識別子を表すために用いられるビット数の情報は、前記ビットストリームにシグナリングされる
ことを特徴とする符号化装置。
【請求項８】
ビットストリームから、フレームを含むビデオデータを復号する復号装置であって、前記フレームはフレーム部分に空間的に分割され、前記復号装置は、
前記ビットストリームの所定のパラメータセットから、前記フレームに含まれるフレーム部分の各々の識別子を取得する手段と、
前記ビットストリームの前記所定のパラメータセットから、前記フレームに含まれるフレーム部分の位置に関する空間情報を取得する手段と、
前記空間情報に基づいて前記フレーム内のフレーム部分の位置を決定する手段と、
前記フレーム部分を復号する手段と
を有し、
前記識別子を表すために用いられるビット数の情報は、前記ビットストリームにシグナリングされている
ことを特徴とする復号装置。
【請求項９】
コンピュータに請求項１～６のいずれか１項に記載の方法を実行させるためのコンピュータプログラムを記憶する記憶媒体。
【請求項１０】
コンピュータに、請求項１～６のいずれか１項に記載の方法を実行させるためのコンピュータプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】

本開示は、空間部分を有するビデオデータを符号化または復号する方法および装置に関する。

【背景技術】

【０００２】

並列符号化のために導入され設計されたＨＥＶＣタイル。しかしながら、高サイズのビデオコンテンツでは、タイルが異なって使用される幾つかのユースケースがある。特に、個々のタイル又はタイルのセットをストリーミングする必要性が高まってきている。いくつかのアプリケーションはまた、新しいビデオシーケンスを構成するために、同じシーケンス又は異なるシーケンスから異なるタイルを組み合わせる必要性を高めた。

【０００３】

ＨＥＶＣにおける現在のメカニズムは、この種のシナリオを念頭に置いて設計されていない。現在のＨＥＶＣメカニズムを用いてこれらのシナリオを実施することは、タイルに対するエンコード制約を追加することを意味し、復号時のタイルの任意の構成は、データの書き換え処理を含む。特に、スライスセグメントヘッダの操作を含むデータの書き換えが一般的に要求される。

【発明の概要】

【０００４】

本発明は、前述の問題のうちの１つまたは複数に対処するように考案された。それは、フレーム部分の定義と、ビットストリームにおけるこれらのフレーム部分のシグナリングと、に関する。本発明は、復号時にこれらのフレーム部分の抽出および再結合を容易にする一方で、そうするときに必要な書き換え処理を制限することを目的としている。

【０００５】

本発明の第１の態様によれば、フレームを含むビデオデータをビットストリームに符号化する方法が提供され、フレームはフレーム部分に空間的に分割され、方法は、

－少なくとも１つのフレーム部分を１つまたは複数の第１の符号化ユニットに符号化することを有し、

前記方法はさらに、

－前記第１の符号化ユニットへ、少なくとも１つのフレーム部分識別子をシグナリングすることと、フレーム部分識別子は１つの符号化フレーム部分を識別し、

－フレーム部分識別子およびフレーム部分に関する空間情報を含むフレーム部分構成情報を提供すること

を有する。

【０００６】

本発明の第１の態様は、ＨＥＶＣタイルのような既知のタイリングデザインと比較して圧縮を改善する可能性を可能にしながら、より柔軟性があり、より簡単な操作を提供するという利点を有する。

【０００７】

一実施形態では、フレーム部分構成情報が１つの第２の符号化ユニットに提供される。

【０００８】

一実施形態では、少なくとも１つのフレーム部分が独立して符号化される。

【０００９】

一実施形態では、この方法がさらに、フレーム部分が独立して符号化されたことを示すフラグを提供することを含む。

【００１０】

一実施形態では、１つまたは複数の第１の符号化ユニットが、フレーム部分が独立して符号化されたことを各フレーム部分について示すフラグを有する。

【００１１】

一実施形態では、１つまたは複数の第１の符号化ユニットが、少なくとも１つのフレーム部分が独立して符号化されたことを示すフラグを有する。

【００１２】

一実施形態では、１つまたは複数の第１の符号化ユニットが、フレーム部分を符号化するために使用される符号化制約のレベルを示すフラグを有する。

【００１３】

一実施形態では、フレーム部分はスライスであり、第１の符号化ユニットはデータ部分を含むスライスユニットであり、フラグはスライスユニットのデータ部分のスライスセグメントヘッダに含まれる。

【００１４】

一実施形態では、フレーム部分はスライスであり、第１の符号化ユニットはデータ部分を含むスライスユニットであり、フレーム部分識別子はスライスユニットのデータ部分のスライスセグメントヘッダに含まれる。

【００１５】

一実施形態では、第１の符号化ユニットがヘッダ部分と、符号化フレーム部分を含むデータ部分と、を含み、前記フレーム部分識別子はヘッダ部分に含まれる。

【００１６】

一実施形態では、フレーム部分識別子がすべてのフレーム部分符号化ユニットでシグナリングされ、事前定義されたフレーム部分識別子値はフレーム部分が独立して符号化されていないことを示す。

【００１７】

一実施形態では、第２の符号化ユニットが１つまたは複数のフレームに関する情報専用のパラメータセットである。

【００１８】

一実施形態では、第２の符号化ユニットがフレーム部分情報専用のパラメータセットである。

【００１９】

一実施形態では、第１の符号化ユニットが、フレーム部分が独立して符号化されたことを示す特定のタイプを有する。

【００２０】

一実施形態では、フレーム部分識別子が固定の所定数のビットを使用して符号化される。

【００２１】

一実施形態では、フレーム部分識別子がシグナリングされた数のビットを使用して符号化される。

【００２２】

一実施形態では、空間情報が、符号化ツリー単位アドレスによって与えられるフレーム部分の位置を含む。

【００２３】

一実施形態では、空間情報が、サンプルアドレスによって与えられるフレーム部分の位置を含む。

【００２４】

一実施形態では、空間情報が、フレーム部分のサイズを含む。

【００２５】

一実施形態では、フレーム部分の位置がフレームに対して与えられる。

【００２６】

一実施形態では、いくつかのパラメータデータユニットが、同じフレーム部分に対する異なるフレーム部分構成を含むビットストリームにおいてシグナリングされる。

【００２７】

一実施形態では、第２の符号化ユニットが、所与のポストフィルタリングアルゴリズムがフレーム部分に使用可能かどうかを示すフラグを有する。

【００２８】

一実施形態では、同じフレーム部分識別子を使用して、フレーム部分セットを定義するいくつかのフレーム部分を識別することができる。

【００２９】
一実施形態では、ヘッダ部分がレイヤ識別子を含み、レイヤ識別子はフレーム部分識別子をシグナリングするために使用される。
【００３０】
本発明の第２の態様によれば、少なくとも１つのビットストリームから、フレームを含むビデオデータを復号する方法が提供され、フレームはフレーム部分に空間的に分割され、方法は、
－ビットストリームから、フレーム部分識別子とフレーム部分に関する空間情報とを含むフレーム部分構成情報を取得することと、
－ビットストリーム内の１つまたは複数の第１の符号化ユニットから少なくともフレーム部分を抽出することと、フレーム部分はフレーム部分識別子を含み、
－空間情報に基づいてフレーム内のフレーム部分の位置を決定することと、
－決定された位置に従ってフレーム部分をフレームにレンダリングするためにフレーム部分を復号することと
を有する。
【００３１】
一実施形態では、フレーム部分構成情報が１つの第２の符号化ユニットに提供される。
【００３２】
一実施形態では、少なくとも１つのフレーム部分が独立して符号化される。
【００３３】
一実施形態では、この方法がさらに、フレーム部分が独立して符号化されたことを示すフラグを取得することを含む。
【００３４】
一実施形態では、１つまたは複数の第１の符号化ユニットが、フレーム部分が独立して符号化されたことを各フレーム部分について示すフラグを有する。
【００３５】
一実施形態では、１つまたは複数の第１の符号化ユニットが、少なくとも１つのフレーム部分が独立して符号化されたことを示すフラグを有する。
【００３６】
一実施形態では、１つまたは複数の第１の符号化ユニットが、フレーム部分を符号化するために使用される符号化制約のレベルを示すフラグを有する。
【００３７】
一実施形態では、フレーム部分はスライスであり、第１の符号化ユニットはデータ部分を含むスライスユニットであり、フラグはスライスユニットのデータ部分のスライスセグメントヘッダに含まれる。
【００３８】
一実施形態では、フレーム部分はスライスであり、第１の符号化ユニットはデータ部分を含むスライスユニットであり、フレーム部分識別子はスライスユニットのデータ部分のスライスセグメントヘッダに含まれる。
【００３９】
一実施形態では、第１の符号化ユニットがヘッダ部分と、符号化フレーム部分を含むデータ部分とを含み、前記フレーム部分識別子はヘッダ部分に含まれる。
【００４０】
一実施形態では、フレーム部分識別子がすべてのフレーム部分符号化ユニットでシグナリングされ、事前定義されたフレーム部分識別子値はフレーム部分が独立して符号化されていないことを示す。
【００４１】
一実施形態では、第２の符号化ユニットが１つまたは複数のフレームに関する情報専用のパラメータセットである。
【００４２】

一実施形態では、第２の符号化ユニットがフレーム部分情報専用のパラメータセットである。
【００４３】
一実施形態では、第１の符号化ユニットが、フレーム部分が独立して符号化されたことを示す特定のタイプを有する。
【００４４】
一実施形態では、フレーム部分識別子が固定の所定数のビットを使用して符号化される。
【００４５】
一実施形態では、フレーム部分識別子がシグナリングされたビット数を使用して符号化される。
【００４６】
一実施形態では、空間情報が符号化ツリー単位アドレスによって与えられるフレーム部分の位置を含む。
【００４７】
一実施形態では、空間情報がサンプルアドレスによって与えられるフレーム部分の位置を含む。
【００４８】
一実施形態では、空間情報がフレーム部分のサイズを含む。
【００４９】
一実施形態では、フレーム部分の位置がフレームに対して与えられる。
【００５０】
一実施形態では、いくつかのパラメータデータユニットが、同じフレーム部分に対する異なるフレーム部分構成を含むビットストリームから取得される。
【００５１】
一実施形態では、第２の符号化ユニットが、所与のポストフィルタリングアルゴリズムがフレーム部分に使用可能かどうかを示すフラグを有する。
【００５２】
一実施形態では、同じフレーム部分識別子を使用して、フレーム部分セットを定義するいくつかのフレーム部分を識別することができる。
【００５３】
一実施形態では、ヘッダ部分がレイヤ識別子を含み、レイヤ識別子はフレーム部分識別子をシグナリングするために使用される。
【００５４】
本発明の第３の態様によれば、フレームを含むビデオデータを含む新しいビットストリームを生成する方法が提供され、フレームはフレーム部分に空間的に分割され、方法は、
－複数のビットストリームから抽出され且つ新しいビットストリームにマージされる複数のフレーム部分を決定することと、複数のビットストリームは、請求項１～２５のいずれか１項に従って符号化され、
－抽出するフレーム部分のフレーム部分識別子を決定することと、
－新しいビットストリームに対するフレーム部分構成情報を生成することと、
－複数のビットストリームから抽出すべき複数のフレーム部分を抽出することと、
－複数のフレーム部分および生成されたフレーム部分構成情報を、新しいビットストリームに埋め込むことと
を有する。
【００５５】
一実施形態では、方法がさらに、
－抽出されたフレーム部分に対して新しいフレーム部分識別子を決定することと、
－フレーム部分識別子を、新しいフレーム部分識別子によって、抽出されたフレーム部分に置き換えることと
を有する。

【００５６】
一実施形態では、複数のフレーム部分を抽出することが、
－複数のビットストリームを構文解析することと、
－決定されたフレーム部分識別子のうちの１つを含むフレーム部分符号化データユニットを抽出する。
【００５７】
本発明の第４の態様によれば、フレームを含むビデオデータをビットストリームに符号化する装置が提供され、フレームはフレーム部分に空間的に分割され、装置は、
－少なくとも１つのフレーム部分を、１つまたは複数の第１の符号化ユニットに符号化するように構成された回路を有し、
方法はさらに、
－前記第１の符号化ユニットへ、少なくとも１つのフレーム部分識別子をシグナリングすることと、フレーム部分識別子は１つの符号化フレーム部分を識別し、
－フレーム部分識別子およびフレーム部分に関する空間情報を含むフレーム部分構成情報を提供すること
を有する。
【００５８】
本発明の第５の態様によれば、少なくとも１つのビットストリームから、フレームを含むビデオデータを復号する装置が提供され、フレームはフレーム部分に空間的に分割され、装置は、
－ビットストリームから、フレーム部分識別子とフレーム部分に関する空間情報とを含むフレーム部分構成情報を取得し、
－ビットストリーム内の１つまたは複数の第１の符号化ユニットから少なくともフレーム部分を抽出し、フレーム部分はフレーム部分識別子を含み、
－空間情報に基づいてフレーム内のフレーム部分の位置を決定し、
－決定された位置に従ってフレーム部分をフレームにレンダリングするためにフレーム部分を復号する
ように構成された回路を有する。
【００５９】
本発明の第６の態様によれば、フレームを含むビデオデータを含む新しいビットストリームを生成する装置が提供され、フレームはフレーム部分に空間的に分割され、装置は、
－複数のビットストリームから抽出され且つ新しいビットストリームにマージされる複数のフレーム部分を決定し、複数のビットストリームは、請求項１～２５のいずれか１項に従って符号化され、
－抽出するフレーム部分のフレーム部分識別子を決定し、
－新しいビットストリームに対するフレーム部分構成情報を生成し、
－複数のビットストリームから抽出すべき複数のフレーム部分を抽出し、
－複数のフレーム部分および生成されたフレーム部分構成情報を、新しいビットストリームに埋め込む
回路を有する。
【００６０】
本発明の第７の態様によれば、プログラマブル装置のためのコンピュータプログラム製品が提供され、コンピュータプログラム製品は、プログラマブル装置にロードされて実行されるときに、本発明による方法を実施するための一連の命令を含む。
【００６１】
本発明の第８の態様によれば、本発明による方法を実施するためのコンピュータプログラムの命令を格納したコンピュータ可読記憶媒体が提供される。
【００６２】
本発明の第９の態様によれば、実行時に本発明による方法を実行させるコンピュータプログラムが提供される。

【００６３】
本発明による方法の少なくとも一部は、コンピュータで実施することができる。したがって、本発明は、全体的にハードウェアの実施形態、全体的にソフトウェアの実施形態（ファームウェア、常駐ソフトウェア、マイクロコードなどを含む）、または本明細書ではすべて一般に「回路」、「モジュール」、または「システム」と呼ばれることがあるソフトウェアおよびハードウェアの態様を組み合わせた実施形態の形態をとることができる。さらに、本発明は、媒体に具現化されたコンピュータ使用可能プログラムコードを有する表現の任意の有形の媒体に具現化されたコンピュータプログラム製品の形成をとることができる。
【００６４】
本発明はソフトウェアで実施することができるので、本発明は、任意の適切なキャリア媒体上のプログラマブル装置に提供するためのコンピュータ可読コードとして実施することができる。有形、非一時的キャリア媒体は、フロッピーディスク、ＣＤ－ＲＯＭ、ハードディスクドライブ、磁気テープ装置またはソリッドステートメモリ装置などの記憶媒体を含むことができる。過渡キャリア媒体は、電気信号、電子信号、光信号、音響信号、磁気信号、または電磁信号、例えばマイクロ波またはＲＦ信号等の信号を含むことができる。
【図面の簡単な説明】
【００６５】
ここで、本発明の実施形態を、単なる例として、以下の図面を参照して説明する。
【図１】図１は、本発明を統合することができるシステムを示す。
【図２】図２は、ブロックベースのビデオエンコーダ、例えばＨＥＶＣの画像符号化構造を示す。
【図３】図３は、ＨＥＶＣにおけるスライスセグメント及びタイルと呼ばれる２種類のパーティションによる画像の分割を示す。
【図４】図４は、画像の境界を横切るＣＴＵのためのＨＥＶＣで使用される４分木推論メカニズムを示す。
【図５】図５は、例えばＨＥＶＣで使用される境界拡張メカニズムを示す。
【図６】図６は、ＨＥＶＣビットストリーム構成の例を示す。
【図７】図７は、注目領域（ＲＯＩ）ストリーミングのためのＨＥＶＣ分割の例を示す。
【図８ａ】図８ａは、注目領域の組合せのための２つの異なる使用シナリオ例を示す。
【図８ｂ】図８ｂは、注目領域の組合せのための２つの異なる使用シナリオ例を示す。
【図９】図９は、本発明が統合されたビデオエンコーダの典型的な符号化処理を示す。
【図１０】図１０は、本発明が統合されたビデオデコーダの典型的な復号処理を示す。
【図１１】図１１は、本発明の例示的な使用を説明する。
【図１２】図１２は、カプセル化ステップの詳細を示す。
【図１３ａ】図１３ａは、符号化処理によって実行されるフレーム部分構成のシグナリングを示す。
【図１３ｂ】図１３ｂは、符号化処理によって実行されるフレーム部分構成のシグナリングを示す。
【図１３ｃ】図１３ｃは、符号化処理によって実行されるフレーム部分構成のシグナリングを示す。
【図１４】図１４は、非格子ベースのパーティショニングの例を示す。
【図１５】図１５は、ＣＴｉｌｅ識別子をシグナリングするための代替実施形態を示す。
【図１６ａ】図１６ａは、ＣＴｉｌｅごとの依存関係リストを含むＸＰＳを示す。
【図１６ｂ】図１６ｂは、ＣＴｉｌｅ依存関係の第１の例を示す。
【図１６ｃ】図１６ｃは、ＣＴｉｌｅ依存関係の第２の例を示す。
【図１７ａ】図１７ａは、ＣＴｉｌｅが連続的に符号化されたフレーム間で位置またはサイズを変更することができる実施形態の例を提供する。
【図１７ｂ】図１７ｂは、ＣＴｉｌｅが連続的に符号化されたフレーム間で位置またはサイズを変更することができる実施形態の例を提供する。

【図１８】図１８は、本発明の１つまたは複数の実施形態を実装するためのコンピューティングデバイスの概略ブロック図。
【発明を実施するための形態】
【００６６】
ビデオシーケンスのフレームを空間フレーム部分に符号化することは、例えば、いわゆる３６０度ビデオのストリーミングに関連するシナリオにおいて特に有用であり、これは実際には、３６０度パノラマビデオ又は球面ビデオの古典的な２Ｄビデオ表現への投影の結果である。
【００６７】
３６０度ビデオ（または３６０度ビデオのみ）は、良好なユーザ体験を提供するために非常に高い解像度を有することができるビデオである。ヘッドマウントディスプレイの内側（または画面上）に表示される場合、３６０ビデオコンテンツの空間サブパートのみがユーザに提示される。
【００６８】
したがって、ＨＴＴＰ（ＤＡＳＨ）上の動的アダプティブストリーミングのようなストリーミングプロトコルを利用することは興味深いことであり、例えば、ユーザが見ている領域に対してのみ、高品質の空間フレーム部分を要求することである。見られていない領域（すなわち、ユーザが見ていない領域）については、空間フレーム部分を単にスキップすることができる。
【００６９】
本発明のアプリケーションは、ストリーミングをユーザの視線方向に適用するストリーミングアプローチに言及する。言い換えると、ビューポート依存のストリーミングを指す。このようなアプローチでは、記憶コスト、計算コスト、およびユーザ体験の間の１つの良好な妥協点は、シーケンスを様々な品質を有する独立した空間フレーム部分に符号化することである。次いで、フレーム部分は、ニーズおよび帯域幅制約に従って、ランダムにアクセスされ、抽出され、および／または他のフレーム部分シーケンスと組み合わされ得る。追加の符号化やトランスコーディングは要求されない。このようなシナリオの例を、図８ａを参照して説明する。
【００７０】
アプリケーションは、いくつかの異なるビデオの空間フレーム部分がシステムオペレータから要求された構成に一致するように新しいビデオに再編成されるビデオ監視システムに関する。例えば、オペレータは、オリジナルビデオの一部のみを望む場合がある。この用途は、特に図８ｂに示されている。
【００７１】
最後に、別のアプリケーションでは、フルビデオシーケンスから抽出された単一の空間フレーム部分のみを含む新しい「ビデオ」が、新しいビデオにおける空間フレーム部分の新しい位置が異なる場合、符号化パラメータの書き換えを含むことができる。
【００７２】
ＨＥＶＣを使用する場合、空間フレーム部分の符号化は、ＨＥＶＣタイルに基づく。しかしながら、ＨＥＶＣタイル、より一般的にはＨＥＶＣタイプのタイルは、上述の用途に対処するようには設計されていない。
【００７３】
図１は、本発明を統合することができるシステム（例えば、インタラクティブストリーミングビデオシステム）の実施形態を示す。
【００７４】
ビデオビットストリームは、ネットワーク１０１を介して、サーバまたはプロキシサーバ１００からクライアント１０２に送信される。サーバ１００は、ブロックベースのビデオコーデック、例えばＨＥＶＣビデオコーデックの仕様に適合するビデオエンコーダ１０３によって生成されたビデオストリーム（またはビデオファイル）を使用する。
【００７５】

エンコーダは以下に説明するように、本発明に従っていくつかの空間フレーム部分への空間ランダムアクセスを提供しながら、異なるレート／歪みトレードオフを有するビデオシーケンスのセットを圧縮する。

【００７６】

サーバ１００は、通信ネットワーク１０１を介して、インタラクティブストリーミングのための利用可能なビデオストリームの記述の要求を受信する。通信ネットワーク１０１は、インターネットプロトコル標準に基づいている。ＩＰネットワーク１０１上でメディアプレゼンテーションを伝送するために採用される標準プロトコルは、好ましくはMPEG DASH: Dynamic Adaptive Streaming over HTTPである。しかしながら、本発明は、任意の他のストリーミングプロトコルにも使用することができる。

【００７７】

図２は、２種類のパーティション：スライスセグメントおよび空間フレーム部分による画像の分割を示す。画像２０６は、３つのスライスセグメントに分割されている。スライスセグメントは、画像の一部又は画像全体である。各スライスセグメントは、（ＨＥＶＣの符号化ユニットに対応することができる）整数個の符号化ブロックを含む。符号化ブロックはサンプルで構成される。

【００７８】

２種類のスライスセグメント：独立スライスセグメント２０７と従属スライスセグメント２０８。各スライスセグメントは、１つのＮＡＬユニットに埋め込まれ、これはパケット指向およびビットストリーム指向トランスポートシステムの両方で使用するための汎用フォーマットを有する構造である。２種類のスライスセグメント間の差異は、独立スライスセグメントヘッダにおいて指定されたデータがスライスセグメントの符号化ブロックを復号するのに必要なすべてのパラメータを定義するという事実にある。一方、従属スライスセグメントは縮小されたヘッダを有し、そのヘッダ内で利用可能でないパラメータを推論するために、最初の先行する独立スライスセグメントに依存する。スライス内の最初の符号化ユニットの宛先は、独立スライスセグメントヘッダに指定される。

【００７９】

図３は、空間フレーム部分（ＳＰＦ）への別の分割を示し、フレーム３０５に示されるように、各フレームを独立して符号化矩形領域に分割することを可能にする。

【００８０】

ＨＥＶＣタイプのタイルのように、空間フレーム部分は、整数個の符号化ブロックを含む。スライス境界と同様に、ＳＰＦ境界３１０は、すべてのイントラ予測メカニズムを破る。

【００８１】

ＨＥＶＣタイプのタイルのように、ＳＰＦは、復号処理を初期化するために使用される特定のＮＡＬユニットに含まれるピクチャパラメータセットで定義される。PPS NALユニットは、タイル行の数と、ピクチャ内のタイル列の数およびそれらの関連するサイズを指定できる構文要素を含む。その他のパラメータセットNALユニット(Video Parameter SetまたはＶＰＳ、Sequence Parameter SetsまたはＳＰＳなど）は、ビットストリームの符号化設定を記述するパラメータを伝える。本発明では、これらのパラメータセットの何れかはＸＰＳ（Ｘはワイルドカード文字として使用される）として参照される。１つのスライスセグメントにおけるＳＰＦの位置、例えば、ビットにおけるオフセット、は、スライスセグメントヘッダの末尾で使用可能な構文要素で識別される。

【００８２】

ＳＰＦおよびスライスセグメントは一緒に使用できるが、いくつかの制約がある。以下の条件のうち１つまたは両方を検証する必要がある。

－１つのスライス（またはスライスセグメント）のすべての符号化ブロックは同じＳＰＦに属する

－１つのＳＰＦのすべての符号化ブロックは同じスライス（またはスライスセグメント）に属する

【００８３】

１つのスライス（又はスライスセグメント）は、幾つかのＳＰＦ全体を含む、又は単一タイルのサブ部分のみであることを意味する。第２に、ＳＰＦは、いくつかのスライス全体（またはスライスセグメント）を含むことができる、または単一のスライス（またはスライスセグメント）のサブ部分のみであることができる。

【００８４】

図４は、画像の境界を横切る符号化ユニットについて、ＨＥＶＣで使用される四分木推論メカニズムを、説明のためだけに、概略的に示す。ＨＥＶＣでは、画像は、符号化ユニットのサイズの倍数の幅及び高さを有するようには限定されない。次いで、フレームの右端の符号化ユニットは、画像の右側の境界４０１を横切り、フレームの最下部の符号化ユニットは、画像の最下部の境界４０２を横切る。これらの場合、ＨＥＶＣは、境界を横切る符号化ユニットについて四分木推論メカニズムを定義する。このメカニズムは、境界を越えるＣＵがなくなるまで、あるいはこれらの符号化ユニットについて最大四分木深さに達するまで、画像境界を越えている符号化ユニットの任意のＣＵを再帰的に分割することにある。例えば、符号化ユニット４０３は自動的に分割されず、符号化ユニット４０４、４０５及び４０６は分割される。推論された四分木のシグナリングは存在せず：デコーダは、画像境界上の同じ四分木を推論しなければならない。しかしながら、自動的に取得された四分木は、例えば、４０７におけるように、（最大四分木深さに達しない場合に）その符号化ユニットについての分割情報をシグナリングすることによって、フレームの内側にある符号化ユニットについてさらに洗練されてもよい。

【００８５】

図６は、サーバからクライアントに送信される典型的なビデオビットストリーム６００を示す。ビットストリームは、ＨＥＶＣまたはブロックベースのビットストリームに準拠している。

【００８６】

ビットストリーム６００は、一連のネットワークアブストラクトレイヤ（ＮＡＬ）ユニットとして構造化される。ＮＡＬユニットにはいくつかの種類（タイプ）がある。パラメータセットＮＡＬユニット（例えば、ＨＥＶＣのためのＶＰＳ、ＳＰＳ、およびＰＰＳ）は、シーケンスを符号化するために使用される符号化ツールの設定を記述する。それらはまた、画像の特性（解像度、フレームレート等）に関するいくつかの情報を記述する。

【００８７】

第１のＮＡＬユニット６０１は、ビットストリーム全体の情報を提供するビデオパラメータセット（ＶＰＳ）である。特に、ビットストリーム内のスケーラビリティレイヤの数を示す。

【００８８】

続くＮＡＬユニット６０２は、シーケンスパラメータセット（ＳＰＳ）である。それは、シーケンスレベルパラメータを提供する。ピクチャレベルパラメータを提供するピクチャパラメータセット（ＰＰＳ）ＮＡＬユニット６０３が後に続く。次に、スライスセグメント６０４を提供することができる。フレーム当たり１つのスライスセグメントを有することが一般的である。スライスセグメント６０４は、様々なＮＡＬユニットタイプ（ＣＲＡ、ＩＤＲ、ＢＬＡ、ＲＡＳＬ、ＲＡＤＬ、ＳＴＳＡ、ＴＳＡ、またはＴＲＡＩＬ．．．）を有するＮＡＬユニットに含まれ得る。スライスセグメントを含むＮＡＬユニットは、ＮＡＬヘッダ６０５（ＮＡＬヘッダのさらなる説明は図１０の説明で提供される）と、生バイトシーケンスペイロード（ＲＢＳＰ）６０６とから構成される。ＮＡＬヘッダ６０５は、ＮＡＬユニットタイプを含む情報を含む。ＲＢＳＰ（すなわち、ＮＡＬユニットデータ）は、ＮＡＬユニットタイプに固有の情報を含む。スライスセグメントの場合、ＲＢＳＰは、スライスセグメントヘッダ６０７と、それに続くスライスセグメントデータ６０８とを含む。スライスセグメントデータは、スライスセグメントのラスタスキャン順序付き符号化ツリーユニット６０９の符号化データの連続である。

【００８９】

一実施形態（ここでは図示せず）では、（タイリングパラメータセットのための）ＴＰＳ

と呼ばれるパラメータセットを、スライスセグメントＮＡＬユニットの前にビットストリームに挿入することができる。対応するパラメータは、新しいＴＰＳが見つかるまで有効である。ＴＰＳは、フレームのパーティショニング構成を記述する。

【００９０】

別の実施形態では、ビットストリーム中にＴＰＳが存在しない場合、ビットストリーム中に１つの空間フレーム部分のみが存在すると仮定される。前記空間フレーム部分は、ビデオフレームと同じ寸法（dimensions）を有し、その原点に配置される。

【００９１】

ビットストリームは、独立したフレーム部分又は注目領域（ＲＯＩ）を含むことができる。図７は、ここではフレーム内の矩形領域と見なされる注目領域の例を概略的に示す。ＲＯＩは、ＨＥＶＣおよびブロックベースのコーデックにおいて周知である。

【００９２】

ＲＯＩまたは独立したフレーム部分をストリーミングすることは、パーティショニング戦略を意味する。これは、タイル境界の導入がいくつかのＨＥＶＣ予測メカニズムを破壊するので、符号化効率に影響を及ぼす。

【００９３】

図７において、フレーム７００は、４×４ＳＰＦ格子に分割される。予め定義されたＲＯＩ７０１にアクセスするために、ＳＰＦ６、７、１０、１１のための対応するスライスセグメントを埋め込むＮＡＬユニットが選択され、クライアントに送信される。

【００９４】

好ましくは、本発明では、１つの独立したスライスセグメントと、ゼロまたはそれ以上の従属スライスセグメントと、がＳＰＦに埋め込まれる。利点は、このＲＯＩを含むフレームの他の部分とは独立して、ＲＯＩへのアクセスを保証することである。

【００９５】

実際に、ＨＥＶＣ及びより一般的にはブロックベースのコーデックに対して、ＨＥＶＣタイル又は類似物は、それらの境界において全てのイントラフレーム予測メカニズム（ループフィルタリング処理を除く）を中断することに留意されたい。したがって、空間的予測メカニズムは許されない。しかしながら、いくつかの予測メカニズムは、圧縮を改善するために、ビデオシーケンスのフレーム間のデータの時間的冗長性に依存する。例えば、ＨＥＶＣタイル内の１つのブロックは、現在のＨＥＶＣタイル境界の部分的に又は全体的に外側にある予測ブロックから予測することができる。さらに、ＨＥＶＣは、予測ブロックが参照画像の部分的または全体的に外側にあることを可能にするために画像の境界を拡張する周知の境界拡張メカニズムを提供するので、予測ブロックは部分的または全体的にフレーム境界の外側にあってもよい。

【００９６】

最後に、予測ブロックは、サブピクセル位置に配置されてもよい。これは、基準ブロック画素値がサブ画素補間フィルタの結果であることを意味しており、これは、予測ブロックの対応する全画素座標に位置する画素のブロックの外側の最大４画素の範囲からサブ画素値を生成する。その結果、時間予測は、ＨＥＶＣタイル内のブロックと、ＨＥＶＣタイル境界の外側に位置する画素データのセットと、の間に符号化依存関係を導入することができる。

【００９７】

時間予測に関与する第２のＨＥＶＣメカニズムは、動きベクトル予測子を用いた動きベクトルの予測符号化にある。

【００９８】

最後に、ＨＥＶＣは、連続するタイルの画素間に依存関係を導入するループフィルタのセットを提供する。これらのループフィルタは、特に残差ブロックの量子化によって導入されるいくつかのアーチファクトを除去するデブロッキングフィルタとＳＡＯフィルタである。ＨＥＶＣは、これらのループフィルタがタイル境界又は／及びスライス境界でディスエーブルされているかどうかを示すために、ピクチャパラメータセット内にフラグを提供

する。ディスエーブルされると、タイル間の符号化依存関係はこれらの圧縮ツールによって導入されない。

【００９９】

注目領域の復号（注目領域を独立して復号することを意味する）を保証するために、解決策は、前述の予測メカニズムの一部または全部を無効にすることである。

【０１００】

これは、結果として得られるビットストリームについて、より高いビットレートおよびより効率的でない圧縮につながる。結果として得られるビットストリームのビットレートを最適化するために、注目領域使用シナリオに応じて予測メカニズムのアクティブ化／非アクティブ化を適応させることが可能である。

【０１０１】

図８ａおよび８ｂは、既に上述した、注目領域の組合せについて２つの異なる適用例を示す。

【０１０２】

例えば、第１の例では、図８ａが４つの注目領域から構成される２つの異なるビデオストリームからの２つのフレーム８００及び８０１を表す。第１のビデオストリーム８００は、高品質符号化パラメータを有し、第２の８０１は低品質であり、したがって、低ビットレートバージョンである。クライアントは、注目領域＃３の高品質バージョンを、領域１、２、および４の低品質注目領域と効率的に組み合わせる。これにより、他の重要度の低い領域に対してビットレートを比較的低く維持しながら、注目領域＃３の品質を強調することができる。

【０１０３】

第２の例では、４つのビデオストリーム（８０３、８０４、８０５、８０６）のセットが図８ｂに示されている。この使用シナリオでは、クライアントが各ビデオストリームの異なる注目領域の新しいモザイクビデオを形成する。彼は、各ビデオストリームの注目領域を、結果として得られるビデオストリーム内の新しい位置に再配置または結合する。

【０１０４】

本発明の一実施形態によれば、ここでは制約タイル（以下の説明ではＣＴｉｌｅとしてショートカット）と呼ばれる空間フレーム部分を定義することが提案される。これは、ランダムにアクセスされ、復号エラーなしに完全に復号されることができる空間フレーム部分に分割されたフレームのシーケンスに属する空間フレーム部分を指す。ＣＴｉｌｅの復号は、その空間的位置および／またはその近傍とは無関係に実行することができる。言い換えると、ＣＴｉｌｅは、復号器が常にエラーなしに復号できるように符号化される、または独立して符号化される。

【０１０５】

ＣＴｉｌｅを形成するサンプルに対応する符号化データは、独立して符号化される。例えば、データは、１つのパーサがＣＴｉｌｅに対応するサンプルを抽出できるように、スライス（または同様の機能を有するフレームの任意の他の部分）を形成するＮＡＬユニットまたは符号化ユニットに符号化される。その結果、２つのＣＴｉｌｅは、２つの異なる符号化ユニットのセットに符号化される。任意の空間位置で復号されるために、ＣＴｉｌｅは、他のＣＴｉｌｅを含むスライスの一部であることはできない。したがって、ＣＴｉｌｅに対応するスライスの符号化データは、他のスライスからの符号化データから独立している。

【０１０６】

スライスに対応する符号化データはさらに、いくつかのスライスセグメント符号化ユニットに分割されてもよい。

【０１０７】

一実施形態では、ＣＴｉｌｅは厳密に独立して復号可能であり、これはＣＴｉｌｅを形成する符号化データを解析するのに必要なすべてのデータが前記ＣＴｉｌｅに含まれることを意味する。さらに、予測メカニズムは、同じＣＴｉｌｅの符号化データから計算された

予測情報を使用する。インター予測の場合、参照ブロックは、別のフレーム内の同じＣＴｉｌｅから取り出される。

【０１０８】

他の実施形態では、符号化制限を解除することができる。

【０１０９】

第１の他の実施形態では、境界拡張メカニズムが画像境界のために使用される。図５に概略的に示すこのメカニズムは制限のない、より効率的な動き補償を可能にするために、ＣＴｉｌｅの境界に使用される。

【０１１０】

図５は、例えばＨＥＶＣで使用される境界拡張メカニズムを簡略化して示す。このメカニズムは、フレーム外のサンプル値を参照して、インター予測（現在のフレーム外のデータの使用を可能にする周知の予測モード）における動き補償を可能にする。

【０１１１】

フレーム５０２を符号化しながらブロック５０１を予測する場合、参照フレーム５０４の境界を横切るブロック５０３からの予測を可能にすることが有用である。これにより、例えば、以前のフレームの視野上で部分的に外側にあった同じコンテンツから動くコンテンツを予測することが可能になる。好ましくは、サンプルパディング方法は、参照ピクチャのフレーム境界の周囲のマージン内のサンプルにアクセスできるように定義される。

【０１１２】

第２の他の実施形態では、動きベクトル予測子（または任意の他の予測子）の派生メカニズムが、隣接するタイル情報やタイリング構成に全く依存しない方法で、ＣＴｉｌｅについて認可され得る。

【０１１３】

図９は、本発明によるビデオエンコーダにおいて実施される符号化処理の一例を示す。

【０１１４】

第１に、考慮される入力ビデオシーケンス９００ごとに、エンコーダは、ステップ９０１において、フレームのフレーム部分へのパーティショニング（すなわち、フレーム部分構成）を決定する。いくつかの実施形態では、フレーム部分のサイズが、１つのフレーム部分が注目領域の単一または一部をカバーするように予め決定される。例えば、フレーム部分は、５１２×５１２画素のサイズを有することができる。

【０１１５】

次いで、エンコーダは、ステップ９０２において、どのフレーム部分をＣＴｉｌｅとして符号化する必要があるかを決定する。例えば、そのようなフレーム部分は、クライアントが単独で復号することを望むか、またはクライアントが１つまたは複数の他の注目領域で構成することを望むことができる注目領域（ＲＯＩ）に対応することができる。

【０１１６】

次に、エンコーダは、ステップ９０３で、すべてのＣＴｉｌｅに識別子を決定して割り当てる。変形例では、エンコーダが識別子を決定し、ＣＴｉｌｅの選択のみに識別子を割り当てることができる。変形例では、ＣＴｉｌｅの識別子が推論されても良い。

【０１１７】

ＣＴｉｌｅに複数のエンコードされたフレームに同じ識別子が割り当てられている場合、これらのＣＴｉｌｅが同じＣＴｉｌｅシーケンスに属していることを意味する。ＣＴｉｌｅのシーケンス（またはＣＴｉｌｅシーケンス）は、他のフレーム部分とは独立して復号することができる。ＣＴｉｌｅシーケンスからのデータのみが、前記ＣＴｉｌｅシーケンスを復号するために必要とされる。換言すれば、ＣＴｉｌｅシーケンスからのＣＴｉｌｅは、一緒に時間的依存関係を有することができる。

【０１１８】

ＣＴｉｌｅおよびＣＴｉｌｅ識別子を決定した後、エンコーダは符号化設定に従ってフレーム部分９０４を圧縮して符号化する。フレーム部分の符号化は、任意のデコーダが前に説明したようにそれらを復号できることを保証する。

【０１１９】
エンコーダは、ステップ９０５で、フレーム部分構成情報を生成する。フレーム部分構成情報は、フレーム部分にパーティショニングするフレームの記述パラメータを決定することにある。また、ＣＴｉｌｅ識別子をフレーム内またはフレームのシーケンス内のそれらの位置に関連付けることによって、ＣＴｉｌｅの記述パラメータを決定することにある。図１３ａ、図１３ｂおよび図１３ｃ（第１の代替案）または図１５（第２の代替案）を参照して説明される、異なるシグナリング代替案が提案される。ステップ９０５は、パラメータセット（ＸＰＳ）においてシグナリングタイリングパラメータを生成することを含む。変形例では、ステップ９０５が後ではなく符号化ステップ９０４の前に実行される。
【０１２０】
ステップ９０６は、ＸＰＳのＮＡＬユニットと、圧縮されたＣＴｉｌｅフレーム部分とをビットストリームに任意にカプセル化することを含む。
【０１２１】
例えば、ストリーミングプロトコルに基づいて、このステップは、例えば、ISO BMFファイルフォーマットのような上位レベルのビデオ記述フォーマット内のビットストリームのカプセル化をさらに含み得る。例えば、ビデオデータをオーディオデータと多重化することも可能である。
【０１２２】
ステップ９０１、９０２、および９０３は、所定のフレーム部分位置を提供する１つ以上の設定ファイルを使用することによって実現することができ、フレーム部分がＣＴｉｌｅであるかどうか、およびＣＴｉｌｅのためにどの識別子を使用する必要があるかの情報を提供する。代替の実施形態では、フレーム部分およびＣＴｉｌｅが、例えば、ディープニューラルネットワーク、またはより単純なセグメンテーションアルゴリズムを使用して、ビデオコンテンツの分析から自動的に決定することができる。
【０１２３】
いくつかの実施形態で説明されるように、ステップ９０１は、ビデオシーケンス全体（またはビデオセグメント内のいくつかの連続するフレームについては少なくとも）内で一定であるパーティショニングを決定するために使用されることができる。これは、ＣＴｉｌｅの位置およびサイズがＣＴｉｌｅシーケンスの一部を含む少なくともいくつかのフレーム内のＣＴｉｌｅシーケンス全体内で一定であることであることを意味する。
【０１２４】
あるいは、決定されたフレーム部分がフレーム間の可変サイズおよび位置を有しても良い。
【０１２５】
図１０は、ビデオデコーダ内で、ビデオデコーダにおいて実施される復号処理の例を示す。復号処理は、上で定義したように、ＣＴｉｌｅの使用を含む。
【０１２６】
まず、ビデオデコーダは、パラメータセット（ＸＰＳ）を含むＮＡＬユニットを抽出する。ステップ１０００において、フレーム部分構成情報がパラメータセットから得られる。
【０１２７】
考慮されるフレーム部分１００１ごとに、デコーダは、ステップ１００２において、フレーム部分がＣＴｉｌｅであるかどうかをフレーム部分構成情報から判定する。
【０１２８】
フレーム部分がＣＴｉｌｅとしてシグナリングされる場合、テスト１００３の後、ブランチ「ｙｅｓ」で、デコーダは、フレーム部分構成情報からＣＴｉｌｅ識別子を抽出し（または推論し）、ステップ１００５において、ＣＴｉｌｅ識別子および識別子に関連付けられたＣＴｉｌｅ位置情報のおかげで、ＣＴｉｌｅの復号位置を決定する。そうではなく、フレーム部分がＣＴｉｌｅでない場合、テスト１００３の後、ブランチ「ｎｏ」で、デコーダは、ステップ１００５において、フレーム部分符号化データに記述された測位情報から、およびＸＰＳ情報から、フレーム部分の復号位置を決定する。
【０１２９】

最後に、デコーダはフレーム部分がＣＴｉｌｅであるか否かを考慮して、フレーム部分符号化データ１００６を復号し、復号されたサンプル値をレンダリングピクチャバッファ内に入れる。

【０１３０】

図１１は、図９の符号化処理で生成された２つのビットストリームのマージ処理の例（図８ａおよび図８ｂのアプリケーションを参照）を示す。マージ処理は、抽出されたＣＴｉｌｅがクライアントに送信される新しいビデオビットストリームに結合されることを意味する。

【０１３１】

マージ処理は、ステップ１１００において、１つ以上のビデオビットストリームから抽出され、新しいビットストリームにマージされるＣＴｉｌｅのセットを決定することによって開始する。例えば、グラフィカルユーザインターフェースは、ユーザがＣＴｉｌｅのセットを選択し、フレーム内でそれらを再配置することも可能にする。別の例では、ビットストリームの内容に基づいて選択が自動的に実行される。アプリケーションは、移動するコンテンツを含むＣＴｉｌｅのセットを選択することができる。

【０１３２】

処理は、ステップ１１０１において、新しいビデオビットストリームにマージされたときのＣＴｉｌｅの新しい位置を決定する。

【０１３３】

抽出されるＣＴｉｌｅが分かると、ステップ１１０２において、抽出される決定されたＣＴｉｌｅの各々の現在のＣＴｉｌｅ識別子を取得することによって、それらの新しい識別子が決定される。これらの識別子は、本発明の実施形態に従って、フレーム構成情報においてシグナリングされる。前述したように、代替の実施形態では、フレーム構成情報が、入力ビットストリームをカプセル化するために使用されるファイルフォーマットで記述されても良い。フレーム構成情報は、ＸＰＳおよびファイルフォーマットの中に存在することがある。

【０１３４】

２つ以上のＣＴｉｌｅが同じ識別子を有することを意味する識別子衝突の場合、ステップ１１０１は、これらの衝突を解決するための新しいＣＴｉｌｅ識別子を決定することをさらに含む。

【０１３５】

次に、処理は、新しいビデオビットストリームのマージされたビデオシーケンスのためのフレーム部分構成情報を生成する（１１０３）。これは、マージされたビットストリーム内のＣＴｉｌｅの新しい位置をそれらの新しいＣＴｉｌｅ識別子に関連付けるＸＰＳの１つにおいてパラメータを生成することを含む。

【０１３６】

ステップ１１０４では、ステップ１１００で決定されたＣＴｉｌｅのセットの符号化フレーム部分データが抽出または取得される。それは、ＣＴｉｌｅの符号化フレーム部分データを含むＮＡＬユニットを取り出すことを含む。これは、ステップ１１０２で決定されたＣＴｉｌｅ識別子を有するものを抽出するために、入力ビットストリーム中のすべてのＮＡＬユニットを構文解析することによって行うことができる。入力ビットストリームがファイルフォーマット仕様に準拠する場合、１つのフレーム部分に対応するすべてのＮＡＬユニットは１つのコンテナ、例えば、ＩＳＯＢＭＦＦ用のビデオトラックにカプセル化される。次に、ステップ１１０４は、選択されたフレーム部分のトラックに対応するデータを取り出すことを含む。

【０１３７】

最後に、オプションのステップ１１０５で、ＸＰＳのＮＡＬユニットと、抽出されたＣＴｉｌｅ符号化フレーム部分データを含むＮＡＬユニットとを新しいビットストリームに埋め込み、場合によっては、このビットストリームをより高いレベルの記述フォーマットにカプセル化することによって、新しいビットストリームが生成される。

【０１３８】
ＣＴｉｌｅ識別子衝突のためにステップ１１０１で新しいＣＴｉｌｅ識別子が決定されたＣＴｉｌｅについて、ステップ１１０５は、オリジナルのＣＴｉｌｅ識別子を含むＮＡＬユニットに含まれるヘッダを修正することをさらに含む。これらのヘッダは、オリジナルのＣＴｉｌｅ識別子がステップ１１０２で決定されたＣＴｉｌｅ識別子によって置き換えられるように修正される。
【０１３９】
１つの例では、図１１のマージ処理は、同じビットストリームからＣＴｉｌｅのサブセットを抽出することにある。このような場合、識別子の衝突を処理する必要はない。
【０１４０】
図１３ａおよび１３ｂおよび１３ｃは、本発明のいくつかの実施形態による、符号化処理によって実行されるフレーム部分構成のシグナリングの例を示す。
【０１４１】
図１３ａは、本発明の一実施形態によるビットストリーム内のＣＴｉｌｅの識別を示す。
【０１４２】
ここで、ｃｔｉｌｅ＿ｕｎｉｑｕｅ＿ｉｄｅｎｔｉｆｉｅｒ１３０１という名前のＣＴｉｌｅ識別子は、フレーム部分符号化データ中に示される。好ましくは、識別子がフレーム部分符号化データに属する各データシーケンス（すなわち、スライスセグメントヘッダ）において示される。したがって：
－ビットストリームのどの部分がＣＴｉｌｅに属しているかについての簡単な識別、および
－これらの部分の迅速なアクセスまたは抽出
が可能である。
【０１４３】
より正確には、図１３ａに示す実施形態では、ＣＴｉｌｅ識別子１３０１が、識別子１３０１を有するＣＴｉｌｅに対応する各スライスセグメントのスライスセグメントヘッダ（ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｈｅａｄｅｒ）１３０２においてシグナリングされる。
【０１４４】
前述のように、デコーダは、フレーム部分構成情報に基づき、ＣＴｉｌｅ識別子を解析して、ＣＴｉｌｅの関連する位置を決定する。実施形態によれば、フレーム部分構成情報は、図１３ｂまたは図１３ｃを参照して後述するように、パラメータセット、例えば、ＴＰＳ）で提供される。
【０１４５】
明示的に言及されない限り、または適用できない限り、以下の簡潔さのために、ＣＴｉｌｅとＣＴｉｌｅシーケンスとの間に区別はない。さらに、ＣＴｉｌｅ識別子は、ＣＴｉｌｅシーケンス識別子と見なすこともできる。
【０１４６】
実施形態では、識別子が必ずしも必要ではないＨＥＶＣタイプのタイルをＣＴｉｌｅと区別するために、例えばｃｔｉｌｅ＿ｆｌａｇ１３０３のような情報がフレーム部分符号化データに属するデータシーケンス（例えば、スライスセグメントヘッダ内）で使用されてもよい。ｃｔｉｌｅ＿ｆｌａｇが非アクティブの場合（例えば’ｆａｌｓｅ’に設定されている）、ＨＥＶＣタイプのタイルのパラメータ１３０４が提供される。これらのパラメータは、たとえばｆｉｒｓｔ＿ｓｌｉｃｅ＿ｉｎ＿ｐｉｃ＿ｆｌａｇやＣＴＵ宛先（ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ａｄｒｅｓｓ）のようなタイル位置決め情報や、ｓｌｉｃｅ＿ｐｉｃ＿ｐａｒａｍｅｔｅｒ＿ｓｅｔ＿ｉｄのような他のビットストリーム要素への参照を含み得る。これらの構文要素は、フレームパーティショニングに依存し、あるビデオシーケンスと他とで異なる場合がある。
【０１４７】
ｃｔｉｌｅ＿ｆｌａｇがアクティブの場合、これらのパラメータは省略され、ＣＴｉｌｅ固有識別子１３０１）を構成するＣＴｉｌｅ固有情報が代わりに提供される。ＣＴｉｌｅで複数のスライスを持つことができるようにするため、１つのソリューションは、ここで

ｃｔｂ＿ａｄｄｒ＿ｏｆｆｓｅｔ＿ｉｎｓｉｄｅ＿ｔｉｌｅ１３０５という名前の情報を提供することである。この情報１３０５はまた、考慮されるフレームを有するＣＴｉｌｅ位置に対してスライスセグメントの復号を開始する位置を指定するために使用される。例えば、この位置は、ＣＴｉｌｅの先頭とその幅（ＣＴＢ）に対して、生のスキャンの順序付けられた符号化ブロック数（例えば、ＨＥＶＣ標準符号化ツリーブロックであるＣＴＢ）で表されるので、ｃｔｂ＿ａｄｄｒ＿ｏｆｆｓｅｔ＿ｉｎｓｉｄｅ＿ｔｉｌｅ情報は、ＣＴｉｌｅ符号化／復号位置とは無関係である。

【０１４８】

別の実施形態では、フラグｃｔｉｌｅ＿ｆｌａｇは使用されない。例えば、ＣＴｉｌｅ識別子は、全てのタイル、ＣＴｉｌｅ及び他のタイル（ＨＥＶＣタイプのタイル）に存在する。所定の値、例えば値０は、ＨＥＶＣタイプのタイルを識別するために使用されてもよい。

【０１４９】

一実施形態では、情報が、空間フレーム部分がＣＴｉｌｅであるか否かを識別するために提供される。

【０１５０】

別の実施形態では、ＣＴｉｌｅのみが使用されるフレーム部分であると仮定すると、空間フレーム部分がＣＴｉｌｅであるか否かを識別するために情報は提供されない。

【０１５１】

好ましくは、所与のフレームにおいて、所与の識別子を有するＣＴｉｌｅは１つ以下である。同じＣＴｉｌｅ識別子が時間的に依存する（例えば、ＣＴｉｌｅシーケンスにおいて）全てのＣＴｉｌｅにおいて使用される。したがって、連続して符号化されたピクチャ内の同じＣＴｉｌｅ識別子を有するＣＴｉｌｅが抽出される場合、それらは適切に復号されることになる。

【０１５２】

言い換えると、ＣＴｉｌｅ識別子は、符号化されたビデオシーケンス内のＣＴｉｌｅを識別する一意の識別子である。一実施形態では、ＣＴｉｌｅ識別子が、ＣＴｉｌｅに含まれるスライスセグメントのスライスヘッダに挿入される。これは、ビットストリームにおいて、ＣＴｉｌｅに対応するＮＡＬユニット（スライスセグメント）がＣＴｉｌｅ識別子を含むことを意味する。したがって、任意のＣＴｉｌｅを解析し、このＣＴｉｌｅ識別子に基づいてビットストリームから簡単に抽出できる。

【０１５３】

ビットストリーム内のＣＴｉｌｅ構成情報をシグナリングすることが有利である。例えば、ＣＴｉｌｅ構成情報は、ＣＴｉｌｅの数、関連するＣＴｉｌｅ識別子、およびフレーム内のＣＴｉｌｅの位置によって定義される。

【０１５４】

図１３ｂは、本発明の一実施形態によるＣＴｉｌｅ構成を示す。

【０１５５】

第１の実施形態では、エンコーダが、復号されるピクチャ内のＣＴｉｌｅのフレーム部分構成情報に関連する追加のシグナリング情報を特定する。シグナリング情報は、パラメータセット（ＸＰＳ）において提供され、好ましくは、タイリングパラメータセット（ＴＰＳにおいて提供される。好ましくは、追加のシグナリング情報が、ここでｎｕｍ＿ｃｔｉｌｅｓと呼ばれるピクチャ内のＣＴｉｌｅの番号１３１１を含む。ＣＴｉｌｅごとに、ここではｔｉｌｅ＿ｃｔｂ＿ａｄｄｒと呼ばれるＣＴｉｌｅ位置１３１３と共にＣＴｉｌｅの一意の識別子１３１２を関連付け、ピクチャ内の復号位置を意味する。ＣＴｉｌｅ位置が、ピクチャ内の復号位置として提供される。ＣＴＢインデックス番号（例えば、ラスタースキャン順序に対して相対的）で表現されてもよい。

【０１５６】

別の実施形態では、ここでｓｌｉｃｅ＿ｐｉｃ＿ｐａｒａｍｅｔｅｒ＿ｓｅｔ＿ｉｄと命名されたパラメータは、１３０４によって指定された部分における図１３ａで表されたス

ライスセグメントヘッダにも言及されており、ＴＰＳを表す固有の識別子を参照する。変形例では、一意の識別子はＰＰＳを表す。この他の実施形態では：
－各ＴＰＳは、ＴＰＳを識別するｔｉｌｅ_ｐａｒａｍｅｔｅｒ_ｓｅｔ_ｉｄ（簡略化のために図示せず）パラメータを含み－たとえば、ピクチャにおいてＣＴｉｌｅ構成が変化するたびに、エンコーダが新しいＴＰＳを生成する可能性があることを意味しており－フレームの各スライスヘッダを書き換えることを避けるために、同じＴＰＳ（またはＰＰＳ）一意の識別子を持つＴＰＳを生成することをお勧めする。
－スライスセグメントヘッダのｓｌｉｃｅ＿ｐｉｃ＿ｐａｒａｍｅｔｅｒ＿ｓｅｔ＿ｉｄ１３１４は、スライスに適用されるＴＰＳのｔｉｌｅ＿ｐａｒａｍｅｔｅｒ＿ｓｅｔ＿ｉｄに等しい。そのような場合、スライスセグメントヘッダのｓｌｉｃｅ＿ｐｉｃ＿ｐａｒａｍｅｔｅｒ＿ｓｅｔ＿ｉｄの名前をｓｌｉｃｅ＿ｔｉｌｅ＿ｐａｒａｍｅｔｅｒ＿ｓｅｔ＿ｉｄに名前を変更できる。

【０１５７】
１つの代替例では、ＴＰＳ識別子は、スライスデータにおいて指定されず、デコーダは、スライスＮＡＬユニットに先行する最後のＴＰＳ　ＮＡＬユニットが現在のＣＴｉｌｅのフレーム部分構成を含むことを推論する。

【０１５８】
図１３ｃは、本発明の別の実施形態によるＣＴｉｌｅ構成を示す。この実施形態によれば、ＴＰＳ１３２０は、タイルの数マイナス１を示すパラメータ値、例えば’ｎｕｍ＿ｔｉｌｅｓ＿ｍｉｎｕｓ１’１３２１を含む。または、ＴＰＳは、フレーム内のタイル数を直接的または間接的に提供するインスタンス’ｎｕｍ＿ｔｉｌｅｓ’という名前のパラメータ値を含む。

【０１５９】
一実施形態では、ＴＰＳが１つのフレーム部分しかないことを示す場合、これはビデオフレームと同じ寸法を有し、その起点に位置するＣＴｉｌｅであると仮定される。さもなければ（ＴＰＳはいくつかのフレーム部分を記述する）、フレーム部分位置は前の実施形態のように記述される。

【０１６０】
別の実施形態では、ＴＰＳがない場合、ビデオフレームと同じ寸法を有する１つのＣＴｉｌｅがあると仮定される。前記１つのＣＴｉｌｅは、その起点に位置する。

【０１６１】
別の実施形態によれば、ＴＰＳはＨＥＶＣ格子に類似した構文、例えば、’ｎｕｍ＿ｔｉｌｅ＿ｒｏｗｓ＿ｍｉｎｕｓ１’、’ｎｕｍ＿ｔｉｌｅ＿ｃｏｌｓ＿ｍｉｎｕｓ１’、および’ｕｎｉｆｏｒｍ＿ｓｐａｃｉｎｇ＿ｆｌａｇ’を特定し、該構文を有する空間フレーム部分格子を記述することができる。’ｕｎｉｆｏｒｍ＿ｓｐａｃｉｎｇ＿ｆｌａｇ’が設定されていない場合、各列の幅と各行の高さ（推定可能な列のサイズと最後の行のサイズを除く）も指定される。’ｕｎｉｆｏｒｍ＿ｓｐａｃｉｎｇ＿ｆｌａｇ’が設定されている場合、ＣＴｉｌｅの幅と高さは、例えばＨＥＶＣ仕様のように、ピクチャの幅と高さから計算される。そのような実施形態では、格子インデックスが対応するＣＴｉｌｅを局所化することを可能にするので、ＣＴｉｌｅ位置は、空間フレーム部分格子インデックスに対応するＣＴｉｌｅ数（例えば、タイルのラスタ走査順序を使用する）によって表現され得る。

【０１６２】
代替の実施形態によれば、’ｃｔｉｌｅ＿ｆｌａｇ’は、いくつかの値をとることができる’ｃｔｉｌｅ＿ｌｅｖｅｌ’に置き換えられ、各値は、ＣＴｉｌｅに適用される異なるレベルのエンコード制約を示す。例えば、ｃｔｉｌｅ＿ｌｅｖｅｌがゼロに等しい場合は、ＣＴｉｌｅが制約されないことを示す（ＨＥＶＣタイプのタイルのように）。ｃｔｉｌｅ＿ｌｅｖｅｌが’１’に等しい場合は、オリジナルの近傍で復号できるが他のＣＴｉｌｅでシャッフルすると適切に復号されない可能性がある、若しくは単独で（オリジナルの近傍なしで）抽出して適切に復号できるように、ＣＴｉｌｅが制約されることを示す。ｃｔｉｌｅ＿ｌｅｖｅｌが’２’に等しい場合は、任意の近傍（前の実施形態における１に等しいｃｔ

ｉｌｅ＿ｆｌａｇと等価）でシャッフルされ、どこでも復号できるようにＣＴｉｌｅが制約されることを示す。

【０１６３】

別の実施形態では、’ｃｔｉｌｅ＿ｌｅｖｅｌ’が、エンコーダが制約のレベルを満たすためにエンコード決定を行った情報を提供するだけである。したがって、任意のレベルの制約を有するＣＴｉｌｅの復号処理は、ＨＥＶＣタイプのタイルと同じ復号処理によって実施することができる（例えば、ＣＴｉｌｅ境界に対して境界拡張は実行されない）。

【０１６４】

別の実施形態では、符号化および復号処理がすべてのレベルの制約について同じではない。たとえば、ｃｔｉｌｅ＿ｌｅｖｅｌが’１’のＣＴｉｌｅは、ＨＥＶＣタイプのタイルと同じ復号処理を使用する（エンコーダで使用され、デコーダに影響を与えないいくつかの制限）。一方、ｃｔｉｌｅ＿ｌｅｖｅｌが’２’のＣＴｉｌｅは、ＣＴｉｌｅ境界の境界拡張を使用して、モーションベクトル予測子のリストの特定の派生処理を使用して、復号される必要がある。

【０１６５】

別の実施形態によれば、ＨＥＶＣタイプのタイルであっても、識別子（例えば、ＸＰＳ内のそれらのパラメータを関連づけるため）を有する必要がある場合があり、この識別子は、ＣＴｉｌｅ識別子と同様の方法でスライスセグメントヘッダ内に指定される。所与のフレームでは、ＨＥＶＣタイプのタイルがＣＴｉｌｅと同じ識別子も、別のＨＥＶＣタイプのタイルと同じ識別子も有しない。

【０１６６】

一実施形態によれば、エンコーダは、パラメータセットのうちの１つ、例えばＰＰＳまたはＴＰＳにおける各空間フレーム部分の’ｃｔｉｌｅ＿ｆｌａｇ’の値をシグナリングすることによって、空間フレーム部分がＣＴｉｌｅであることを示す。例えば、エンコーダは、フレームの各タイルについて固有の識別子を生成する。フレーム部分構成を記述するとき、エンコーダは、フラグ（例えばｃｔｉｌｅ＿ｆｌａｇ）を各タイル固有識別子に関連付ける。このフラグは、対応するタイル（すなわち、関連する固有識別子と等しい識別子をもつタイル）の符号化が、独立した復号を保証するために制約されるとき、真である。反対に、タイルの符号化が独立した復号を保証するのに十分に制約されていない場合、フラグは偽である。

【０１６７】

第２の実施形態によれば、エンコーダは、別のフラグ（例えば、ａｌｌ＿ｃｔｉｌｅ＿ｆｌａｇ）を含むフレーム部分構成情報を生成する。このフラグが「１」に設定されている場合には、フレーム部分構成に記述されている全てのタイルがＣＴｉｌｅであることを意味する。空間フレーム部分がＣＴｉｌｅであるかどうかを示すフラグ（例えば、各ｃｔｉｌｅ＿ｆｌａｇ）は省略され、真と等しいと推論される。このフラグがゼロに設定されている場合、ＣＴｉｌｅは、先の実施形態の１つを使用して明示的に記述されている。パラメータがＨＥＶＣタイプのタイルに固有の場合、スライスセグメントヘッダではなく、ＴＰＳなどでＸＰＳでシグナリングされる。例えば、別の実施形態のために参照により１３０４に組み込まれるｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ａｄｄｒｅｓｓは、ＨＥＶＣタイプのタイルに特有である。一実施形態では、ＴＰＳが、空間フレーム部分がＣＴｉｌｅではないことも示す場合、ＴＰＳに示される。この実施形態は、スライスセグメントヘッダの構文解析及び構文を単純化することを可能にする。

【０１６８】

別の実施形態によれば、エンコーダは、スライスセグメントヘッダ内の’ｃｔｉｌｅ＿ｆｌａｇ’を使用する代わりに、ＣＴｉｌｅに対応するスライスデータのための新しいＮＡＬユニットタイプを定義する。例えば、エンコーダは、ＣＴｉｌｅ内にある瞬時復号リフレッシュ（ＩＤＲ）フレームからスライスＮＡＬユニットのＣＴＩＬＥ＿ＩＤＲＮＡＬユニットを定義する。エンコーダは、符号化フォーマットが通常のスライスデータに対して指定するのと同じ数の新しいＮＡＬユニットタイプを定義する。たとえば、ＨＥＶＣは、以下

のＮＡＬユニットのタイプ クリーンランダムアクセス（ＣＲＡ）ピクチャのスライスセグメントに対してＣＲＡ＿ＮＵＴ、ランダムアクセス復号可能リーディング（ＲＡＤＬ）ＩＤＲピクチャのスライスセグメントに対してＩＤＲ＿Ｗ＿ＲＡＤＬ、ビットストリームに関連する先行ピクチャが存在しないＩＤＲピクチャのスライスセグメントのＩＤＲ＿Ｎ＿ＬＰ、破損リンクアクセス（ＢＬＡ）ピクチャのスライスセグメントに対してＢＬＡ＿Ｗ＿ＬＰ、ＢＬＡ＿Ｗ＿ＲＡＤＬ、ＢＬＡ＿Ｎ＿ＬＰ、ランダムアクセススキップドリーディング（ＲＡＳＬ）ピクチャのスライスセグメントに対してＲＡＳＬ＿Ｎ、ＲＡＤＬ＿Ｒ、ＲＡＤＬピクチャのスライスセグメントに対してＲＡＤＬ＿Ｎ、ＲＡＤＬ＿Ｒ、ステップワイズテンポラルサブレイヤアクセス（ＳＴＳＡ）ピクチャのスライスセグメントに対してＳＴＳＡ＿Ｎ、ＳＴＳＡ＿Ｒ、テンポラルサブレイヤアクセス（ＴＳＡ）ピクチャのスライスセグメントに対してＴＳＡ＿Ｎ、ＴＳＡ＿Ｒ、非ＳＴＳＡの末尾のピクチャ、非ＴＳＡのスライスセグメントに対するＴＲＡＩＬ＿Ｎ、ＴＲＡＩＬ＿Ｒ、を定義する。Ｗ＿ＬＰ：関連付けられたＲＡＳＬまたはＲＡＤＬピクチャを持つことができる。Ｗ＿ＲＡＤＬ：関連付けられたＲＡＳＬピクチャなし。Ｎ＿ＬＰ：関連付けられたリーディングピクチャなし。＊＿Ｎ：ピクチャはサブレイヤ非参照（ＳＬＮＲ）ピクチャである（それ以外の場合は、サブレイヤ参照ピクチャである）。＊＿Ｒ：ピクチャは単なるサブレイヤ参照ピクチャである。

【０１６９】

これらのHEVC NALユニットタイプは、制約されたタイルデータに対して同じ目的で、新しい対応するＮＡＬユニットタイプＣＴＩＬＥ＿ＢＬＡ＿＊、ＣＴＩＬＥ＿ＣＲＡ＿＊、ＣＴＩＬＥ＿ＩＤＲ＿＊、ＣＴＩＬＥ＿ＲＡＳＬ＿＊、ＣＴＩＬＥ＿ＲＡＤＬ＿＊、ＣＴＩＬＥ＿ＳＴＳＡ＿＊、ＣＴＩＬＥ＿ＴＳＡ＿＊、ＣＴＩＬＥ＿ＴＲＡＩＬ＿＊で拡張できる。これらの新しいＮＡＬユニットタイプの１つを使用することは、ＮＡＬユニットがＣＴｉｌｅに属することを示す。

【０１７０】

この代替案は、エンコーダが各ＮＡＬユニットの第１のビットを解析して、スライスデータがＣＴｉｌｅ内にあるか否かを判定するだけでよいので、復号処理を単純化する。

【０１７１】

好適には、ＣＴｉｌｅ識別子の単一性は、与えられたビットストリームにおいて意味する、与えられたシーケンスのための符号化時の構築によって保証される。ただし、異なるビットストリームから由来することを意味する、異なるシーケンスからＣＴｉｌｅをシャッフルする場合、単一性は保証されない。一実施形態によれば、様々なシーケンスから潜在的に由来するＣＴｉｌｅとの空間フレーム部分のシャッフルを容易にするために、ＣＴｉｌｅ識別子は、限られた数のビットで一意である。一意の値はランダムな値、例えば、ハッシュ値、またはその位置を必ずしも表すとは限らない他の任意の値とすることができる。したがって、異なるビットストリームのＣＴｉｌｅを取るときに識別子衝突を有する確率を低減する。

【０１７２】

一実施形態では、複数のシーケンスからＣＴｉｌｅのシャッフリングを実行するとき、２つのＣＴｉｌｅ識別子間の衝突の場合、衝突したＣＴｉｌｅ識別子を置き換えれば十分である。すべてのスライスセグメントヘッダを再生成することを強制されないことによって効率的に行うために、好ましい実施形態では、ＣＴｉｌｅ識別子を符号化するために、固定の所定数のビットが使用される。例えば、図１３ａおよび１３ｂでは、ＣＴｉｌｅ識別子は８ビットで符号化される。

【０１７３】

代替の実施形態では、シーケンスまたはピクチャのすべてのＣＴｉｌｅ識別子が同じビット数で符号化される。このビット数は、例えばＳＰＳ、ＰＰＳ、ＲＰＳ、’ｕｉｄ＿ｎｕｍ＿ｂｉｔｓ’のような一つのパラメータセットのパラメータで指定される。スライスセグメントヘッダでは、ＣＴｉｌｅ識別子の後にバイトアライメントメカニズムを持つことが好ましい（８の倍数ではないビット数を取る場合）。または、ビット数をバイト数（８ビッ

ト）で表すこともでき、たとえば、’ｕｉｄ＿ｎｕｍ＿ｂｙｔｅｓ’である。ＣＴｉｌｅをさまざまなシーケンスからまとめてシャッフルする場合、ＣＴｉｌｅ識別子を、すべて同じビット数を有さないときに変更する必要があるかもしれない。これには、いくつかのスライスセグメントヘッダを変更する必要があるが、バイトのみを追加／削除または置換する必要があるため、スライスセグメントヘッダを更新するよりも簡単である。

【０１７４】

さらに別の代替の実施形態では、各ＣＴｉｌｅ識別子が可変数のビットで符号化されてもよい。そのビット数は、スライスセグメントヘッダで指定される。あるいはビット数がＣＴｉｌｅ識別子のために使用されるコードから自動的に決定され得る：可変バイト長コードは例えば、指数ゴロム符号化（または等価的に可変長コードの後にバイトアライメントビットが続く）を使用される。

【０１７５】

実施形態によれば、ＣＴｉｌｅ識別子はシグナリングサイズを低減するために、従属スライスセグメントヘッダにおいてシグナリングされない。次に、従属スライスセグメントヘッダのＣＴｉｌｅ識別子が、前の独立スライスセグメントヘッダから推論される。代替の実施形態によれば、ＣＴｉｌｅ識別子はＣＴｉｌｅを含むサブビットストリームの構文解析および抽出を容易にするために、従属スライスヘッダでシグナリングされる。

【０１７６】

’ｔｉｌｅ＿ｃｔｂ＿ａｄｄｒ［ｉ］’１３１３または’ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ａｄｄｒｅｓｓ’１３０６符号化ユニットアドレスでＣＴｉｌｅ位置をシグナリングする代わりに、きめ細かいＣＴｉｌｅが導入され、より細かい位置決めが行われる。この粒度はルマサンプル位置まで精緻化することができるが、別の実施形態では、「２」の累乗（ＣＴＵサイズより小さい）に対応する多数のルマサンプルの粒度で十分である。いくつかの実施形態では、粒度が予め決定されてもよい。代替の実施形態では、粒度が例えば、ＶＰＳ、ＳＰＳ、またはＰＰＳでシグナリングされる。細粒ＣＴｉｌｅが使用される場合、ＣＴｉｌｅの寸法は、必ずしもＣＴＵサイズの倍数ではない。

【０１７７】

ＣＴｉｌｅのサイズがＣＴＵサイズの倍数でない場合、ＣＴｉｌｅの右側および下側の符号化ユニットは図４に示すように、ピクチャの右側および下側のＨＥＶＣ　ＣＴＵに使用されるものと同様の自動分割メカニズムを使用する。

【０１７８】

代替の実施形態によれば、符号化ユニットが不完全であっても、構文は完全な符号化を記述し、分解ツリー（例えば、クアツリーまたはＱＴＢＴ）のレート歪み最適化のためのいくつかの空間を与え、圧縮を改善するのに適した情報の最終的なパディングを可能にする。

【０１７９】

ＨＥＶＣタイプのタイルでは、寸法はグリッドを用いて指定される。従って、全てのＨＥＶＣタイプのタイルは行及び列によって整列され、所与の行の全てのＨＥＶＣタイプは同じ高さを有し、所与の列の全てのＨＥＶＣタイプのタイルは同じ幅を有する。各列の幅と各行の高さは、ＸＰＳで指定される。きめの細かいＨＥＶＣタイプのタイルでは、例えば、複数のＲＯＩのより効率的な符号化を可能にするために、より厳密でないアレンジメントを可能にすることが便利であり得る。

【０１８０】

一実施形態によれば、ＣＴｉｌｅの寸法は、ＣＴｉｌｅのスライスセグメントのスライスセグメントヘッダ内で指定することができる。結果として得られるビットストリームのサイズを縮小するために、ＣＴｉｌｅの寸法は、第１のスライスセグメントにおいてのみ指定される。以下のスライスセグメントは、同じＣＴｉｌｅ寸法を再利用する。代わりに、すべてのＣＴｉｌｅの寸法がＸＰＳで提供され、たとえば、ＣＴｉｌｅの位置と一緒に提供される。

【０１８１】

別の代替として、寸法は、ＣＴｉｌｅの第１のスライスセグメントヘッダおよびＸＰＳの

両方において提供される。
【０１８２】
別の代替として、ＣＴｉｌｅの寸法は提供されないが、ＸＰＳ内のタイル情報（例えば、位置または’ｃｔｉｌｅ＿ｆｌａｇ’）を提供するために使用される順序付けから、およびＣＴｉｌｅ位置から推論され、例えば、ＣＴｉｌｅ位置はＸＰＳ内で宣言され、ＣＴｉｌｅ位置は、ＣＴｉｌｅの対応する右下隅のそれぞれが（例えば）ラスタスキャン順序において昇順で順序づけられるように、並べられる。以下の図１４は、そのような順序付けの例を提供する。
【０１８３】
実施形態によれば、図１３ａの例で使用されるｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇは、ＨＥＶＣにおけるものと同じ意味を有する。これは、従属スライスセグメントが許されるか否かを示すために使用される。ＨＥＶＣでは、ｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇがＰＰＳでシグナリングされる。我々の好ましい実施形態によれば、ｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇは、ｔｉｌｉｎｇ_ｐａｒａｍｅｔｅｒ_ｓｅｔ（ＴＰＳ）において、各ＣＴｉｌｅについてシグナリングされる（従属スライスセグメントと共に符号化されたＣＴｉｌｅを、同じバイストリーム内の従属スライスセグメントなしに符号化されたＣＴｉｌｅと共に使用することを可能にするため）。すべてのＣＴｉｌｅが従属スライスセグメントの有無にかかわらず符号化される一般的なユースケースのＴＰＳの構文を減らすために、ＴＰＳ構造のルートにｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇ＿ｆｏｒ＿ａｌｌ＿ｃｔｉｌｅｓという別のフラグが使用される。このフラグが１に設定されている場合、ｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｄｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇはＣＴｉｌｅごとにシグナリングされない。代わりに、ｃｔｉｌｅ＿ｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇもＴＰＳ構造のルートにシグナリングされ、各ＣＴｉｌｅのｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｄｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇに対して推論される値を提供する。ＨＥＶＣタイプのタイルの場合、ｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔ＿ｅｎａｂｌｅｄ＿ｆｌａｇは、ＰＰＳでシグナリングすることができるが、好ましい実施形態ではＴＰＳでシグナリングされる。
【０１８４】
代替の実施形態によれば、ｄｅｐｅｎｄｅｎｔ＿ｓｌｉｃｅ＿ｓｅｇｍｅｎｔｓ＿ｅｎａｂｌｅｄ＿ｆｌａｇは構文を単純化するために、まったくシグナリングされず、常にｔｒｕｅとして推論される。
【０１８５】
図１４は、非格子ベースのパーティショニングの例を示す。フレーム１４０１は、＃１から＃１５まで番号付けされた１５個のＣＴｉｌｅに分割される。この番号付けは、各タイルの右下隅がラスタスキャン順序で順序付けられるように、ＸＰＳ内のタイル位置の宣言順序を提供する。この順序付けを使用して、各ＣＴｉｌｅのサイズを推定することができる。例えば、最後のＣＴｉｌｅ、ＣＴｉｌｅ＃１５を取ると、最後のタイルであるので、その右下隅がラスタスキャン順序において最後であり、従って、それがフレームの右下隅であるので、その寸法を推定することができる。スライス＃１５の寸法は、フレームの寸法からその位置を引いたものである:ｈ＃１５＝ｈ＿ｆｒａｍｅ－ｙ＃１５；ｗ＃１５＝ｗ＿ｆｒａｍｅ－ｘ＃１５。タイル＃１４は、ＣＴｉｌｅ＃１５の前の最後の右下隅をそれ自体の右下隅として有しなければならず、したがって、右下隅位置は最も下（フレームの底部）であり、最も右（前のタイルの左端）である。ＣＴｉｌｅ＃１４の寸法は、ｈ＃１４＝ｈ＿ｆｒａｍｅ－ｙ＃１４；ｗ＃１４＝ｘ＃１５－ｘ＃１４となる。同じことがタイル＃１３および＃１２についても繰り返される。次に、タイル＃１１については、最も下の位置が満たされているので、新しい最も下の位置はｙ＃１４である。次に、ｈ＃１１＝ｙ＃１４－ｘ＃１１である。そしてＣＴｉｌｅ＃１まで続く。
【０１８６】

代替の実施形態によれば、ＸＰＳ内のＣＴｉｌｅ位置を指定する代わりに、ＣＴｉｌｅの寸法のみが指定され、ＣＴｉｌｅ位置は、左上の位置に従って順序付けられたＣＴｉｌｅを用いて（例えば、ラスタスキャン順序が増加して）、ＣＴｉｌｅの寸法から計算される。寸法から位置を計算するアルゴリズムは、位置から寸法を計算するために前述したアルゴリズムから容易に導出され得る。

【０１８７】

一実施形態によれば、ＸＰＳに記述されたＣＴｉｌｅパラメータは存在しないＣＴｉｌｅに対してＣＴｉｌｅ位置（および／またはＣＴｉｌｅの寸法）を提供することができ、それらのＣＴｉｌｅに対してスライスセグメントは存在しない。このＣＴｉｌｅの記述は、位置または寸法のみが提供され、寸法または位置が推論される実施形態において、適切な推論を可能にするために必要である。

【０１８８】

ビデオレンダリングでは、デフォルトのサンプル値もしくはパディング方法が、存在しないＣＴｉｌｅを埋めるために使用される。あるいはパディング方法のインデックスもしくは値がＸＰＳパラメータにおいて提供される。これは例えば、

－適切なデフォルトサンプル値でレンダリングバッファ内のフレームの内容を初期化すること、および／または

－例えば、インペインティング方法を使用することによって、タイルまたはＣＴｉｌｅによってカバーされていないすべての領域をパディングすることからなるすべてのフレーム部分が復号された後に、新しいステップを追加すること

からなる（例えば、図９のステップ９００の前の）復号処理の開始時に予備ステップを追加することによって実施することができる。

【０１８９】

一実施形態によれば、同じ空間位置にある複数のＣＴｉｌｅ、または重複しているＣＴｉｌｅを処理することが可能である。ＣＴｉｌｅ識別子ごとに、関連する復号されたＣＴｉｌｅバッファ（ＨＥＶＣのデコーデッドピクチャバッファ（ＤＰＢ）と均等物であるが、ここでは復号されたＣＴｉｌｅデータのみを含む）がある。所与のフレームに対して、各ＣＴｉｌｅは、関連する復号されたＣＴｉｌｅバッファで使用可能な時間データを使用して復号される。次に、第１の代替によれば、ＣＴｉｌｅのレンダリング順序は、バイストリーム内のＣＴｉｌｅ順序と同じ順序である。２番目の代替では、ＣＴｉｌｅは、ＸＰＳデータから決定できるレンダリング順序に関連付けられている。両方の代替について、各ＣＴｉｌｅの復号結果のサンプルは、ＣＴｉｌｅのレンダリング順序でレンダリングフレームバッファのフレームに入れられる（次いで、前のＣＴｉｌｅによって以前に順序付けられたサンプルをおそらく消去／マスキングする）。

【０１９０】

一実施形態によれば、ＣＴｉｌｅサンプルは、レンダリングフレームバッファのフレーム内でＣＴｉｌｅをレンダリングするときに適用される透明度のレベルを示すアルファチャネルをさらに含む。あるいは、サンプルがＣＴｉｌｅのどのサンプルがレンダリングフレームバッファのフレーム内でレンダリングされなければならないかを示すバイナリマスク値をさらに含む。

【０１９１】

一実施形態によれば、同じ位置にある複数のＣＴｉｌｅ、または重複しているＣＴｉｌｅを処理することが可能である場合、ＣＴｉｌｅ位置およびＣＴｉｌｅサイズの両方がＸＰＳにおいて指定されなければならず、それは、その文脈において一方を他方から推論することが不可能であるからである。

【０１９２】

任意の所与のポストフィルタリングアルゴリズム（例えば、デブロッキングフィルタ、サンプル適応オフセット、または適応ループフィルタ）の実施形態によれば、ＣＴｉｌｅ境界ポストフィルタリングフラグをＸＰＳで指定して、ポストフィルタリングアルゴリズムがＣＴｉｌｅに使用可能であるか否かを示すことができる。ＣＴｉｌｅ境界ポストフィル

タリングフラグ、例えば’ｕｓａｂｌｅ＿ｆｏｒ＿ｐｏｓｔ＿ｆｉｌｔｅｒｉｎｇ＿ｆｌａｇ’は、所与のポストフィルタリングアルゴリズムがレンダリングフレームバッファのレンダリングされたフレームのＣＴｉｌｅ境界に適用される可能性があることを示す（時間的復号を修正できるため、デコーデッドピクチャバッファには適用されない）。有利には、例えば、視覚品質を改善することを目的とする。フラグは、フレームレベル全体について、および／またはＣＴｉｌｅのそれぞれについて指定することができる。このフラグは、ポストフィルタリングされたときにアーチファクトを導入する傾向があることが知られているいくつかのエッジのフィルタリングを防止するために有用であり得る。例えば、フラグは、適応品質ストリーミングの文脈におけるＣＴｉｌｅシャッフリングに対して真であるが、ＣＴｉｌｅ境界が３６０°コンテンツの立方投射の２つの面の間にあり、面がそのエッジ上で隣接していない場合には偽である。ポストフィルタリングされるＣＴｉｌｅ境界は、エッジの２つの側がポストフィルタリングを適用することができることを指定しているもの、またはエッジがＨＥＶＣタイプのタイルとＣＴｉｌｅ許可ポストフィルタリングとの間にあるときのものである。

【０１９３】

代替の実施形態によれば、ＣＴｉｌｅ境界は、デコーデッドピクチャバッファ（ＤＰＢ）においてポストフィルタリングされてもよく、その実施形態では、復号が任意の復号構成において正しいことを保証するために、インター予測が使用される場合、予測に使用されるサンプルはポストフィルタリングされないサンプルである。したがって、境界拡張メカニズムは境界情報を使用してポストフィルタリングされたもの前の境界上の最後のサンプルに適用され、これは、境界拡張が、フィルタリングされていないサンプルに対して実行されることを意味する。

【０１９４】

代替の実施形態によれば、２つ以上のＣＴｉｌｅが同じＣＴｉｌｅ識別子を有しても良い。この実施形態では、ＣＴｉｌｅ識別子はＣＴｉｌｅ集合識別子となる。ＣＴｉｌｅ集合を形成するＣＴｉｌｅの設定は、適切に復号されるために、すべて一緒に、かつ同じ相対位置で保持されなければならない。

【０１９５】

これらの実施形態では、ＣＴｉｌｅ集合の位置およびサイズがＸＰＳから推論される。これは、ＣＴｉｌｅ集合に属するＣＴｉｌｅの設定の境界ボックスの位置およびサイズに対応する。したがって、ＸＰＳでは、ＣＴｉｌｅ集合識別子が１つまたは複数の位置およびサイズ（ＣＴｉｌｅ集合内の各ＣＴｉｌｅに対して１つ）に関連付けられる。

【０１９６】

これらの実施形態では、スライスセグメントヘッダ’ｃｔｂ＿ａｄｄｒ＿ｏｆｆｓｅｔ＿ｉｎｓｉｄｅ＿ｔｉｌｅ’１３０５情報が、’ｃｔｂ＿ａｄｄｒ＿ｏｆｆｓｅｔ＿ｉｎｓｉｄｅ＿ｔｉｌｅ＿ｓｅｔ’情報に置き換えられてもよい。’ｃｔｂ＿ａｄｄｒ＿ｏｆｆｓｅｔ＿ｉｎｓｉｄｅ＿ｔｉｌｅ＿ｓｅｔ’を用いると、どのＣＴｉｌｅがスライスセグメントに属しているのかを推定できるので、スライスセグメントの復号時に使用されるジオメトリを推定できる。

【０１９７】

そのような実施形態のうちの１つでは、ＣＴｉｌｅのセットの任意のサンプルを、時間的動き補償に使用することができる。動き補償がＣＴｉｌｅ集合の外側のサンプル値を使用する場合、このサンプル値はＣＴｉｌｅ集合のいずれか１つの空間的に最も近いサンプルの値に設定される（境界拡張を適用することと同等であるが、２つのＣＴｉｌｅによって共有されないＣＴｉｌｅ境界部分についてのみ）。ＣＴｉｌｅの外側の任意のサンプルが２つ以上の最も近いＣＴｉｌｅサンプルを有する場合、単純な規則を使用して、どのサンプルを使用すべきか、例えば、最小のラスタスキャン順序を有するサンプルを決定する。

【０１９８】

図１５は、ＣＴｉｌｅ識別子をシグナリングするための、図１３ａ、１３ｂ、および１３ｃに示される実施形態の代替の実施形態を示す。現在のブロックベースのコーデック、典

型的なＨＥＶＣでは、ＮＡＬユニットヘッダ１５０１が以下のフィールドを含む：
－０：（偽）に設定された１ビット；
－ＮＡＬユニットタイプを含む６ビット：（タイプ）；
－レイヤ識別子を含む６ビット:（ＬａｙｅｒＩＤ）、これはＨＥＶＣでは常にゼロに等しいが、スケーラブルＨＥＶＣ（ＳＨＶＣ）ではスケーラブルレイヤインデックスに対応する、もしくはたとえば、マルチビューＨＥＶＣ（ＭＶ－ＨＥＶＣ）ではビューインデックスに対応する。
－時間レイヤ識別子を示す３ビット:（ＴＩＤ）、これはＨＥＶＣでは時間的スケーラビリティのための時間レイヤインデックスに対応する。

【０１９９】

ＮＡＬユニットヘッダ１５０１に基づく一実施形態では、エンコーダがビデオシーケンスをフレーム部分に分割する。エンコーダは、各フレーム部分に対して１つの符号化またはスケーラビリティレイヤを使用する。これは、空間領域に基づくレイヤ符号化として見ることができる。エンコーダは、他の領域とは独立して各空間領域レイヤを符号化することができる。このような場合、各空間領域レイヤは、１つのＣＴｉｌｅに対応する。この特定のケースでは、符号化時の空間領域レイヤのすべてのスライスは、ｔｒｕｅに設定されているｃｔｉｌｅ＿ｆｌａｇを有する。主な違いは、各空間領域レイヤをさらにＨＥＶＣタイプのタイルに分割できることである。

【０２００】

エンコーダは、ＬａｙｅｒＩｄを用いて異なる空間領域レイヤをシグナリングする。これは、ＬａｙｅｒＩｄの値をＣＴｉｌｅの識別子に等しく設定する。その結果、ＣＴｉｌｅ識別子は、スライスセグメントヘッダでは必要ない。それは固定ビット長であるので、ＣＴｉｌｅ識別子の処理は、ビデオストリームのフレーム部分をシャフリングするときに単純なままである。

【０２０１】

エンコーダは、パラメータセットのうちの１つ、例えば、ＶＰＳにおいて、フレーム部分構成をシグナリングする。ＶＰＳは、空間領域レイヤの固有識別子を、前の実施形態で説明された構文に対応する構文で復号位置と関連付けることによって、各空間領域レイヤの復号位置を示す。

【０２０２】

エンコーダは、ビデオストリームの異なるレイヤ間の依存関係も記述する。そして、デコーダは、パラメータセットＮＡＬユニットに記述されたレイヤ間の依存関係の解析を通じて、他のレイヤとは独立に符号化された空間領域レイヤを決定する。

【０２０３】

エンコーダは、他の空間領域レイヤとは独立して、空間領域レイヤのサブセットをＣＴｉｌｅとして圧縮する。ある空間領域レイヤが前のフレーム（つまり同じＣＴｉｌｅ識別子を有する）における別の空間領域レイヤ（このレイヤのスライスは、ｆａｌｓｅに設定されたｃｔｉｌｅ＿ｆｌａｇを持つ）に従属する場合、エンコーダは、この従属レイヤからの参照フレームを現在のレイヤのデコーデッドピクチャバッファに追加する。アップサンプリングまたはダウンサンプリングフィルタは、参照フレームが現在のレイヤのサイズに等しいサイズを有するように、２つのレイヤのサイズが異なる場合に適用される。

【０２０４】

一実施形態によれば、ＬａｙｅｒＩｄは、ｃｔｉｌｅ＿ｆｌａｇを推論するためにも使用することができる：ＬａｙｅｒＩｄがゼロのとき、ＮＡＬユニットは、ＨＥＶＣタイプのタイルに属している。ＬａｙｅｒＩｄがゼロでない場合、ＮＡＬユニットはＬａｙｅｒＩｄと等しいＣＴｉｌｅ識別子を持つＣＴｉｌｅに属する。あるいは、ＬａｙｅｒＩｄの１ビットは、ｃｔｉｌｅ＿ｆｌａｇをシグナリングするために予約される。ＬａｙｅｒＩｄを使用してＣＴｉｌｅ識別子を転送する利点は、ＣＴｉｌｅを抽出するためにビットストリームを解析する複雑さが大幅に軽減されることである。

【０２０５】

別の実施形態では、空間領域スケーラビリティが、ＨＥＶＣにおける時間的スケーラビリティと同様に定義され、すなわち、異なるレイヤ識別子は他のスケーラビリティレイヤ（例えば、ＳＮＲ、解像度、マルチビュー）から時間空間領域を識別する。実際、このアプローチの利点は、空間領域スケーラビリティおよびＳＮＲまたは解像度スケーラビリティレイヤの両方を使用することが可能である、ということである。

【０２０６】

ＮＡＬユニットヘッダ１５０２は、現在フレーム部分識別子を示すランダムアクセス識別子（ＲＡＩＤ）で拡張される。ＬａｙｅｒＩｄセマンティクスは、ＨＥＶＣと同じままであり、すなわち、マルチビュー、ＳＮＲ、または解像度スケーラビリティレイヤを示す。

【０２０７】

エンコーダは、各空間領域レイヤの位置を、そのＲＡＩＤ値を、パラメータセットのうちの１つ、例えばＶＰＳ内の復号位置に関連付けることによって指定する。空間領域を符号化する各ＮＡＬユニット（ＳＰＳ、ＰＰＳおよびＶＣＬ　ＮＡＬユニットを含む）は、空間領域に対応するフレーム部分識別子（ＣＴｉｌｅ（ｓｅｔ）識別子）に等しいＲＡＩＤを有する。

【０２０８】

結果として、上述のマージ処理（ビデオビットストリームのセットからＣＴｉｌｅを抽出し、それらを新しいビデオビットストリームに結合することからなる）は、結合するビデオストリームに関連するフレーム部分構成からマージするために空間領域レイヤのＣＴｉｌｅ識別子を抽出する。そして、抽出した識別子の組に等しいＲＡＩＤ値を有するＮＡＬユニットを全て抽出する。

【０２０９】

２つのビデオシーケンスを結合するときの識別子の衝突のリスクを制限するために、エンコーダは、ＲＡＩＤ値をランダム値で設定する。これは、ビデオシーケンスが単一のフレーム部分を含むケースを含む。

【０２１０】

一実施形態によれば、ＲＡＩＤは、空間領域がＣＴｉｌｅであるかどうかを指定する（スライスセグメントヘッダ内のｃｔｉｌｅ＿ｆｌａｇのシグナリングを置き換える）：ＲＡＩＤがゼロのとき、ＮＡＬユニットはＨＥＶＣタイプのタイルに属する。ＲＡＩＤがゼロでない場合、ＮＡＬユニットはＲＡＩＤに等しい識別子を持つＣＴｉｌｅに属する。あるいは、ＲＡＩＤの１ビットはｃｔｉｌｅ＿ｆｌａｇをシグナリングするために確保される。代替の実施形態では、より多くのＣＴｉｌｅを可能にするために、ＲＡＩＤ識別子は１６ビットまたは２４ビットである。

【０２１１】

一実施形態によれば、ＣＴｉｌｅのシーケンスは、独立したビットストリームと見なされる。例えば、いくつかの実施形態では、同じ識別子を有するＣＴｉｌｅのシーケンス順序が、別の識別子を有するＣＴｉｌｅのシーケンス順序とは異なり得る（すなわち、ＧＯＰ構造は、２つのＣＴｉｌｅの間で異なり得る）。したがって、同じフレーム内の２つのＣＴｉｌｅは、異なるＮＡＬユニットタイプまたはＴＩＤを有することができる。

【０２１２】

別の実施形態では、ＸＰＳは、デコーダが誤差無しでＣＴｉｌｅを処理できる方法で、ＣＴｉｌｅ間の何らかの依存関係を記述する追加情報を含む。ＣＴｉｌｅの独立性は、各ＣＴｉｌｅのレベルではなく、ＣＴｉｌｅのセットのレベルで考えられることが分かった。この構成では、ＣＴｉｌｅ集合内の幾つかのＣＴｉｌｅが幾つかの依存関係を持つことがある。

【０２１３】

例えば、図１６ａは、ＣＴｉｌｅごとの依存関係リストを含むＸＰＳを示す。依存関係リストは、識別子１６００を有する所与のＣＴｉｌｅに依存するＣＴｉｌｅ識別子１６０１を提供する。所与のＣＴｉｌｅが別のＣＴｉｌｅとの依存関係を有するものとして示される場合、所与のＣＴｉｌｅが他のＣＴｉｌｅなしでは抽出できないことを意味する。

【０２１４】
図１６ｂは、ＣＴｉｌｅ依存関係の第１の例を示す。現在のフレーム１６０２内のＣＴｉｌｅ　＃１は、前の符号化フレーム１６０３から動き補償を実行するときにＣＴｉｌｅ　＃２からのサンプル値を使用し、ＣＴｉｌｅ　＃２は、ＣＴｉｌｅ　＃１からのサンプル値を使用する。このような例では、ＸＰＳは、識別子＃１を有するＣＴｉｌｅが識別子＃２を有するＣＴｉｌｅとの依存関係を有し、識別子＃２を有するＣＴｉｌｅが識別子＃１を有するＣＴｉｌｅとの依存関係を有することを示している。これらの相互依存関係のシグナリングは、ＣＴｉｌｅセットのシグナリングの代替である。
【０２１５】
図１６ｃは、ＣＴｉｌｅ依存関係の第２の例を示す。このような例では、ＣＴｉｌｅ　＃３が、フレーム１６０４および１６０７内のＣＴｉｌｅ　＃１および＃２に対する依存関係として、フレーム１６０５および１６０７内に存在する。このような例では、ＣＴｉｌｅ　＃１および＃２も抽出せずにＣＴｉｌｅ　＃３を抽出することはできない。しかし、ＣＴｉｌｅ＃１および＃２は、依存関係を有さず、単独で抽出することができる。実施形態によれば、このシナリオは例えば、フレーム１６０４～１６０６が時間的に順序付けられている場合に、様々なフレームレートでＣＴｉｌｅの抽出を容易にするために適用することができる。あるいは、例えば、フレーム１６０５がフレーム１６０４のリファインメントレイヤであり、フレーム１６０７がフレーム１６０６のリファインメントレイヤである場合、異なる品質レイヤのＣＴｉｌｅの抽出を容易にするために、スケーラブル符号化に使用することができる。
【０２１６】
いくつかの実施形態によれば、ＣＴｉｌｅは、連続するフレーム間で空間的位置またはサイズを変更することができる。
【０２１７】
実施形態によれば、’タイルパラメータセット’、ＴｉｌｅＰＳも導入される。ＴＰＳは、ＣＴｉｌｅのサブセットのＣＴｉｌｅパラメータ（移動および／またはサイズの変更のみ）を更新でき、例えば、’ｎｕｍ＿ｕｐｄａｔｅｄ＿ｔｉｌｅｓ’値を含み、次に’ｎｕｍ＿ｕｐｄａｔｅｄ＿ｔｉｌｅｓ’修正されたＣＴｉｌｅの新しいプロパティにタイル識別子を関連付ける。
【０２１８】
古典的には、動きベクトルは、符号化するブロックと一緒に配置されたブロックに対する参照画像内の予測ブロックの位置を与える。符号化する所与のブロックについて、最初のステップは、参照画像の中で、併置されたブロックを識別することである。併置されたブロックは、符号化されるブロックと同じ原点（左上の位置）および同じサイズを意味する、同じ位置を有する参照画像内のブロックとして定義される。次に、動きベクトルを、併置されたブロックの原点に適用して、予測ブロックの原点を決定する。
【０２１９】
ＣＴｉｌｅを考慮する場合、併置されたブロックの決定は、ＣＴｉｌｅ内で同じ位置を有し、もはやフレーム内ではないブロックを考慮するように適用される。ＣＴｉｌｅがシャッフルされたとき、フレーム内のその位置が、符号化時のフレーム内のその位置に対して復号時にフレーム内で修正されていることを意味する。しかしながら、予測が独立した復号を保証するためにＣＴｉｌｅ内に拘束されることを考慮すると、ＣＴｉｌｅ内の符号化ブロックに併置されたブロックに動きベクトルを適用することによって、正しい予測ブロックを依然として決定することができる。これは、ＣＴｉｌｅがフレーム毎にそのサイズ及び位置を保持する限り、真実である。ＣＴｉｌｅがフレーム内のその位置および／または２つの連続するフレーム間のそのサイズを変化させるときに、困難が生じる。この場合、エンコーダおよびデコーダは、予測ブロックを正確に決定するために動きベクトルが適用される参照フレーム内の併置されたブロックの位置を決定する方法に同意する。
【０２２０】
ＣＴｉｌｅが連続的に符号化されたフレーム間で位置またはサイズを変化させることがで

きる実施形態によれば、２つの連続するフレームにおけるＣＴｉｌｅの相対位置は、２つの異なるビットストリームにおいて同じでなくてもよい。図１７は、第１のビットストリームがビデオ監視１７００のフレームを含む、そのような実施形態の例を提供する。第１のフレーム１７０１には、所与のｃｔｉｌｅ＿ｉｄを有する移動注目領域のＣＴｉｌｅ１７０２を含むいくつかのｓｐｔａｉｌフレーム部分がある。別のフレーム１７０３では、所与のｃｔｉｌｅ＿ｉｄを有するＣＴｉｌｅが移動し、サイズ１７０４を変更している。第２のビットストリームは、１７００から抽出されたＣＴｉｌｅをアセンブルすることによって生成されたビデオ１７０５を含み、生成されたＣＴｉｌｅは均一な色（例えば、黒）を含む。第１のフレーム１７０６において、所与のｃｔｉｌｅ＿ｉｄ１７０２を有するＣＴｉｌｅは、第１のビットストリームから抽出され、フレーム１７０７の中心に置かれる。別のフレーム１７０８では、所与のｃｔｉｌｅ＿ｉｄ１７０４を有するＣＴｉｌｅが第１のビットストリームから抽出され、フレーム１７０９の中心に置かれている。第１のビットストリームでは、ＣＴｉｌｅ１７０４は、ＣＴｉｌｅ１７０１への時間的参照とともにインター予測を使用する。したがって、生成されたビデオ１７０５において、ＣＴｉｌｅ１７０９は、ＣＴｉｌｅ１７０７への時間的参照を伴うインター予測を使用する。ＣＴｉｌｅ１７０２と１７０４との間の相対空間位置は、１７０７と１７０９との間と同じではない。したがって、復号相対位置が何であれ、適切に復号されるために、インター予測モードが使用されるとき、符号化動きベクトルは、ＣＴｉｌｅ位置の変化（すなわち、連続するフレーム間の相対空間位置）を考慮に入れない。

【０２２１】

第１の代替によれば、動きベクトルは、２つの連続して符号化されたフレームにおけるＣＴｉｌｅの所定の基準点が同じ空間位置（例えば、上部左、上部右、下部左、下部右、中央上部、中央下部、中央左、中央右または中央）にあるかのように計算される。したがって、符号化動きベクトルは、フレームリファレンシャル内のブロックの動きベクトルからＣＴｉｌｅの基準点間の動きベクトルを引いたものに対応し、フレームリファレンシャル内の動きベクトルも対応し、したがって、ＣＴｉｌｅに対するリファレンシャル内の動きベクトルが得られる。その結果、ＣＴｉｌｅは、後の空間的変化とは独立して復号可能である。

【０２２２】

図１７ｂは、参照フレーム１７１３内に同じｃｔｉｌｅ＿ｉｄを有する参照ＣＴｉｌｅ１７１２を使用して、フレーム１７１１内に符号化されたＣＴｉｌｅ１７１０の場合を示す。ブロック１７１４は、フレーム１７１６内の動きベクトルと、所定の基準点１７１７（その例では、所定の基準点はＣＴｉｌｅの左上隅である）間の動きベクトルと、の間の差に対応する動きベクトル１７１５を使用して符号化される。図１７ｂはまた、ＣＴｉｌｅの相対時間復号位置が符号化位置と同じでない場合でも、復号フレーム１７２０内の動きベクトルを得るために、復号フレーム１７１９内の所定の基準点間の動きベクトルにそれを加えることによって、ブロック１７１８を復号するときに、符号化ベクトル１７１５が依然として有効であることを示す。

【０２２３】

第２の代替によれば、動きベクトルは、ＣＴｉｌｅの所与の点が２つの連続して符号化されたフレーム内の同じ空間位置にあるかのように計算される。所与の点は、所定の点（たとえば、上－左、上－右、下－左、下－右、中－上、中－下、中－左、中－右、または中央）のリスト内のインデックスとして、ＣＴｉｌｅ符号化データでシグナリングされる。

【０２２４】

第３の代替によれば、固定点（または代替的にシグナリングされた点）が考慮され、動きベクトルがＣＴｉｌｅ符号化データにおいて符号化される。ＣＴｉｌｅ符号化データにおいて符号化された動きベクトルは、２つの連続して符号化されたフレーム内のＣＴｉｌｅの固定（またはシグナリングされた）点が同じ空間位置にあることを考慮すると、ＣＴｉｌｅ内の時間予測に関連するインター動きベクトルのそれぞれに追加される動きベクトルを提供する。それは、エンコーダが動きベクトルの符号化コストを低減することを可能に

する。例えば、エンコーダは、ＣＴｉｌｅの動き補償ブロックの平均動きベクトルを選択することができる。例えば図１７ｂを見ると、この平均動きベクトルは、ベクトル１７１５に差し引かれる。この差し引きの結果を、符号化対象の動きベクトルとする。結果に関して同等である代替は、固定小数点または固定小数点インデックスおよび動きベクトルの代わりに（またはそれに加えて）、符号化されたＣＴｉｌｅ内の基準点の（サブ）ピクセル位置を提供することである。

【０２２５】

第４の代替によれば、固定点（又は代替的にシグナリングされた点）が考慮される。ＣＴｉｌｅ符号化データには、動きベクトルフィールドのパラメータが符号化される。動きフィールドは、２つの連続的に符号化されたフレーム内のＣＴｉｌｅの固定（またはシグナリングされた）点が同じ空間位置にあることを考慮するとき、ＣＴｉｌｅ内の時間予測に関連するインター動きベクトルのそれぞれに追加される動きベクトルを決定することを可能にする。例えば、エンコーダは、ＣＴｉｌｅ内のブロックの動きベクトルを推定することができ、そしてそれらの残差を最小化し且つそれらの符号化のコストを低減するために、それらの予測を最小化する動きベクトルフィールドを推定することができる。次に、インター符号化ブロックの各動き補償ベクトルは、動きベクトルフィールドパラメータから計算された動きベクトルを動き補償ベクトルに減算した結果である（例えば、１７１５）。

【０２２６】

実施形態によれば、インター予測モードは、２つ以上の以前に符号化された参照フレームを参照することができる。その実施形態では、前述の実施形態を拡張して、ＣＴｉｌｅの固定（またはシグナリングされた点）が符号化フレームおよびすべての参照フレーム内で整列されることを考慮することができる。

【０２２７】

動きベクトルまたは動きベクトルフィールドがシグナリングされる実施形態では、複数フレームへの拡張が２つの代替方法で行うことができる。

－参照フレームの数と同じ数の動きベクトル（または動きベクトルフィールドパラメータ）をシグナリングすることによって、または

－符号化フレームとの時間的な差に従って、各参照フレームについて１つの動きベクトル（または動きベクトルフィールド）を導出するために使用される１つの動きベクトル（または動きベクトルフィールド）’ｘ’だけをシグナリングすることによって

のいずれか

【０２２８】

例えば、線形スケーリングが使用される。参照フレームの時間的な位置が’ｔ－ｓ’（ここで、’ｓ’は、フレーム間の一定の時間的なサンプリング周期であり、’ｔ’は時間である）であり、符号化フレームの時間的な位置が’ｔ’である場合、使用されるスケーリングファクタは’（ｔ）／ｓ－（ｔ－ｓ）／ｓ＝１’であるが、参照フレームの時間的な位置が’ｔ＋２ｓ’である場合、使用されるスケーリングファクタは’（ｔ）／ｓ－（ｔ＋２ｓ）／ｓ＝－２’である。スケーリングファクターは、各参照フレームの動きベクトルを計算するために適用される。例えば、図１７ｂを見ると、参照フレーム１７１３が’ｔ＋２ｓ’である場合、動きベクトル１７１５は－２＊’ｘ’減算される。減算の結果’ｙ’は、符号化される動きベクトルの値である（例えば、符号化モードがＨＥＶＣのインター予測モードである場合、’ｙ’は、動きベクトル予測インデックスを使用して予測される動きベクトルである）。言い換えれば、デコーダ側では、動きベクトル’ｙ’は、動き補償されたブロックに対して復号され、次に、それが－２＊’ｘ’に加算されてベクトル１７１５が得られる。さらに、動きベクトル１７１７を加算して、フレームレベルの動きベクトルを得る。

【０２２９】

図１２は、前の説明で述べたように、ビットストリームがより高いレベルの記述フォーマットにカプセル化されているときの、エンコーダ側のカプセル化ステップ９０６または１１０５の詳細を提供する。

【０２３０】

好ましい実施形態では、ＣＴｉｌｅを有するビデオビットストリームがＩＳＯ　Ｂａｓｅ　Ｍｅｄｉａ　Ｆｉｌｅ　Ｆｏｒｍａｔ（ＩＳＯＢＭＦＦ、ＩＳＯ／ＩＥＣ１４４９６－１２および１４４９６－１５）に従ってカプセル化される。図１２に関連する以下の説明では、”フレーム”に対応する単語”サンプル”、すなわち、ＩＳＯＢＭＦＦについて定義されるような、符号化ピクチャに対応するビデオビットストリームからのＮＡＬユニットのセットに対応する。

【０２３１】

カプセル化は、ＩＳＯＢＭＦＦまたはｍｐ４ライタによって処理される。このライターは、ＮＡＬユニットヘッダのパーサを含む。ＮＡＬＵタイプ、識別子、および対応する圧縮データを抽出することができる。典型的には、抽出されたＮＡＬＵデータがカプセル化されたファイルのメディアデータコンテナ：’ｍｄａｔ’ボックスに配置される。ＮＡＬユニットの記述のためのメタデータは、メインの’ｍｏｏｖ’ボックスの下のボックスの構造化された階層に配置される。ビデオトラックにカプセル化された１つのビデオビットストリームは、そのサブボックスを有する’ｔｒａｋ’ボックスによって記述される。

【０２３２】

分割されたビデオフレームについては、ビデオの予測される使用に応じて、異なる可能なカプセル化が存在する。この使用は、ｍｐ４ライタアプリケーションにおいてハードコーディングすることができ、あるいは、例えば初期化ステップ１２００において、あるユーザまたは別のプログラムによって入力パラメータとして提供することができる。一実施形態では、１つのフレーム部分またはフレーム部分の所与のセットを１つのビデオトラックにカプセル化し、したがってマルチトラックカプセル化につなげることが便利であり得る。

【０２３３】

ＩＳＯＢＭＦＦライタの初期化が行われると、エンコーダは、ステップ１２０１において、ＮＡＬＵタイプ、特にパラメータセット（ＸＰＳ）に対応するものを読み出すことによって、ビデオビットストリームの構文解析を開始する。既に上述したように、パラメータセットは、ビデオビットストリームの符号化構成に関する高レベルの一般的な情報を提供する特定のＮＡＬユニットである。

【０２３４】

これらのパラメータセットの構文解析から、ｍｐ４ライタは、テスト１２０２において、ビデオビットストリームがフレーム部分を含むかどうかを決定することができる（例えば、パラメータセットの１つにＴＰＳまたは特定の分割構成が存在するかどうか）。フレーム部分が存在する場合、ｍｐ４ライタは、同じテスト１２０２で、これらが”制約タイル”、すなわちＣＴｉｌｅであるかどうかを判定する。ビットストリームがフレーム部分を含まないか、ＣＴｉｌｅを含まない場合、テスト１２０２は偽であり、ビデオビットストリームはステップ１２０３で１つのビデオトラックとしてカプセル化される。

【０２３５】

ＴＰＳ（Ｔｉｌｉｎｇ　Ｐａｒａｍｅｔｅｒ　Ｓｅｔ）は、パラメータセット情報の１つのＮＡＬＵと見なされ、デコーダ設定を提供するメタデータに埋め込むか、ＤｅｃｏｄｅｒＣｏｎｆｉｇｕｒａｔｉｏｎＲｅｃｏｒｄボックスのようなセットアップ情報を提供するメタデータに埋め込むことができる。この情報は、’ｓｔｓｄ’ボックスなどのサンプル記述専用のボックスの１つにある。通常、一部のコーデック固有のサンプルエントリにある。

【０２３６】

あるいは、本発明の一実施形態によれば、ＴＰＳは、ビデオデータのためのＮＡＬユニット（ＶＣＬ　ＮＡＬＵ）として処理され、’ｍｄａｔ’ボックスに１つのサンプルデータとして記憶されることができる。それはまた、サンプルエントリおよびサンプルデータの両方に存在してもよい。フレーム部分構成がビデオシーケンスに沿って変化する場合、それをサンプル記述レベル（サンプルエントリ）よりもサンプルレベル（サンプルデータ）に記憶する方が便利である。

【０２３７】

フレーム部分分割構成の変更が受信側でデコーダをリセットすることを必要とする場合、

ＩＳＯＢＭＦＦライタは、ビデオビットストリームからのＴＰＳおよびＣＴｉｌｅ関連情報をサンプルエントリに格納することが好ましい。このデコーダリセットにより、ファイルを受信または消費するデバイスは、新しい分割構成を考慮に入れることができる。新しい分割構成は例えば、処理するデータの量（すなわちレベル）をサポートするための符号化ツール（すなわち、プロファイル）の指示を含むことがある。分割構成のプロファイルとレベルの値、またはその他のパラメータによっては、デバイスが新しい分割構成をサポートする場合としない場合とがある。サポートしない場合、デバイスは、伝送に適応したり、利用可能な場合はビデオの代替バージョンを選択したりすることがある。新しい分割構成がサポートされる場合、デバイスは、ファイルの復号とレンダリングを続行する。

【０２３８】

ＲＯＩへの空間アクセスが必要とされる場合、テスト１２０４の後のブランチ”ｙｅｓ”において、ＩＳＯＢＭＦＦライタは、ユースケースに応じて異なるカプセル化ストラテジを有することができる。空間アクセスは、例えば、ＲＯＩまたはパーティションベースのディスプレイ（すなわち、フレーム部分のセットまたはフレーム部分または部分またはＲＯＩに対応するデータの部分のみの抽出および復号）、またはＲＯＩまたは部分ベースのストリーミング（すなわち、フレーム部分のセットまたは部分またはＲＯＩに対応するデータおよびメタデータの部分のみを送信する）を意味する。予測されるユースケースが、真であるテスト１２０５に対応するローカルディスプレイのための記憶である場合（ブランチ”ｙｅｓ”）、分割されたビデオビットストリームを１つのトラックに記憶するが、空間アクセスが必要とされるＲＯＩまたはフレーム部分またはフレーム部分のセットへのＮＡＬＵマッピングを含むことが好都合であり得る。ＮＡＬＵマッピングは、ステップ１２０６で生成される。これは、ＩＳＯＢＭＦＦライタについて、ビデオビットストリームの各ＮＡＬＵについて、所与のＣＴｉｌｅに関連するか、または同じランダムアクセス識別子（図１５のＲＡＩＤ基準１５０２）を有する（すなわち、選択可能かつ復号可能なフレーム部分または空間領域に対応する）ＮＡＬユニットをリストすることからなる。リスティングを実行できるようにするために、ＩＳＯＢＭＦＦライタのＮＡＬＵパーサモジュールは、ビットストリーム生成の実施形態に応じて、ステップ９０３において空間アクセスが必要とされるＲＯＩまたはフレーム部分またはフレーム部分のセットに割り当てられた識別子の値をチェックする。

【０２３９】

ビットストリームがＮＡＬＵヘッダに、空間アクセスが必要なフレーム部分のセットまたはフレーム部分またはＲＯＩについて固有の識別子を含まない場合、ＩＳＯＢＭＦＦライタは、スライスヘッダパーサを要求して、９０３で割り当てられたＣＴｉｌｅの識別子の値を取得し、例えば図１３ａの１３０１で参照されているｃｔｉｌｅ＿ｕｎｉｑｕｅ＿ｉｄｅｎｔｉｆｉｅｒ）。

【０２４０】

次に、ＮＡＬＵＭａｐＥｎｔｒｙ構造’ｎａｌｍ’が、ステップ１２０６において、’ｔｒａｋ’ボックス階層下のボックスとして作成され、ＮＡＬＵのリストおよびフレーム部分またはフレーム部分のセットへのそれらのマッピングが格納される。フレーム部分またはフレーム部分のセットごとに、タイプ’ｔｒｉｆ’のＳａｍｐｌｅＧｒｏｕｐＤｅｓｃｒｉｐｔｉｏｎＢｏｘは、フレーム部分またはフレーム部分のセットごとに、フレーム部分またはフレーム部分のセットの記述を提供し、たとえば、ＩＳＯ／ＩＥＣ１４４９６－１５からのＴｉｌｅＲｅｇｉｏｎＧｒｏｕｐＥｎｔｒｙのパラメータを提供する。フレーム部分識別子’ｔｒｉｆ’のｇｒｏｕｐ＿ＩＤ値は、カプセル化するＣＴｉｌｅの識別子の値に設定される。

【０２４１】

次に、ステップ１２０３で、すべてのフレーム部分またはフレーム部分のセットのデータが単一トラックとしてカプセル化される。テスト１２０８が真（ブランチ”ｙｅｓ”）であることに対応して、ユースケースがストリーミングである場合、テスト１２０８が真である場合、ビデオ内の空間アクセスレベルに対応する各フレーム部分またはフレーム部分の

セットを専用トラックに分割することが便利であり得、単一トラックカプセル化が行われる。

【０２４２】

ストリーミングユースケースでは、フレーム部分またはフレーム部分のセットごとに、ステップ１２０９において、ＮＡＬＵマッピングに関するフレーム部分記述が生成される。フレーム部分またはフレーム部分のセットの数は、ＴＰＳを構文解析することによって決定することができる。’ｔｒｉｆ’サンプルグループが使用され、ステップ１２１０で生成されたフレーム部分トラック（１つのフレーム部分またはフレーム部分のセットに関連するデータをカプセル化するためのトラック）当たり１つのフレーム部分またはフレーム部分の１セットがあるので、デフォルトサンプルグループでさえ使用され得る。次に、すべてのサンプルは、ＩＳＯ／ＩＥＣ１４４９６－１５に従ってフレーム部分記述子’ｔｒｉｆ’である同じサンプルグループ記述にマッピングされる。フレーム部分記述子’ｔｒｉｆ’のｇｒｏｕｐ＿ＩＤ値は、カプセル化するＣＴｉｌｅ（または存在する場合はＲＡＩＤ）の識別子の値に設定される。

【０２４３】

次に、ステップ１２１０で、各フレーム部分またはフレーム部分のセットが、それ自体のトラック、フレーム部分トラックに挿入される。フレーム部分トラックは、サンプルが実際にビデオの空間部分であることを示し、カプセル化すべきフレーム部分またはフレーム部分セットが残っていない場合には、ステップ１２１２で作成されたフレーム部分ベーストラックを参照する特定のサンプルエントリを含む、テスト１２１１。このフレーム部分ベーストラックは、タイリングパラメータセット（ＴＰＳ）を含むパラメータセットに対応する特定のＮＡＬユニットを含む。フレーム部分ベーストラックは、特定のトラック参照タイプで順番に各フレーム部分トラックを参照し、フレーム部分またはフレーム部分のセットの任意の選択の暗黙の再構成を可能にする。ステップ１２１２は、エクストラクタと呼ばれるＮＡＬユニットが１つまたは複数のフレーム部分トラックから明示的な再構成を提供する複合トラックに置き換えることができる。

【０２４４】

エクストラクタは次に、フレーム部分またはフレーム部分のセットの所与の識別子を指すエクストラクタを単に有することによって、通常、対応するＣＴｉｌｅ（または、ある場合にはＲＡＩＤ）の識別子を参照することによって、元のフレーム部分とは異なるフレーム部分またはフレーム部分のセットの任意の構成を、複合トラックの各サンプルについて、可能にする。

【０２４５】

ステップ１２１２において複合トラックを使用する場合、ステップ１２１０におけるフレーム部分トラックは実際には復号可能なフレーム部分トラックであってもよく、これは、それぞれが（１２０９において生成された）フレーム部分記述およびパラメータセットを含むことを意味する。各フレーム部分トラックにおけるＴＰＳの存在は、エクストラクタが異なるように再結合することができるので、オプションである。次に、サンプル記述は、使用中のコーデックに準拠するサンプルエントリを示すことができる。例えば、ＨＥＶＣが使用中の場合は’ｈｖｃ１’または’ｈｖｃ２’、ＡＶＣ（Ａｄｖａｎｃｅｄ Ｖｉｄｅｏ Ｃｏｄｉｎｇ）が使用中の場合は’ａｖｃ１’または’ａｖｃ２’、使用中のビデオコーダを明確に識別するために予約された４文字コーダである。

【０２４６】

図１８は、本発明の１つまたは複数の実施形態を実施するためのコンピューティングデバイス１８００の概略ブロック図である。コンピューティングデバイス１８００は、マイクロコンピュータ、ワークステーション、またはライトポータブルデバイスなどのデバイスとすることができる。コンピューティングデバイス１８００は、

－CPUで示されるマイクロプロセッサなどの中央処理ユニット１８０１；

－本発明の実施形態の方法の実行可能コード、ならびに本発明の実施形態による方法を実施するために必要な変数およびパラメータを記録するように適合されたレジスタを格納す

るための、ＲＡＭで示されるランダムアクセスメモリ１８０２であり、そのメモリ容量は例えば、拡張ポートに接続された任意のＲＡＭによって拡張することができる；
－本発明の実施形態を実現するためのコンピュータプログラムを記憶するためのＲＯＭで示される読み出し専用メモリ１８０３；
－ネットワークインターフェース１８０４は、典型的には処理されるデジタルデータが送受信される通信ネットワークに接続される。ネットワークインターフェース１８０４は、単一のネットワークインターフェースであってもよいし、異なるネットワークインターフェースのセット（例えば、有線および無線インターフェース、または異なる種類の有線または無線インターフェース）から構成されてもよい。データパケットは、送信のためにネットワークインターフェースに書き込まれるか、またはＣＰＵ１８０１内で実行されているソフトウェアアプリケーションの制御の下で受信のためにネットワークインターフェースから読み出される；
－ユーザインターフェース１８０５は、ユーザからの入力を受信するため、またはユーザに情報を表示するために使用されてもよい；
－ＨＤと示されるハードディスク１８０６は、大容量記憶装置として提供されてもよい；
－入出力モジュール１８０７は、ビデオソースまたはディスプレイなどの外部装置との間でデータを送受信するために使用することができる
に接続された通信バスを備える。

【０２４７】

実行可能コードは、読み出し専用メモリ１８０３、ハードディスク１８０６、または例えばディスクのようなリムーバブルデジタル媒体のいずれかに格納することができる。変形例によれば、プログラムの実行可能コードは、実行前に、ハードディスク１８０６などの、通信装置１８００の記憶手段の１つに記憶されるために、ネットワークインターフェース１８０４を介して、通信ネットワークの手段によって受信することができる。

【０２４８】

中央処理ユニット１８０１は、本発明の実施形態によるプログラムまたは複数のプログラムのソフトウェアコードの一部または命令の実行を制御し、指示するように適合され、命令は前述の記憶手段の１つに格納されている。電源投入後、ＣＰＵ１８０１は例えば、プログラムＲＯＭ１８０３またはハードディスク（ＨＤ）１８０６からそれらの命令がロードされた後に、ソフトウェアアプリケーションに関するメインＲＡＭメモリ１８０２からの命令を実行することができる。このようなソフトウェアアプリケーションは、ＣＰＵ１８０１によって実行されると、本発明のフローチャートの各ステップを実行する。

【０２４９】

本発明のアルゴリズムの任意のステップは、ＰＣ（”パーソナルコンピュータ”）、ＤＳＰ（”デジタルシグナルプロセッサ”）、またはマイクロコントローラなどのプログラマブルコンピューティングマシンによる命令またはプログラムのセットの実行によってソフトウェアで実施することができ、あるいは、ＦＰＧＡ（”フィールドプログラマブルゲートアレイ”）またはＡＳＩＣ（”特定用途向け集積回路”）などのマシンまたは専用コンポーネントによってハードウェアで実施することができる。

【０２５０】

以上、特定の実施形態を参照して本発明を説明したが、本発明は特定の実施形態に限定されるものではなく、本発明の範囲内における変更は当業者には明らかであろう。

【０２５１】

多くのさらなる変更および変形は、単に例として与えられ、添付の特許請求の範囲によってのみ決定される本発明の範囲を限定することを意図しない、前述の例示的な実施形態を参照することにより、当業者に示唆されるのであろう。特に、様々な実施形態からの異なる特徴は、適宜、交換されてもよい。

【０２５２】

上述した本発明の各実施形態は単独で実施してもよいし、複数の実施形態の組み合わせとして実施してもよい。また、様々な実施形態からの特徴は、必要な場合、または単一の実

施形態における個々の実施形態からの要素または特徴の組み合わせが有益である場合に組み合わせることができる。

【０２５３】

本明細書に開示される各特徴（添付の特許請求の範囲、要約および図面を含む）は、明示的に別段の記載がない限り、同一の、同等の、または類似の目的を果たす代替的特徴に置き換えることができる。したがって、特に断らない限り、開示される各特徴は、同等または同様の特徴の一般的なシリーズの一例にすぎない。

【０２５４】

特許請求の範囲において、単語”有する（comprising）”は、他の要素又はステップを排除するものではなく、不定冠詞「ａ」又は「ａｎ」は複数を排除するものではない。異なる特徴が相互に異なる従属請求項に記載されているという単なる事実は、これらの特徴の組合せが有利に使用されることができないことを示すものではない。

【図面】

【図１】

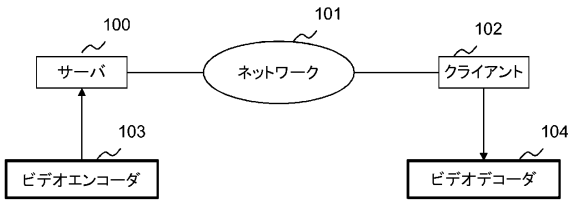


Fig. 1

【図２】

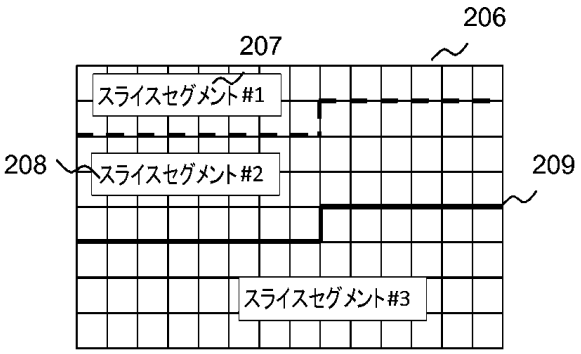


Fig. 2

【図３】

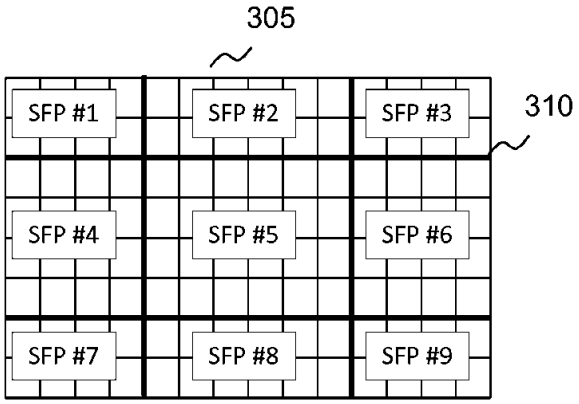


Fig. 3

【図４】

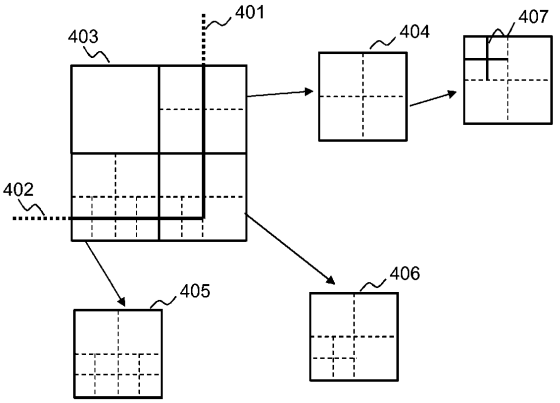


Fig. 4

【図５】

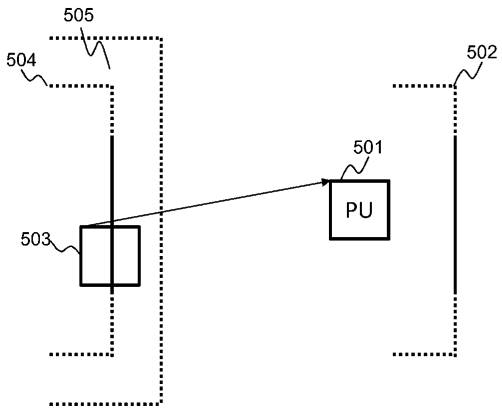


Fig. 5

【図６】

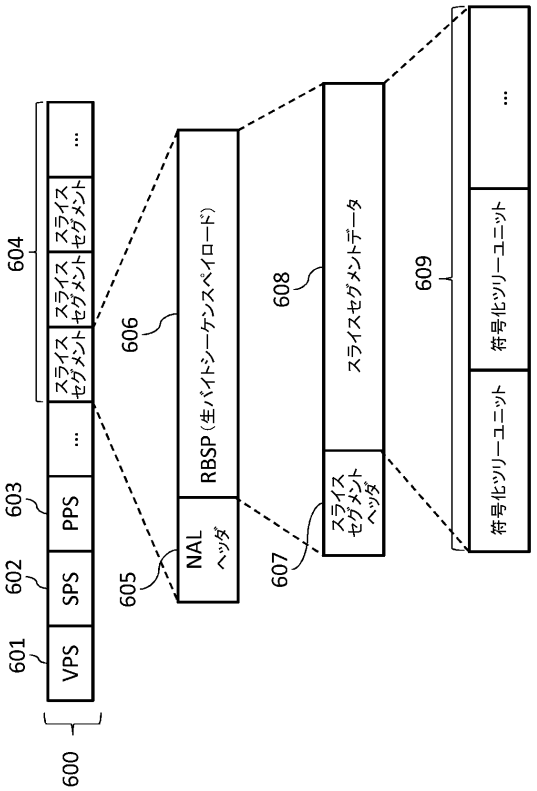


Fig. 6

【図７】

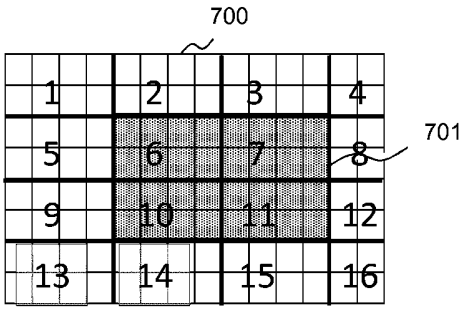


Fig. 7

【図８ａ】

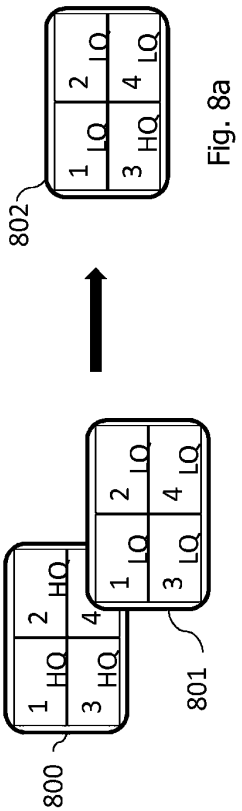


Fig. 8a

【図８ｂ】

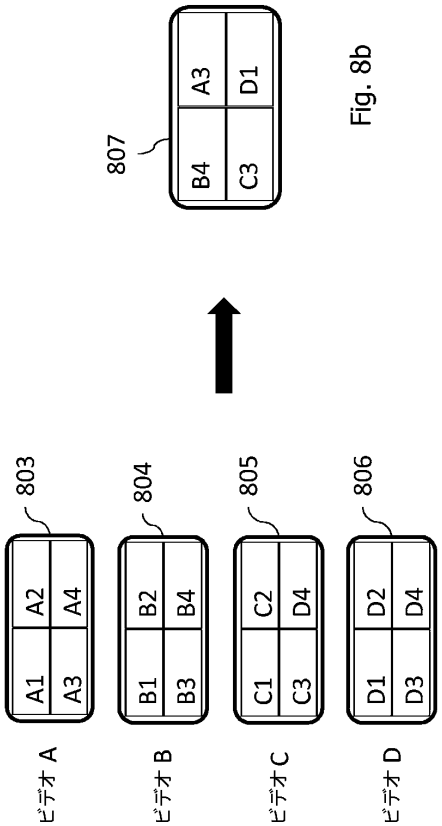


Fig. 8b

【図９】

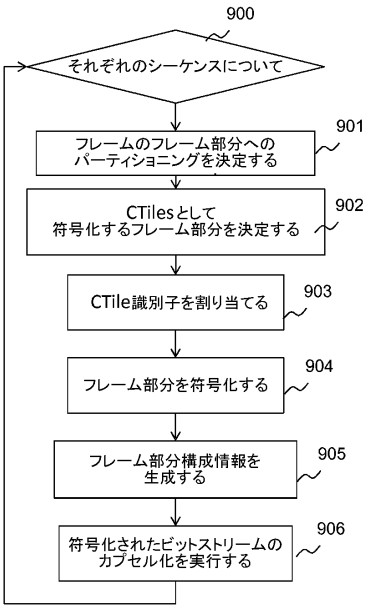


Fig. 9

【図１０】

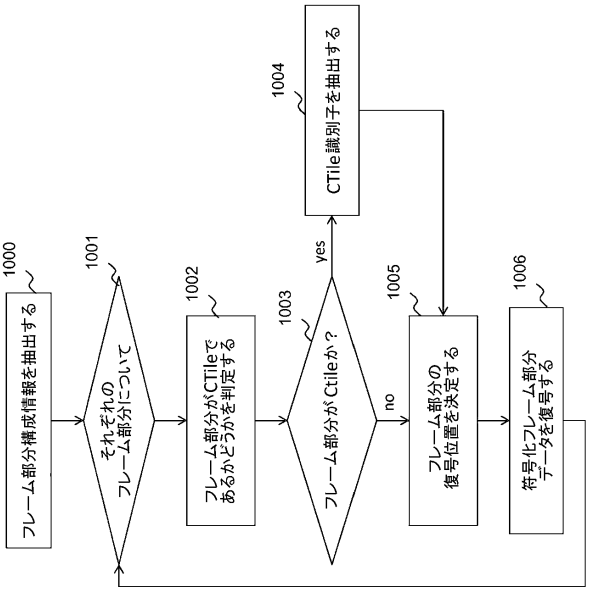


Fig. 10

【図１１】

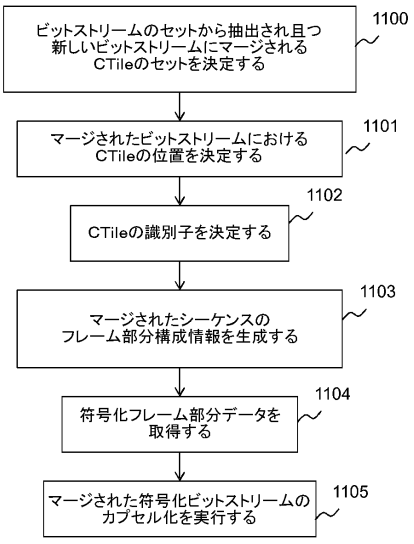


Fig. 11

【図１２】

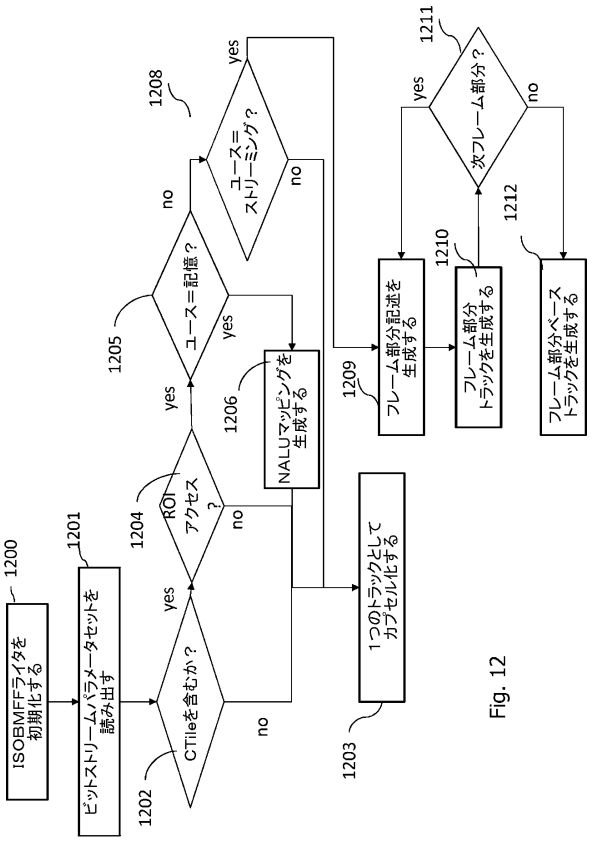


Fig. 12

【図１３ａ】

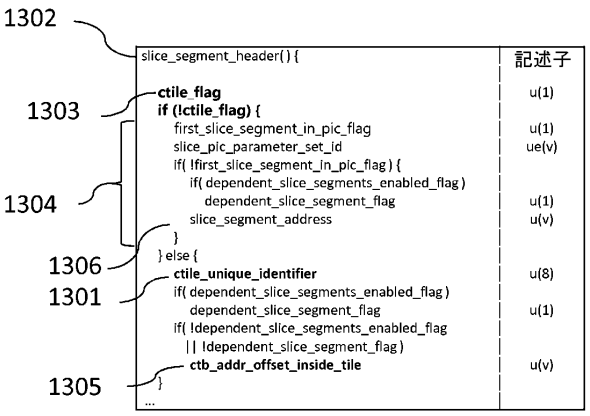


Fig. 13a

【図１３ｂ】

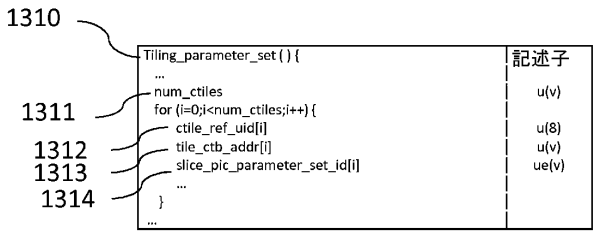


Fig. 13b

【図１３ｃ】

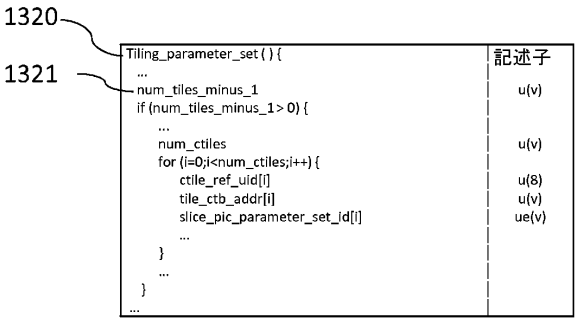


Fig. 13c

【図１４】

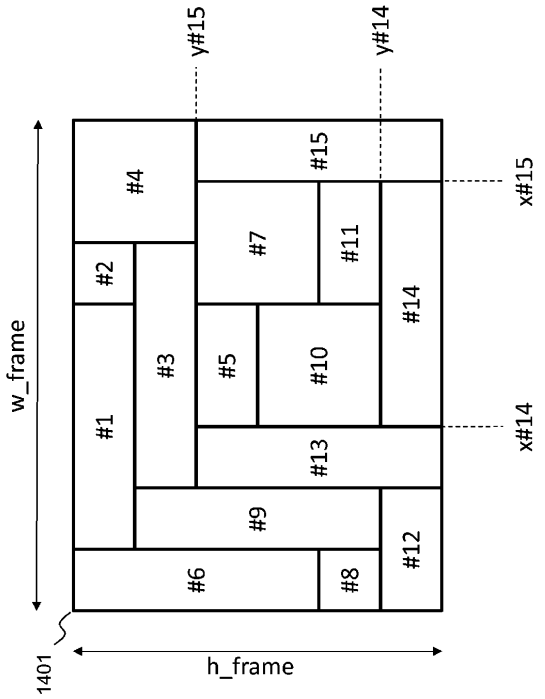


Fig. 14

【図１５】

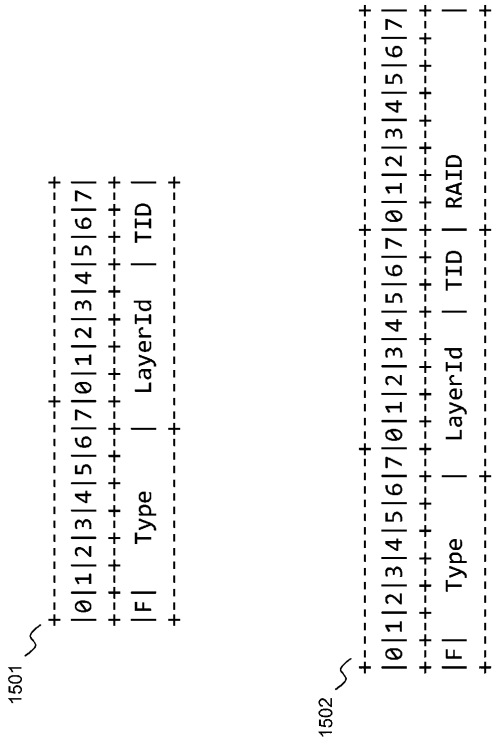


Fig. 15

【図１６ａ】

```
Tiling_parameter_set ( ) {                                  記述子
  ...
  num_tiles_minus_1                                         u(v)
  if (num_tiles_minus_1 > 0) {
      ...
      num_ctiles                                            u(v)
      for (i=0;i<num_ctiles;i++) {
        ctile_ref_uid[i]                                    u(8)
        ...
        num_ctile_dependencies                              u(v)
        for (j=0;j<num_ctile_dependencies;j++) {
          ctile_dep_uid[j]                                  u(8)
        }
        ...
      }
      ...
  }
...
```

1600
1601

Fig. 16a

【図１６ｂ】

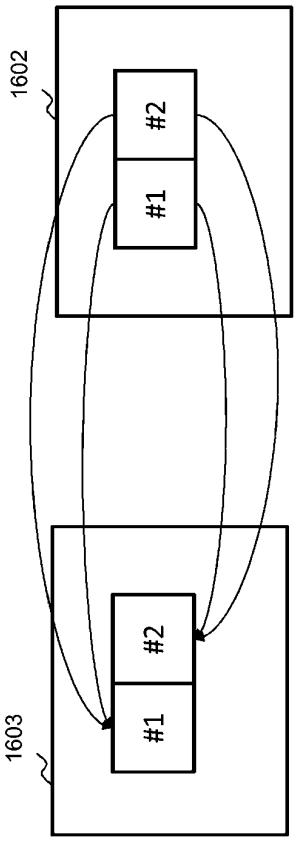


Fig. 16b

【図１６ｃ】

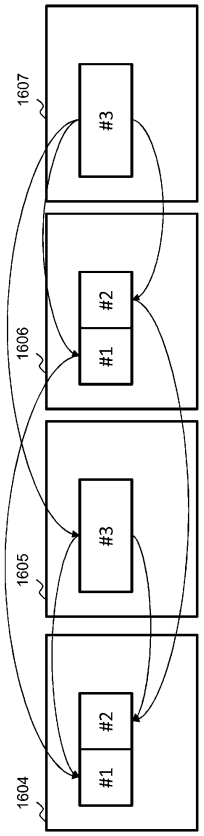


Fig. 16c

【図１７ａ】

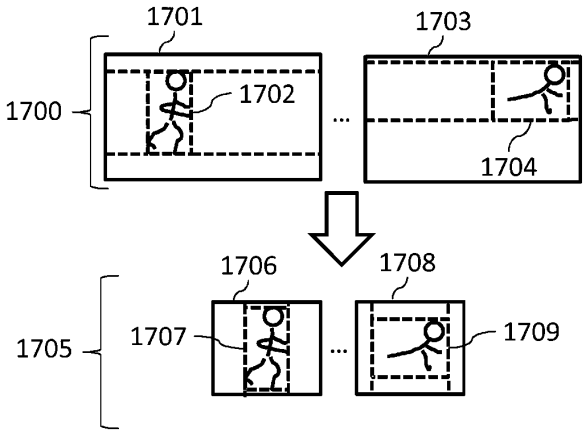


Fig. 17a

【図１７ｂ】

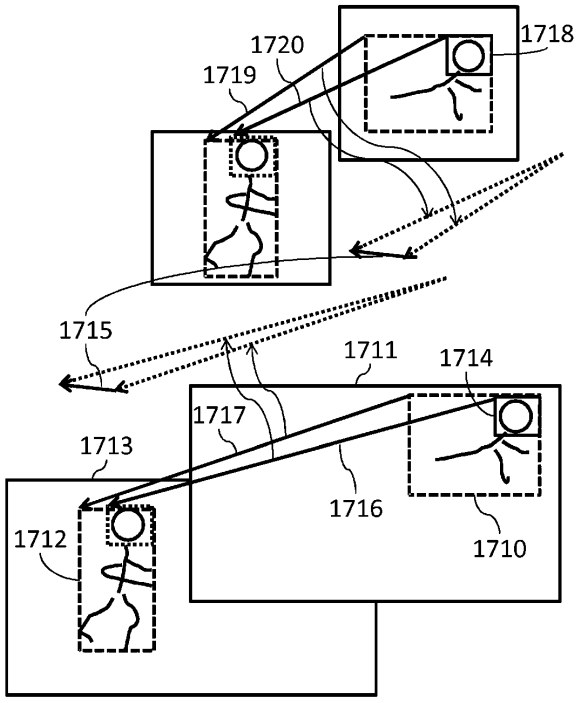


Fig. 17b

【図１８】

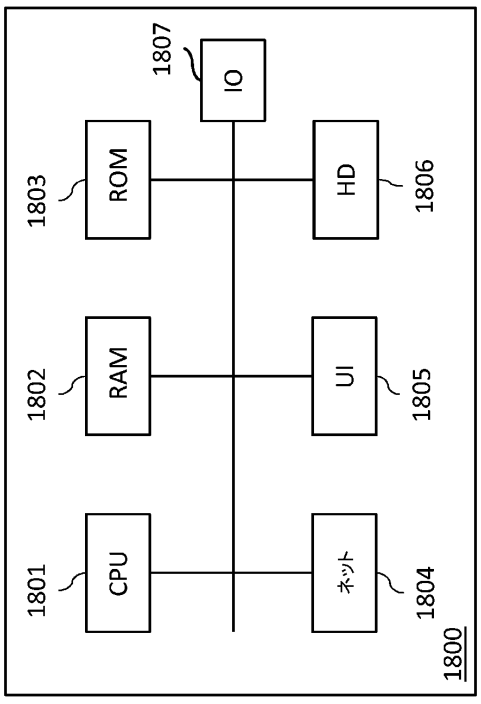


Fig. 18

フロントページの続き

ル キヤノン リサーチ センター フランス エス，エー，エス 内
(72)発明者 ドゥヌアル， フランク
フランス国 レンヌ－アタラント， セデックス セッソン－セヴィニエ ３５５１７， リュ ドゥラ トゥッシュランベール キヤノン リサーチ センター フランス エス，エー，エス 内
(72)発明者 マゼ， フレデリック
フランス国 レンヌ－アタラント， セデックス セッソン－セヴィニエ ３５５１７， リュ ドゥラ トゥッシュランベール キヤノン リサーチ センター フランス エス，エー，エス 内

審査官 鉢呂 健

(56)参考文献
特表２０１５－５２６００６（ＪＰ，Ａ）
特開２０１６－０４６７７８（ＪＰ，Ａ）
国際公開第２０１９／１４５２６２（WO，Ａ１）
WANG, Ye-Kui et al.，Tile groups，Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 8th Meeting: San Jose, CA, USA, 1-10 February, 2012, [JCTVC-H0520r1]，JCTVC-H0520 (version 2)，ITU-T，2012年01月31日，＜URL:http://phenix.it-sudparis.eu/jct/doc_end_user/documents/8_San%20Jose/wg11/JCTVC-H0520-v2.zip＞: JCTVC-H0520r1.doc: pp.1-8

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｈ０４Ｎ １９／００－１９／９８