

【公報種別】特許法第17条の2の規定による補正の掲載

【部門区分】第7部門第3区分

【発行日】平成26年10月30日(2014.10.30)

【公表番号】特表2013-543317(P2013-543317A)

【公表日】平成25年11月28日(2013.11.28)

【年通号数】公開・登録公報2013-064

【出願番号】特願2013-531941(P2013-531941)

【国際特許分類】

H 0 4 N 19/60 (2014.01)

H 0 4 N 19/50 (2014.01)

【F I】

H 0 4 N 7/133 Z

H 0 4 N 7/137 Z

【誤訳訂正書】

【提出日】平成26年9月9日(2014.9.9)

【誤訳訂正1】

【訂正対象書類名】明細書

【訂正対象項目名】全文

【訂正方法】変更

【訂正の内容】

【発明の詳細な説明】

【発明の名称】ジョイントコンテキストモデルを使用した係数のエントロピーコード化

【技術分野】

【0001】

本開示は、ビデオコード化に関し、より詳細には、ビデオコード化のためのエントロピー-コード化に関する。

【背景技術】

【0002】

デジタルビデオ機能は、デジタルテレビジョン、デジタルダイレクトブロードキャストシステム、ワイヤレスブロードキャストシステム、携帯情報端末(PDA)、ラップトップ又はデスクトップコンピュータ、デジタルカメラ、デジタル記録装置、デジタルメディアプレーヤ、ビデオゲーム機器、ビデオゲームコンソール、セルラー電話又は衛星無線電話、ビデオ遠隔会議機器などを含む、広範囲にわたる機器に組み込まれ得る。デジタルビデオ機器は、MPEG-2、MPEG-4、ITU-T H.263、ITU-T H.264/MPEG-4、Part 10、アドバンストビデオコード化(AVC:Advanced Video Coding)、又は新生の高効率ビデオコード化(HEVC)規格によって定義された規格、及びそのような規格の拡張に記載されているビデオ圧縮技法など、ビデオ圧縮技法を実施する。

【0003】

ビデオ圧縮技法では、ビデオシーケンスに固有の冗長性を低減又は除去するために空間的予測及び/又は時間的予測を実行する。ブロックベースのビデオコード化の場合、ビデオフレーム又はスライスがビデオブロック又はコード化単位(CU)に区分され得る。CUの予測ビデオデータを求めるために、CUをさらに1つ以上の予測単位(PU)に区分してもよい。ビデオ圧縮技法では、CUを、コード化すべきビデオブロックと予測ビデオデータとの差を表す残差ビデオブロックデータの1つ以上の変換単位(TU)に区分してもよい。2次元離散コサイン変換(DCT)などの線形変換をTUに施して残差ビデオブロックデータを画素領域から周波数領域に変換して、さらなる圧縮を実現してもよい。

【0004】

変換に続いて、TU内の変換係数を量子化によりさらに圧縮してもよい。量子化に続いて、エントロピーコード化ユニットが、ジグザグスキヤン又はTUのサイズに関する別のスキヤン順序を適用してTU内の係数の2次元アレイをスキヤンし、エントロピーコード化することができるシリアル化されたベクトルを生成してもよい。エントロピーコード化ユニットは次いで、係数のシリアル化ベクトルをエントロピーコード化してもよい。例えば、エントロピーコード化ユニットは、コンテキスト適応型可変長コード化(CAVLC)、コンテキスト適応型2値算術コード化(CABAC)、又は別のエントロピーコード化技法を実行し得る。コンテキスト適応エントロピーコード化の場合、エントロピーコード化ユニットは、TUのサイズに関するコンテキストモデルに従ってTU内の各係数のコンテキストを選択することができる。次いで、選択されたコンテキストに基づいて各係数をエントロピーコード化してもよい。

【発明の概要】

【0005】

概して、本開示では、それぞれ異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用して残差ビデオデータのブロックに関連する変換係数のエントロピーコード化及び復号を実行するための技法について説明する。例えば、ジョイントコンテキストモデルは、 32×32 の第1のサイズを有する変換単位と 16×16 の第2のサイズを有する変換単位との間で共有されてもよい。場合によっては、2つよりも多くのサイズの変換単位が同じジョイントコンテキストモデルを共有してもよい。一例として、ジョイントコンテキストモデルは、変換単位の有意性マップのジョイントコンテキストモデルであってもよい。他の例では、ジョイントコンテキストモデルを他のコード化情報又はシンタックス要素に関連付けてもよい。

【0006】

新生の高効率ビデオコード化(HEVC)規格では、コード化単位(CU)は、変換される残差ビデオデータを含む1つ以上の変換単位(TU)を含んでもよい。ビデオコード化効率を向上させるために、例えば、 32×32 から 128×128 までの追加の変換単位サイズが提案されているが、この場合、追加の変換単位サイズの各々にコンテキストモデルを維持するためにメモリ及び計算要件も増大する。それぞれ異なるサイズの変換単位同士の間で共有されるジョイントコンテキストモデルを使用してエントロピーコード化を実行すると、ビデオ符号化装置及びビデオ復号装置上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオ符号化装置及び復号装置上にコンテキストモデルを維持することについての計算コストを削減することができる。

【0007】

幾つかの例では、これらの技法は、2次元変換を実行するときに変換単位サイズを大きくする中間バッファリング要件を低減させることもできる。その場合、この技法は、2次元変換の各方向を適用して保持される係数ブロックを生成した後、第1のサイズの変換単位に含まれる変換係数のうちのより周波数の高い変換係数のサブセットのゼロ化、即ち、値をゼロに設定することを含む。この例では、エントロピーコード化のためのジョイントコンテキストモデルは、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する変換単位と本来第2のサイズを有する変換単位との間で共有されてもよい。場合によっては、保持される係数ブロックは、第2のサイズと等しいサイズを有してもよい。他の例では、エントロピーコード化のためのジョイントコンテキストモデルは、第1のサイズを有する変換単位と第2のサイズを有する変換単位との間で共有されてもよい。

【0008】

一例において、本開示では、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持することを備える、ビデオデータを復号する方法について説明する。この方法は、ジョイントコンテキストモデルに従って、保持される係数ブロックを含む第1のサイズと第2のサイズのうちの一方

を有する変換単位に関連する係数のコンテキストを選択することと、選択されたコンテキストに基づくコード化プロセスに従って変換単位に関連する係数をエントロピーコード化することとも含む。

【0009】

別の例において、本開示では、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを記憶するメモリを備えるビデオコード化装置について説明する。ビデオコード化装置は、ジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って、保持される係数ブロックを含む第1のサイズと第2のサイズのうちの一方を有する変換単位に関連する係数のコンテキストを選択し、選択されたコンテキストに基づくコード化プロセスに従って変換単位に関連する係数をエントロピーコード化するように構成されたプロセッサをさらに含む。

【0010】

さらなる一例において、本開示では、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持するための手段を備えるビデオコード化装置について説明する。ビデオコード化装置は、ジョイントコンテキストモデルに従って、保持される係数ブロックを含む第1のサイズと第2のサイズのうちの一方を有する変換単位に関連する係数のコンテキストを選択するための手段と、選択されたコンテキストに基づくコード化プロセスに従って変換単位に関連する係数をエントロピーコード化するための手段も備える。

【0011】

別の例において、本開示では、実行されたときに、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルをプロセッサに維持させる、ビデオデータをコード化するための命令を備えるコンピュータ可読媒体について説明する。この命令はまた、プロセッサに、ジョイントコンテキストモデルに従って、保持される係数ブロックを含む第1のサイズと第2のサイズのうちの一方を有する変換単位に関連する係数のコンテキストを選択させ、選択されたコンテキストに基づくコード化プロセスに従って変換単位に関連する係数をエントロピーコード化させる。

【0012】

別の例において、本開示では、第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持することを備える、ビデオデータをコード化する方法について説明する。この方法は、ジョイントコンテキストモデルに従って第1のサイズと第2のサイズのうちの一方を有する変換単位に関連する係数のコンテキストを選択することと、選択されたコンテキストに基づくコード化プロセスに従って変換単位の係数をエントロピーコード化することとも含む。

【0013】

さらなる例において、本開示では、第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを記憶するメモリと、ジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って第1のサイズと第2のサイズのうちの一方を有する変換単位に関連する係数のコンテキストを選択し、選択されたコンテキストに基づくコード化プロセスに従って変換単位の係数をエントロピーコード化するように構成されたプロセッサとを備えるビデオコード化装置について説明する。

【0014】

さらなる例において、本開示では、第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持するための手段を備えるビデオコード化装置について説明する。このビデオコード

ド化装置は、ジョイントコンテキストモデルに従って第1のサイズと第2のサイズのうちの一方を有する変換単位に関する係数のコンテキストを選択するための手段と、選択されたコンテキストに基づくコード化プロセスに従って変換単位の係数をエントロピーコード化するための手段とも含む。

【0015】

別の例において、本開示では、実行されたときに、プロセッサに、第1のサイズを有する変換単位と第1のサイズとは異なる第2のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルをプロセッサに維持させ、ジョイントコンテキストモデルに従って第1のサイズと第2のサイズのうちの一方を有する変換単位に関する係数のコンテキストを選択させ、選択されたコンテキストに基づくコード化プロセスに従って変換単位の係数をエントロピーコード化させる、ビデオデータをコード化するための命令を備えるコンピュータ可読媒体について説明する。

【図面の簡単な説明】

【0016】

【図1】それぞれ異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用してビデオ係数のエントロピーコード化を実行するための技法を利用する例示的なビデオ符号化及び復号システムを示すブロック図。

【図2】ジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化するための技法を実施できる例示的なビデオエンコーダを示すブロック図。

【図3A】第1のサイズを有する変換単位から得た第2のサイズを有する保持される係数ブロックの方形領域を示す概念図。

【図3B】第1のサイズを有する変換単位から得た第2のサイズを有する保持される係数ブロックの矩形領域を示す概念図。

【図4】ジョイントコンテキストモデルを使用してビデオ係数をエントロピー復号するための技法を実施することのできる例示的なビデオデコーダを示すブロック図。

【図5】ジョイントコンテキストモデルに従ってビデオ係数のコンテキストを選択するように構成された例示的なエントロピー符号化ユニットを示すブロック図。

【図6】ジョイントコンテキストモデルに従ってビデオ係数のコンテキストを選択するように構成された例示的なエントロピー復号ユニットを示すブロック図。

【図7】第1のサイズを有する第1の変換単位と第2のサイズを有する第2の変換単位との間で共有されるジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化しつつエントロピー復号する例示的な動作を示すフローチャート。

【図8】第2のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する第1の変換単位と第2のサイズを有する第2の変換単位との間で共有されるジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化しつつエントロピー復号する例示的な動作を示すフローチャート。

【発明を実施するための形態】

【0017】

概して、本開示では、それぞれ異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用して残差ビデオデータのブロックに関連する変換係数のエントロピー符号化及び復号を実行するための技法について説明する。例えば、ジョイントコンテキストモデルは、 32×32 の第1のサイズを有する変換単位と 16×16 の第2のサイズを有する変換単位との間で共有されてもよい。新生の高効率ビデオコード化(HEVC)規格では、コード化単位(CU)は、残差ビデオデータを含む1つ以上の変換単位(TU)を含んでもよい。変換の前に、残差ビデオデータは残差画素値を空間領域に含む。変換後、残差ビデオデータは残差変換係数を変換領域に含む。ビデオコード化効率を向上させるために、例えば、 32×32 から 128×128 までの追加の変換単位サイズが提案されているが、その結果、追加の変換単位サイズの各々にコンテキストモデルを維持するためにメモリ及び計算要件も増大する。それぞれ異なるサイズの変換単位同士の間で共有されるジョイントコンテキストモデルを使用してエントロピーコード化を実行

すると、ビデオ符号化装置及びビデオ復号装置上にコンテキスト及び確率を記憶するに必要なメモリの量を低減させ、ビデオ符号化装置及び復号装置上にコンテキストモデルを維持することについての計算コストを削減することができる。

【0018】

幾つかの例では、これらの技法は、2次元変換を実行するときに変換単位サイズを大きくする中間バッファリング要件を低減させることもできる。この技法は、2次元変換の各方向を適用して保持される係数ブロックを生成した後、第1のサイズの変換単位に含まれる変換係数のうちのより周波数の高い変換係数のサブセットのゼロ化、即ち値をゼロに設定することを含む。その場合、ビデオ符号化装置及び復号装置は、2次元変換の各方向、即ち行方向及び列方向を適用する合間にバッファリングする係数の数を減らすことができる。第1のサイズを有する変換単位においてより周波数の高い係数がゼロ化されるとき、保持される係数ブロックに含まれる係数は、本来第2のサイズを有する変換単位に含まれる係数と同様の確率統計を有する。この場合、エントロピーコード化のためのジョイントコンテキストモデルは、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する変換単位と本来第2のサイズを有する変換単位との間で共有されてもよい。場合によっては、保持される係数ブロックは、第2のサイズと等しいサイズを有してもよい。他の場合には、保持される係数ブロックは、第1のサイズとも第2のサイズとも異なる第3のサイズと等しいサイズを有してもよい。

【0019】

他の例では、第1のサイズを有する第1の変換単位に含まれる係数は、第1の変換単位内により周波数の高い係数をゼロ化しない場合でも第2のサイズの第2の変換単位に含まれる係数と同様の確率統計を有してもよい。このことは、より周波数の高い係数が、エントロピーコード化のための近接する係数の確率統計に対する影響が無視できるのほど少ない残差ビデオデータを表し得るために可能になる。この場合、エントロピーコード化のためのジョイントコンテキストモデルは、第1のサイズを有する変換単位と第2のサイズを有する変換単位との間で共有されてもよい。

【0020】

図1は、それぞれ異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用してビデオ係数のエントロピーコード化を実行するための技法を利用することのできる例示的なビデオ符号化及び復号システムを示すブロック図である。図1に示すように、システム10は、符号化されたビデオを記憶し、及び/又は通信チャネル16を介して符号化ビデオを宛先機器14へ送信することができる発信源機器12を含む。発信源機器12及び宛先機器14は、必ずしも互いにリアルタイムにアクティブに通信するとは限らない。場合によっては、発信源機器12は、宛先機器14が必要に応じてディスクアクセスを介してアクセスすることができる記憶媒体に符号化されたビデオデータを記憶してもよく、又は宛先機器14が必要に応じてストリーミングを介してアクセスすることができるファイルサーバに符号化されたビデオデータを記憶してもよい。発信源機器12及び宛先機器14は、広範囲の機器のいずれかを備えてもよい。場合によっては、発信源機器12及び宛先機器14は、通信チャネル16を介してビデオ情報を通信することができるワイヤレス通信機器を備え得、その場合、通信チャネル16はワイヤレスである。

【0021】

但し、ジョイントコンテキストモデルを使用したビデオ係数のエントロピーコード化に関する本開示の技法は、必ずしもワイヤレスアプリケーション又はワイヤレス設定に限定されるとは限らない。例えば、これらの技法は、オーバージェアテレビジョン放送、ケーブルテレビジョン送信、衛星テレビジョン送信、インターネットビデオ送信、記憶媒体上に符号化される符号化デジタルビデオ、又は他のシナリオに適用し得る。従って、通信チャネル16は、符号化ビデオデータの送信に好適なワイヤレス又はワイヤード媒体の任意の組合せを備え得、機器12、14は、携帯電話、スマートフォン、デジタルメディアプレーヤ、セットトップボックス、テレビジョン、表示器、デスクトップコンピュータ、ポ

ータブルコンピュータ、タブレットコンピュータ、ゲームコンソール、ポータブルゲーム機器などの様々なワイヤード又はワイヤレス媒体機器のいずれかを備え得る。

【0022】

図1の例では、発信源機器12は、ビデオ発信源18と、ビデオエンコーダ20と、変調器／復調器（モデム）22と、送信機24とを含む。宛先機器14は、受信機26と、モデム28と、ビデオデコーダ30と、表示装置32とを含む。他の例では、発信源機器及び宛先機器は他の構成要素又は構成を含み得る。例えば、発信源機器12は、外部カメラ、ビデオストレージアーカイブ、コンピュータグラフィックスソースなどの外部ビデオ発信源18からビデオデータを受信し得る。同様に、宛先機器14は、一体型表示装置を含むのではなく、外部表示装置とインターフェースし得る。

【0023】

図1の図示のシステム10は一例にすぎない。他の例では、任意のデジタルビデオ符号化装置及び／又は復号装置が、ジョイントコンテキストモデルを使用してビデオ係数のエントロピー~~コード~~化のための開示された技法を実行してもよい。本技法はまた、一般に「コーデック」と呼ばれるビデオエンコーダ／デコーダによって実行され得る。さらに、本開示の技法はまた、ビデオプリプロセッサによって実行され得る。発信源機器12及び宛先機器14は、発信源機器12が宛先機器14に送信するための~~コード~~化されたビデオデータを発生するような、コード化装置の例にすぎない。幾つかの例では、機器12、14の各々がビデオ符号化構成要素及び復号構成要素を含むように、機器12、14は、実質的に対称的に動作し得る。従って、システム10は、例えば、ビデオストリーミング、ビデオ再生、ビデオブロードキャスト又はビデオ電話通信のためのビデオ機器12とビデオ機器14との間の一方向又は双方向のビデオ送信をサポートすることができる。

【0024】

発信源機器12のビデオ発信源18は、ビデオカメラなどの撮像装置、以前に撮影されたビデオを含んでいるビデオアーカイブ、及び／又はビデオコンテンツプロバイダからのビデオフィードを含み得る。さらなる代替として、ビデオ発信源18は、ソースビデオとしてのコンピュータグラフィックスベースのデータ、又はライブビデオとアーカイブビデオとコンピュータ生成ビデオとの組合せを生成し得る。場合によっては、ビデオ発信源18がビデオカメラである場合、発信源機器12及び宛先機器14は、いわゆるカメラ付き携帯電話又はビデオ電話を形成することができる。但し、上述のように、本開示で説明する技法は、一般にビデオ~~コード~~化に適用可能であり、ワイヤレス及び／又はワイヤードアプリケーションに適用可能であり得る。各場合に、撮影されたビデオ、以前に撮影されたビデオ又はコンピュータ生成ビデオは、ビデオエンコーダ20によって符号化され得る。次いで、符号化ビデオ情報は、通信規格に従ってモデム22によって変調され、送信機24を介して宛先機器14に送信され得る。モデム22は、信号変調のために設計された様々なミキサ、フィルタ、増幅器又は他の構成要素を含んでもよい。送信機24は、増幅器、フィルタ、及び1つ以上のアンテナを含む、データを送信するために設計された回路を含んでもよい。

【0025】

本開示によれば、発信源機器12のビデオエンコーダ20は、ジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化するための技法を適用するように構成され得る。ビデオフレームの~~コード~~化単位（CU）は、残差ビデオデータを含む1つ以上の変換単位（TU）を含んでもよい。変換の前に、残差ビデオデータは残差画素値を空間領域に含む。変換後、残差ビデオデータは残差変換係数を変換領域に含む。ビデオエンコーダ20は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って1つの変換単位に関連する係数のコンテキストを選択してもよい。次いで、ビデオエンコーダ20は、選択されたコンテキストに基づいて係数をエントロピー符号化してもよい。

【0026】

一例として、ジョイントコンテキストモデルは、 32×32 の第1のサイズを有する変

換単位と 16×16 の第 2 のサイズを有する変換単位との間で共有されてもよい。他の例では、2つよりも多くのサイズの変換単位が同じジョイントコンテキストモデルを共有してもよい。さらに、2つ又はそれ以上の変換単位が TU 用の幾つかのコンテキストモデル又は全てのコンテキストモデルを共有してもよい。ある場合には、ジョイントコンテキストモデルは、TU の有意性マップのジョイントコンテキストモデルであってもよい。他の場合には、ジョイントコンテキストモデルを他の コード化 情報又はシンタックス要素に関連付けてもよい。従って、これらの技法は、ビデオエンコーダ 20 上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオエンコーダ 20 上にコンテキストモデルを維持することについての計算コストを削減することができる。

【0027】

一例では、ビデオエンコーダ 20 は、2 次元変換の各方向を適用して保持される係数ブロックを生成した後、第 1 のサイズの変換単位に含まれる変換係数のうちのより周波数の高い変換係数のサブセットをゼロ化し、即ち値をゼロに設定することができる。この場合、各技法は、2 次元変換の各方向、即ち行方向及び列方向を適用する合間にバッファリングすべき係数の数を減らすことができる。変換単位においてより周波数の高い係数がゼロ化されるとき、保持される係数ブロックに含まれる係数は、本来第 2 のサイズを有する変換単位に含まれる係数と同様の確率統計を有する。この例では、ビデオエンコーダ 20 は、保持される係数ブロックを生成するためにゼロ化される係数を含む第 1 のサイズを有する変換単位と第 2 のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って、保持される係数ブロックを含む第 1 のサイズと第 2 のサイズのうちの一方の変換単位の係数のコンテキストを選択してもよい。場合によっては、保持される係数ブロックは、第 2 のサイズと等しいサイズを有してもよい。他の場合には、保持される係数ブロックは、第 1 のサイズとも第 2 のサイズとも異なる第 3 のサイズと等しいサイズを有してもよい。

【0028】

別の例では、第 1 のサイズを有する第 1 の変換単位に含まれる係数は、第 1 の変換単位内により周波数の高い係数をゼロ化しない場合でも第 2 のサイズを有する第 2 の変換単位に含まれる係数と同様の確率統計を有してもよい。このことは、高周波数係数が、エントロピー コード化 のための近接する係数の確率統計に対する影響が無視できるほど少ない残差ビデオデータを表し得るために可能になる。この例では、ビデオエンコーダ 20 は、第 1 のサイズを有する変換単位と第 2 のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って、第 1 のサイズと第 2 のサイズのうちの一方の変換単位の係数のコンテキストを選択してもよい。場合によっては、第 1 のサイズ及び第 2 のサイズの変換単位内の高周波数係数のみがジョイントコンテキストモデルを共有してもよい。第 1 のサイズの変換内の低周波数係数、例えば DC 構成要素及び近接する係数は、第 2 のサイズの変換内の低周波数係数とは異なるコンテキストモデルを使用してもよい。

【0029】

宛先機器 14 の受信機 26 はチャネル 16 を介して情報を受信し、モデム 28 は情報を復調する。チャネル 16 を介して伝達される情報は、コード化 単位 (CU)、予測単位 (PU)、変換単位 (TU)、又は コード化 されたビデオの他の単位、例えば、ビデオストリーム、ビデオフレーム、及びビデオシーケンス又はピクチャグループ (GOP) の特性及び / 又は処理を記述するシンタックス要素を含む、ビデオデコーダ 30 によっても使用される、ビデオエンコーダ 20 によって定義されるシンタックス情報を含み得る。表示装置 32 は、復号されたビデオデータをユーザに対して表示し、陰極線管 (CRT)、液晶表示器 (LCD)、プラズマ表示器、有機発光ダイオード (OLED) 表示器、又は別のタイプの表示装置など、様々な表示装置のいずれかを備え得る。

【0030】

本開示によれば、宛先機器 14 のビデオデコーダ 30 は、ジョイントコンテキストモデルを使用してビデオ係数をエントロピー復号するための技法を適用するように構成され得

る。復号すべきビデオフレームの C U は、変換前及び変換後の残差ビデオデータを含む 1 つ以上の T U を含んでもよい。ビデオデコーダ 3 0 は、それぞれ異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って 1 つの変換単位に関連する係数のコンテキストを選択してもよい。次いで、ビデオデコーダ 3 0 は、選択されたコンテキストに基づいて係数をエントロピー復号してもよい。

【 0 0 3 1 】

一例として、ジョイントコンテキストモデルは、 32×32 の第 1 のサイズを有する変換単位と 16×16 の第 2 のサイズを有する変換単位との間で共有されてもよい。上述のように、他の例では、2つよりも多くのサイズの変換単位が同じジョイントコンテキストモデルを共有してもよい。さらに、2つ又はそれ以上の変換単位が T U 用の幾つかのコンテキストモデル又は全てのコンテキストモデルを共有してもよい。ある場合には、ジョイントコンテキストモデルは、T U の有意性マップのジョイントコンテキストモデルであってもよい。他の場合には、ジョイントコンテキストモデルを他のコード化情報又はシンタックス要素に関連付けてもよい。従って、これらの技法は、ビデオデコーダ 3 0 上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオデコーダ 3 0 上にコンテキストモデルを維持することについての計算コストを削減することができる。

【 0 0 3 2 】

一例では、ビデオデコーダ 3 0 は、保持される係数ブロックに関連付けられた符号化された係数、及び第 1 のサイズを有する変換単位におけるゼロ化された係数を表すビットストリームを受信してもよい。変換単位においてより周波数の高い係数がゼロ化されるとき、保持される係数ブロックに含まれる係数は、本来第 2 のサイズを有する変換単位に含まれる係数と同様の確率統計を有する。この例では、ビデオデコーダ 3 0 は、保持される係数ブロックを生成するためにゼロ化された係数を含む第 1 のサイズを有する変換単位と第 2 のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って、保持される係数ブロックを含む第 1 のサイズと第 2 のサイズのうちの一方の変換単位に関連する符号化された係数のコンテキストを選択してもよい。場合によっては、保持される係数ブロックは、第 2 のサイズと等しいサイズを有してもよい。他の場合には、保持される係数ブロックは、第 1 のサイズとも第 2 のサイズとも異なる第 3 のサイズと等しいサイズを有してもよい。

【 0 0 3 3 】

別の例では、ビデオデコーダ 3 0 は、第 1 のサイズと第 2 のサイズのうちの一方を有する変換単位に関連付けられた符号化された係数を表すビットストリームを受信してもよい。第 1 のサイズを有する第 1 の変換単位に含まれる係数は、第 1 の変換単位内の高周波数係数をゼロ化しない場合でも第 2 のサイズの変換単位に含まれる係数と同様の確率統計を有してもよい。この例では、ビデオデコーダ 3 0 は、第 1 のサイズを有する変換単位と第 2 のサイズを有する変換単位とによって共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って第 1 のサイズと第 2 のサイズのうちの一方の変換単位に関連する符号化された係数のコンテキストを選択してもよい。場合によっては、第 1 のサイズ及び第 2 のサイズの変換単位内の高周波数係数のみがジョイントコンテキストモデルを共有してもよい。第 1 のサイズの変換内の低周波数係数、例えば D C 構成要素及び近接する係数は、第 2 のサイズの変換内の低周波数係数とは異なるコンテキストモデルを使用してもよい。

【 0 0 3 4 】

図 1 の例では、通信チャネル 1 6 は、無線周波数 (R F) スペクトル又は 1 つ以上の物理的伝送線路など、ワイヤレス又はワイヤードの任意の通信媒体、若しくはワイヤレス及びワイヤードの媒体の任意の組合せを備えることができる。通信チャネル 1 6 は、ローカルエリアネットワーク、ワイドエリアネットワーク、又はインターネットなどのグローバルネットワークなど、パケットベースのネットワークの一部を形成することができる。通信チャネル 1 6 は、一般に、ワイヤード又はワイヤレス媒体の任意の好適な組合せを含む

、ビデオデータを発信源機器12から宛先機器14に送信するのに好適な任意の通信媒体、又は様々な通信媒体の集合体を表す。通信チャネル16は、発信源機器12から宛先機器14への通信を可能にするのに有用なルータ、スイッチ、基地局、又は任意の他の機器を含んでもよい。上述のように、場合によっては、発信源機器12と宛先機器14は、通信チャネル16を介してリアルタイムにアクティブに通信しなくてもよい。その代わりに、例えば、発信源機器12は、宛先機器14が必要に応じてディスクアクセスを介してアクセスすることができる記憶媒体に符号化されたビデオデータを記憶してもよく、あるいは宛先機器14が必要に応じてストリーミングを介してアクセスすることができるファイルサーバに符号化されたビデオデータを記憶してもよい。

【0035】

ビデオエンコーダ20及びビデオデコーダ30は、新生の高効率ビデオコード化(HEVC)規格又は代替的にMPEG-4、Part 10、アドバンストビデオコード化(AVC)とも呼ばれるITU-T H.264規格など、ビデオ圧縮規格に従って動作し得る。但し、本開示の技法は、いかなる特定のコード化規格にも限定されない。他の例にはMPEG-2及びITU-T H.263がある。図1には示されていないが、幾つかの態様では、ビデオエンコーダ20及びビデオデコーダ30は、それぞれオーディオエンコーダ及びデコーダと統合され得、適切なMUX-DEMUXユニット、又は他のハードウェア及びソフトウェアを含んで、共通のデータストリーム又は別個のデータストリーム中のオーディオとビデオの両方の符号化に対処し得る。適用可能な場合、MUX-DEMUXユニットはITU H.223マルチプレクサプロトコル、又はユーザデータグラムプロトコル(UDP)などの他のプロトコルに準拠することができる。

【0036】

HEVCの規格化の取り組みは、HEVCテストモデル(HM)と呼ばれるビデオコード化装置のモデルに基づく。HMは、例えば、ITU-T H.264/AVCに従う既存の機器に対してビデオコード化装置の幾つかの追加の能力を仮定する。HMは、ビデオデータのブロックをコード化単位(CU)と称する。ビットストリーム内のシンタックスデータが、画素の数に関して最大のコード化単位である最大コード化単位(LCU)を定義し得る。概して、CUは、CUがサイズの差異を有さないことを除いて、H.264規格のマクロブロックと同様の目的を有する。従って、CUは、サブCUに分割され得る。概して、本開示におけるCUへの言及は、ピクチャの最大コード化単位又はLCUのサブCUを指すことがある。LCUはサブCUに分割され得、各サブCUは、さらに、サブCUに分割され得る。ビットストリームのシンタックスデータは、CU深さと呼ばれる、LCUが分割され得る最大回数を定義し得る。従って、ビットストリームは最小コード化単位(SCU)をも定義し得る。

【0037】

それ以上分割されないCU(即ち、LCUのリーフノード)は、1つ以上の予測単位(PU)を含み得る。概して、PUは、対応するCUの全部又は一部分を表し、そのPUの参照サンプルを取り出すためのデータを含む。例えば、PUがイントラモード符号化されるとき、PUは、そのPUのためのイントラ予測モードを記述するデータを含み得る。別の例として、PUがインターモード符号化されるとき、PUは、PUの動きベクトルを定義するデータを含み得る。動きベクトルを定義するデータは、例えば、動きベクトルの水平成分、動きベクトルの垂直成分、動きベクトルの解像度(例えば、1/2画素精度、1/4画素精度又は1/8画素精度)、動きベクトルが指し示す参照フレーム、及び/又は動きベクトルの参照フレームリスト(例えば、リスト0又はリスト1)を記述し得る。(1つ以上の)PUを定義するCUについてのデータは、例えば、1つ以上のPUへのCUの区分についても記述し得る。区分モードは、CUが、スキップモード符号化又はダイレクトモード符号化されるか、イントラ予測モード符号化されるか、あるいはインター予測モード符号化されるかによって異なり得る。

【0038】

1つ以上のPUを有するCUは、1つ以上の変換単位(TU)をも含み得る。PUを使

用した予測に続いて、ビデオエンコーダは、PUに対応するCUの部分についての残差値を計算し得る。TUに含まれる残差値は、エントロピーコード化のためのシリアル化変換係数を生成するために、変換係数に変換され、次いで量子化され、スキャンされ得る画素差分値に対応する。TUは、必ずしもPUのサイズに制限されるとは限らない。従って、各TUは、同じCUについての対応するPUよりも大きくても小さくてもよい。幾つかの例では、TUの最大サイズは、対応するCUのサイズであり得る。本開示は、CU、PU、又はTUのいずれかを指すために「ビデオブロック」という用語を使用する。

【0039】

ビデオエンコーダ20及びビデオデコーダ30はそれぞれ、1つ以上のマイクロプロセッサ、デジタル信号プロセッサ(DSP)、特定用途向け集積回路(ASIC)、フィールドプログラマブルゲートアレイ(FPGA)、ディスクリート論理、ソフトウェア、ハードウェア、ファームウェアなど、様々な好適なエンコーダ回路のいずれか、又はそれらの任意の組合せとして実施され得る。ビデオエンコーダ20及びビデオデコーダ30の各々は1つ以上のエンコーダ又はデコーダ中に含まれ得、そのいずれも複合エンコーダ/デコーダ(コーデック)の一部としてそれぞれのカメラ、コンピュータ、モバイル機器、加入者機器、ブロードキャスト機器、セットトップボックス、サーバなどに統合され得る。

【0040】

ビデオシーケンス又はピクチャのグループ(GOP)は、一般に一連のビデオフレームを含む。GOPは、GOP中に含まれる幾つかのフレームを記述するシンタックスデータを、GOPのヘッダ、GOPの1つ以上のフレームのヘッダ、又は他の場所中に含み得る。各フレームは、それぞれのフレームについての符号化モードを記述するフレームシンタックスデータを含み得る。ビデオエンコーダ20は、一般に、ビデオデータを符号化するために、個々のビデオフレーム内のビデオブロックに対して動作する。ビデオブロックは、CU又はCUのPUに対応し得る。ビデオブロックは、サイズを固定することも変更することもでき、指定のコード化規格に応じてサイズが異なることがある。各ビデオフレームは複数のスライスを含み得る。各スライスは、1つ以上のPUを含み得る、複数のCUを含み得る。

【0041】

一例として、HEVCテストモデル(HM)は、様々なCUサイズでの予測をサポートする。LCUのサイズは、シンタックス情報によって定義され得る。特定のCUのサイズが $2N \times 2N$ であると仮定すると、HMは、 $2N \times 2N$ 又は $N \times N$ のサイズでのイントラ予測をサポートし、 $2N \times 2N$ 、 $2N \times N$ 、 $N \times 2N$ 、又は $N \times N$ の対称サイズでのインター予測をサポートする。HMはまた、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、及び $nR \times 2N$ のインター予測のための非対称分割をサポートする。非対称分割では、CUの一方は分割されないが、他の方向が25%と75%とに分割される。25%の分割に対応するCUの部分は、「n」とその後ろに付く「Up」、「Down」、「Left」、又は「Right」という表示によって示される。従って、例えば「 $2N \times nU$ 」は、上部の $2N \times 0.5N$ PUと下部の $2N \times 1.5N$ PUとで水平方向に分割される $2N \times 2N$ CUを指す。

【0042】

本開示では、「 $N \times (x)N$ 」と「 $N \times (by)N$ 」とは、垂直寸法及び水平寸法に関するビデオブロック(例えば、CU、PU、又はTU)の画素寸法、例えば、 $16 \times (x)16$ 画素又は $16 \times (by)16$ 画素を指すために互換的に使用され得る。一般に、 16×16 ブロックは、垂直方向に16画素を有し($y = 16$)、水平方向に16画素を有する($x = 16$)。同様に、 $N \times N$ ブロックは、一般に、垂直方向にN画素を有し、水平方向にN画素を有し、Nは、非負整数値を表す。ブロック中の画素は行と列に構成され得る。さらに、ブロックは、必ずしも、水平方向に垂直方向と同じ数の画素を有する必要はない。例えば、ブロックは、 $N \times M$ 画素を有する矩形領域を備え得、Mは必ずしもNに等しいとは限らない。

【0043】

イントラ予測コード化又はインター予測コード化を行ってCUのためのPUを生成した後、ビデオエンコーダ20は、残差データを計算して、CUのための1つ以上のTUを生成し得る。残差データは、非符号化ピクチャの画素とCUのPUの予測値との間の画素差分に対応し得る。ビデオエンコーダ20は、CUについての残差データを含む1つ以上のTUを形成し得る。ビデオエンコーダ20は、次いで、TUを変換し得る。離散コサイン変換(DCT)、整数変換、ウェーブレット変換、又は概念的に同様の変換などの変換が施される前に、CUの各TUは、画素領域に画素差分値を備える残差ビデオデータを備えてよい。変換が施された後、各TUは、残差ビデオデータを表す変換係数を周波数領域に備えてよい。

【0044】

変換係数を生成するための変換の後、変換係数の量子化が実行され得る。量子化は、概して、係数を表すために使用されるデータ量をできるだけ低減するために変換係数を量子化するプロセスを指す。量子化プロセスは、係数の一部又は全部に関連するビット深度を低減させることができる。例えば、量子化中にnビット値がmビット値に切り捨てられ得、nはmよりも大きい。

【0045】

ビデオエンコーダ20は、ジグザグスキャン、水平スキャン、垂直スキャン、又はTUのサイズに関連する別のスキャン順序を適用して量子化変換係数をスキャンし、エントロピー符号化することのできるシリアル化されたベクトルを生成することができる。幾つかの例では、ビデオエンコーダ20は、予め定義されたスキャン順序を利用して量子化変換係数をスキャンし得る。他の例では、ビデオエンコーダ20は、適応型スキャンを実行し得る。ビデオエンコーダ20は、量子化変換係数をスキャンして1次元ベクトルを形成した後、例えば、コンテキスト適応型可変長コード化(CAVLC)、コンテキスト適応型バイナリ算術コード化(CABAC)、又は別のエントロピー符号化方法に従って1次元ベクトルをエントロピー符号化し得る。

【0046】

ビデオエンコーダ20は、コンテキスト適応型エントロピーコード化を実行するためには、例えば、近接する係数の値が非ゼロであるかどうかに関係し得るコンテキストモデルに従って各係数にコンテキストを割り当ててもよい。ビデオエンコーダ20は次いで、コンテキストモデル内の割り当てられたコンテキストに関連する係数用のコード化プロセスを決定する。従来、ビデオエンコーダ20は、実装されるビデオ圧縮規格によって支持されるTUの様々なサイズの各々に別個のコンテキストモデルを維持しなければならない。ビデオコード化効率を向上させるために、例えば、 32×32 から 128×128 までの追加の変換単位サイズが提案されているが、追加のTUサイズによって、追加の変換単位サイズの各々にコンテキストモデルを維持するためにメモリ及び計算要件も増大する。

【0047】

本開示の各技法によれば、ビデオエンコーダ20は、コンテキスト適応型エントロピーコード化を実行するために、様々なサイズのTU同士の間で共有されるジョイントコンテキストモデルに従って各係数のコンテキストを選択することができる。より具体的には、ビデオエンコーダ20は、TUの事前に符号化された近接する係数の値に基づいてTUの所与の係数にジョイントコンテキストモデル内のコンテキストを割り当てることができる。割り当てられるコンテキストは、TUによって共有されるジョイントコンテキストモデルによって定義される基準に基づいて選択される。ビデオエンコーダ20は、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する係数用のコード化プロセスを決定してもよい。次いで、ビデオエンコーダ20は、求められた確率推定値に基づいて係数をエントロピー符号化する。例えば、CABACの場合、ビデオエンコーダ20は、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する係数の値(例えば、0又は1)の確率推定値を求めてよい。ビデオエンコーダ20は次いで、係数の実際のコード化された値に基づいてジョイントコンテキストモデル内の割り当てられたコンテキストに関連する確率推定値を更新する。

【0048】

一例として、ビデオエンコーダ20は、本来第1のサイズを有する第1の変換単位内の保持される係数ブロック又は本来第2のサイズを有する第2の変換単位のいずれかに関連する係数に同じジョイントコンテキストモデルを使用してコンテキストを選択してもよい。さらに、ビデオエンコーダ20は、本来第1のサイズを有する第1の変換単位内の保持される係数ブロック又は本来第2のサイズを有する第2の変換単位のいずれかの係数の実際のコード化された値に基づいてジョイントコンテキストモデル内の選択されたコンテキストに関連する確率推定値を更新してもよい。別の例として、ビデオエンコーダ20は、第1のサイズを有する第1の変換単位又は第2のサイズを有する第2の変換単位のいずれかに関連する係数に同じジョイントコンテキストモデルを使用してコンテキストを選択してもよい。この場合、ビデオエンコーダ20は次いで、第1のサイズの第1の変換単位又は第2のサイズの第2の変換単位のいずれかの係数の実際のコード化された値に基づいてジョイントコンテキストモデル内の選択されたコンテキストに関連する確率推定値を更新してもよい。

【0049】

いずれの場合も、2つ又はそれ以上のサイズの変換単位間でジョイントコンテキストモデルを共有すると、ビデオエンコーダ20上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させることができる。さらに、ジョイントコンテキストモデルを共有すると、ビデオスライスの開始時に全てのコンテキストモデルをリセットすることを含む、ビデオエンコーダ20上にコンテキストモデルを維持することについての計算コストを削減することもできる。

【0050】

ビデオエンコーダ20は、予測情報を示すシンタックス要素をエントロピー符号化してもよい。例えば、ビデオエンコーダ20は、CU、PU、及びTUのサイズを含むビデオブロック情報、イントラモード予測に関する動きベクトル情報、ならびに有効係数マップ情報、即ちCABAC用の、有効係数の位置を示す1及び0のマップを含むシンタックス要素をエントロピー符号化してもよい。ビデオデコーダ30は、ビデオエンコーダ20の動作と本質的に対称的な形で動作し得る。

【0051】

図2は、ジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化するための技法を実施することのできる例示的なビデオエンコーダを示すブロック図である。ビデオエンコーダ20は、ビデオフレーム内のコード化単位のイントラコード化及びインターコード化を実行し得る。イントラコード化は、所与のビデオフレーム内のビデオの空間的冗長性を低減又は除去するために空間的予測に依拠する。インターコード化は時間的予測を利用して、ビデオシーケンスの隣接フレーム内のビデオの時間的冗長性を低減又は除去する。イントラモード(Iモード)は、幾つかの空間ベースの圧縮モードのいずれかを指すことがある。単方向予測(Pモード)、双方方向予測(Bモード)、又は一般化P/B予測(GPBモード)などのインターモードは、幾つかの時間ベースの圧縮モードのいずれかを指すことがある。

【0052】

図2の例では、ビデオエンコーダ20は、モード選択ユニット38と、予測ユニット40と、加算ユニット50と、変換ユニット52と、量子化ユニット54と、エントロピー符号化ユニット56と、参照フレームメモリ64とを含む。予測ユニット40は、動き推定ユニット42と、動き補償ユニット44と、イントラ予測ユニット46とを含む。ビデオブロック再構成のために、ビデオエンコーダ20はまた、逆量子化ユニット58と、逆変換ユニット60と、加算ユニット62とを含む。ブロック境界をフィルタ処理して再構成されたビデオからブロックキネスアーティファクトを除去するために、デブロックングフィルタ又は適合型ループフィルタ(ALF)及びサンプル適応型オフセット(SAO)(図2には示されていない)のような他のループ内フィルタを含めてよい。必要に応じて、デブロックングフィルタは、一般に、加算ユニット62の出力をフィルタ処理する。

【0053】

図2に示すように、ビデオエンコーダ20は、符号化すべきビデオフレーム又はスライス内のビデオブロックを受信する。フレーム又はスライスは、複数のビデオブロック又はCUに分割され得る。モード選択ユニット38は、誤差結果に基づいて、コード化モードのうちの1つ、即ちイントラモード又はインターモードを選択し得る。次いで、予測ユニット40は、得られたイントラコード化又はインターコード化された予測ブロックを、加算ユニット50に供給して残差ブロックデータを生成し、かつ加算ユニット62に供給して参照フレームにおける参照ブロックとして使用される符号化ブロックを再構成する。

【0054】

0061

予測ユニット40内のイントラ予測ユニット46は、コード化すべきビデオブロックと同じフレーム内の1つ以上の近接するブロックに対するビデオブロックのイントラ予測コード化を実行する。予測ユニット40内の動き推定ユニット42及び動き補償ユニット44は、参照フレームメモリ64に記憶された1つ以上の参照フレーム内の1つ以上の参照ブロックに対するビデオブロックのインター予測コード化を実行する。動き推定ユニット42と動き補償ユニット44とは、高度に統合され得るが、概念的な目的のために別々に示されている。動き推定ユニット42によって実行される動き推定は、ビデオブロックの動きを推定する動きベクトルを生成するプロセスである。動きベクトルは、例えば、参照フレーム内の参照ブロック又はPUに対する現在のビデオフレーム内のビデオブロック又はPUの変位を示し得る。参照ブロックは、絶対値差分和(SAD:sum of absolute difference)、2乗差分和(SSD:sum of square difference)、又は他の差分メトリックによって求められ得る画素差分に関して、符号化すべきビデオブロック又はPUに厳密に一致すると考えられるブロックである。

【0055】

0062

動き推定ユニット42は、計算された動きベクトルを動き補償ユニット44に送る。動き補償ユニット44によって実行される動き補償は、動き推定によって求められた動きベクトルに基づいて予測ブロックを取り込むか又は生成することを含んでもよい。ビデオエンコーダ20は、コード化されているビデオブロックから予測ブロックを減算することによって残差ビデオブロックを形成する。加算ユニット50は、この減算演算を実行する1つ以上の構成要素を表す。

【0056】

0063

動き補償ユニット44は、ビデオシーケンスレベル、ビデオフレームレベル、ビデオスライスレベル、ビデオCUレベル、又はビデオPUレベルのうちの1つ以上における予測情報を表すように定められたシンタックス要素を生成してもよい。例えば、動き補償ユニット44は、CU、PU、及びTUのサイズを含むビデオブロック情報、ならびにイントラモード予測に関する動きベクトル情報を含むシンタックス要素を生成してもよい。

【0057】

0064

ビデオエンコーダ20が現在のビデオブロックから予測ブロックを減算することによって残差ビデオブロックを形成した後、変換ユニット52は、残差ブロックから1つ以上のTUを形成してもよい。変換ユニット52は、離散コサイン変換(DCT)、整数変換、ウェーブレット変換、又は概念的に同様の線形変換などの変換をTUに適用して、残差変換係数を備えるビデオブロックを生成する。この変換では、残差ブロックを画素領域から周波数領域などの変換領域に変換し得る。より具体的には、TUは、変換が施される前に、画素領域に残差ビデオデータを備えてもよく、変換が施された後、残差ビデオデータを表す変換係数を周波数領域に備えてもよい。

【0058】

0065

幾つかの例では、変換ユニット52は2次元固定小数点変換を備え得る。変換ユニット52は、まずTU内の残差ビデオデータの行に1次元変換を施し、即ち、第1の方向に変換を施し、次いでTU内の残差ビデオデータの列に1次元変換を施し、即ち、第2の方向に変換を施すか、又はこの動作を逆の順序で実行することによってTUに2次元変換を適用することができる。一例として、TUは 32×32 TUを備えてもよい。変換ユニット52は、まずTU内の画素データの各行に32点1次元変換を施して中間変換係数の 32×32 TUを生成し、次にTU内の中間変換係数の各列に32点1次元変換を施して変換係数の 32×32 TUを生成してもよい。

【0059】

0066

ビデオエンコーダ20は、第1の方向の1次元変換をTU内の残差ビデオデータに施した後、第2の方向への1次元変換を施せるように中間変換係数をバッファリングする。上述のように、HEVC規格では、ビデオコード化効率を向上させるためにより大きい変換単位サイズ、例えば 32×32 から 128×128 までのサイズが提案されている。しかしながら、TUサイズを大きくすると、2次元変換の中間バッファリング要件が増大する。例えば、 32×32 TUの場合、ビデオエンコーダ20は、第1の方向への1次元変換の後1024個の中間変換係数をバッファリングする必要がある。

【0060】

0067

TUサイズを大きくする中間バッファリング要件を低減させるために、本開示で説明する技法は、2次元変換の各方向が適用された後に第1のサイズのTUに含まれる変換係数のうちの高周波数の変換係数のサブセットをゼロ化することを含む。このようにして、変換ユニット52は、TUの第1のサイズよりも小さい第2のサイズを有する保持される係数ブロックをTU内に生成することができる。

【0061】

0068

ゼロ化プロセスは、TU内の変換係数のサブセットの値をゼロに等しく設定することを備える。ゼロ化された変換係数は、計算されることも破棄されることもなく、その代わり、単にゼロに設定され、記憶すべき値も符号化すべき値も有さない。本開示によれば、ゼロ化された変換係数は一般に、TU内の保持される低周波数変換係数と比較して高周波数の変換係数である。高周波数の変換係数は、通常、符号化すべきビデオブロックと予測ブロックとの非常に小さい画素差分に相当する残差ビデオデータを表す。従って、高周波数の変換係数に含まれる残差ビデオデータを非常に少なくすることができ、各値をゼロに等しく設定しても、復号後のビデオ品質に対する影響は無視できる程度の影響である。

【0062】

0069

一例として、変換ユニット52は、第1の方向、例えば行方向の1次元変換を 32×32 TU内の残差ビデオデータに施し、変換から出力される中間変換係数のうちの2分の1をゼロ化してもよい。ビデオエンコーダ20は次いで、中間変換係数のうちの保持される2分の1の係数にバッファリングを施すだけでよい。次いで、変換ユニット52は、第2の方向、例えば列方向の1次元変換を 32×32 TU内の保持される中間変換係数に施し、変換から出力される保持される変換係数のうちの2分の1を再びゼロ化してもよい。このようにして、変換ユニット52は、本来 32×32 のサイズを有するTU内に 16×16 のサイズを有する有効係数の保持される係数ブロックを生成してもよい。

【0063】

0070

上述の例では、変換ユニット52は、 16×16 の保持される係数ブロック、即ちTUの元のサイズの4分の1の係数ブロックを生成するように構成された。他の場合には、変換ユニット52は、コード化プロセスのコード化複雑度要件に応じてゼロ化する係数の割合を高くするか又は低くすることによって異なるサイズを有する保持される係数ブロック

を生成するように構成されてもよい。さらに、場合によっては、変換ユニット 5 2 は、矩形領域を有する保持される係数ブロックを生成するように構成されてもよい。この場合、各技法では、まず矩形領域の短い方の辺（即ち、保持される変換係数がより少ない方）の方向に 1 次元変換を施すことによって中間バッファリング要件をさらに低減させることができる。このようにして、ビデオエンコーダ 2 0 は、矩形領域の長い方の辺の方向に 1 次元変換を施す前に中間変換係数のうちの 2 分の 1 よりも少ない数をバッファリングすることができる。方形領域の保持される係数ブロックと矩形領域の保持される係数ブロックに対するゼロ化プロセスについては、図 3 A 及び図 3 B に関してより詳しく説明する。

【 0 0 6 4 】

0071

変換ユニット 5 2 は、量子化ユニット 5 4 に得られた変換係数を送り得る。量子化ユニット 5 4 は、ビットレートをさらに低減するために変換係数を量子化する。量子化プロセスは、係数の一部又は全部に関連するビット深度を低減させることができる。量子化の程度は、量子化パラメータを調整することによって変更され得る。次いで、エントロピー符号化ユニット 5 6 又は量子化ユニット 5 4 は、量子化変換係数を含む TU のスキャンを実行し得る。エントロピー符号化ユニット 5 6 は、ジグザグスキャン又は TU のサイズに関連する別のスキャン順序を適用して量子化変換係数をスキャンし、エントロピー符号化することのできるシリアル化されたベクトルを生成することができる。

【 0 0 6 5 】

0072

第 2 のサイズを有する保持される係数ブロックを生成するように本来第 1 のサイズを有する TU の係数がゼロ化された一例では、エントロピー符号化ユニット 5 6 は、第 2 のサイズの TU 用のスキャン順序を使用して保持される係数をスキャンしてもよい。この場合、エントロピー符号化ユニット 5 6 は、本来 32×32 のサイズを有する TU 内の 16×16 のサイズを有する保持される係数ブロックに 16×16 スキャン順序を適用してもよい。第 2 のサイズを有する保持される係数ブロックを生成するように本来第 1 のサイズを有する TU の係数がゼロ化された別の例では、エントロピー符号化ユニット 5 6 は、保持される係数ブロックに含まれない TU の係数を飛ばすように修正された第 1 のサイズの TU 用のスキャン順序を使用して保持される係数をスキャンしてもよい。この場合、エントロピー符号化ユニットは、本来 32×32 のサイズを有する TU 内のゼロ化された全ての係数を飛ばすことによって 16×16 のサイズを有する保持される係数ブロックに 32×32 スキャン順序を適用してもよい。

【 0 0 6 6 】

0073

エントロピー符号化ユニット 5 6 は、量子化変換係数をスキャンして 1 次元ベクトルを形成した後、量子化変換係数のベクトルをエントロピーコード化する。例えば、エントロピー符号化ユニット 5 6 は、CABAC、CAVLC、又は別のエントロピーコード化技法のようなコンテキスト適応型エントロピーコード化を実行し得る。エントロピー符号化ユニット 5 6 によるエントロピー符号化の後、符号化されたビットストリームは、ビデオデコーダ 3 0 などのビデオデコーダに送信されるか、又は後で送信若しくは検索するためにアーカイブされ得る。

【 0 0 6 7 】

0074

エントロピー符号化ユニット 5 6 は、コンテキスト適応型エントロピーコード化を実行するために、例えば、近接する係数の値が非ゼロであるかどうかに関係し得るコンテキストモデルに従って各係数にコンテキストを割り当てる。エントロピー符号化ユニット 5 6 はまた、コンテキストモデル内の割り当てられたコンテキストに関連する係数用のコード化プロセスを決定する。次いで、エントロピー符号化ユニット 5 6 は、割り当てられたコンテキストに基づいて係数をエントロピー符号化してもよい。例えば、CABAC の場合、エントロピー符号化ユニット 5 6 は、ジョイントコンテキストモデル内の割り当てられ

たコンテキストに関連する係数の値（例えば、0又は1）の確率推定値を求めてよい。エントロピー符号化ユニット56は次いで、係数の実際の符号化された値に基づいてコンテキストモデル内の割り当てられたコンテキストに関連する確率推定値を更新する。

【0068】

0075

従来、ビデオエンコーダ20は、実装されるビデオ圧縮規格によって支持されるTUの様々なサイズの各々について別個のコンテキストモデルを維持する。ビデオコード化効率を向上させるために、例えば、 32×32 から 128×128 までの追加の変換単位サイズが提案されているが、追加のTUサイズによって、追加の変換単位サイズの各々にコンテキストモデルを維持するためにメモリ及び計算要件も増大する。場合によっては、TUサイズが大きくなるにつれて使用されるコンテキストが多くなることがあり、従って、より大きいTUサイズ用のより多くのコンテキストを維持するためにメモリ及び計算要件も増大する可能性がある。

【0069】

0076

本開示の技法によれば、エントロピー符号化ユニット56は、それぞれの異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用してビデオ係数のエントロピー符号化を実行するように構成されてもよい。各技法について、主としてCABACのエントロピー符号化技法に関して説明する。しかしながら、場合によっては、各技法が他のコンテキスト適応型エントロピー符号化技法に適用されてもよい。ジョイントコンテキストモデルを2つ又はそれ以上のサイズの変換単位間で共有すると、ビデオエンコーダ20上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させることができる。さらに、ジョイントコンテキストモデルを共有すると、ビデオスライスの開始時に全てのコンテキストモデルをリセットすることを含む、ビデオエンコーダ20上にコンテキストモデルを維持することについての計算コストを削減することもできる。CABACの場合、各技法は、係数の実際のコード化された値に基づいてコンテキストモデルの確率推定値を頻繁に更新することについての計算コストを削減することもできる。

【0070】

0077

本開示によれば、エントロピー符号化ユニット56は、それぞれ異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って変換単位の1つに関連する係数のコンテキストを選択してもよい。次いで、エントロピー符号化ユニット56は、選択されたコンテキストに基づいて変換単位内の有効係数をエントロピー符号化してもよい。一例として、ジョイントコンテキストモデルは、 32×32 の第1のサイズを有する変換単位と 16×16 の第2のサイズを有する変換単位との間で共有されてもよい。場合によっては、2つよりも多くのサイズの変換単位が同じジョイントコンテキストモデルを共有してもよい。一例では、ジョイントコンテキストモデルは、変換単位の有意性マップのジョイントコンテキストモデルであってもよい。他の例では、ジョイントコンテキストモデルを他のコード化情報又はシンタック要素に関連付けてもよい。ジョイントコンテキストモデルを使用するCABAC符号化プロセスについて図5に関してより詳しく説明する。

【0071】

0078

一例では、エントロピー符号化ユニット56は、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有するTUと本来第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを維持してもよい。場合によっては、保持される係数ブロックは、第2のサイズと等しいサイズを有してもよい。例えば、第1のサイズを有するTUにおいて高周波数係数がゼロ化されるとき、保持される係数ブロックに含まれる係数は、本来第2のサイズを有するTUに含まれる係数と同様の確率統計を有する。この場合、本来第1のサイズを有するTUの係数をゼロ化することによって第

2のサイズの保持される係数ブロックが生成されたときに、エントロピー符号化56は、ジョイントコンテキストモデルに従って保持される係数ブロックの係数のコンテキストを選択してもよい。次いで、エントロピー符号化ユニット56は、選択されたコンテキストに基づいて保持される係数ブロック内の有効係数をエントロピー符号化する。CABACの場合、エントロピー符号化ユニット56はまた、係数の実際のコード化された値に基づいてジョイントコンテキストモデル内の選択されたコンテキストに関連する確率推定値を更新する。

【0072】

0079

別の例では、エントロピー符号化ユニット56は、第1のサイズを有する第1のTUと第2のサイズを有する第2のTUとによって共有されるジョイントコンテキストモデルを維持してもよい。場合によっては、第1のサイズを有する第1のTUに含まれる係数は、第1のTU内の高周波数係数をゼロ化しない場合でも第2のサイズを有する第2のTUに含まれる係数と同様の確率統計を有してもよい。このことは、高周波数係数が、エントロピーコード化のための近接する係数の確率統計に対する影響が無視できるほど少ない残差ビデオデータを表し得るために可能になる。この例では、エントロピー符号化56は、ジョイントコンテキストモデルに従って、第1のサイズと第2のサイズのうちの一方のTU内の係数のコンテキストを選択してもよい。次いで、エントロピー符号化ユニット56は、選択されたコンテキストに基づいてTU内の有効係数をエントロピー符号化する。CABACの場合、エントロピー符号化ユニット56は次いで、係数の実際のコード化された値に基づいてジョイントコンテキストモデル内の選択されたコンテキストに関連する確率推定値を更新してもよい。

【0073】

0080

エントロピー符号化ユニット56はまた、符号化されているビデオブロックのための動きベクトルと他の予測シンタックス要素とをエントロピー符号化し得る。例えば、エントロピー符号化ユニット56は、本開示で説明する技法を使用してTU内の有効係数の位置を示す有意性マップ、即ち0と1のマップを示すシンタックス要素をエントロピー符号化してもよい。例えば、エントロピー符号化ユニット56は、符号化されたビットストリームで送信できるように動き補償ユニット44によって生成された適切なシンタックス要素を含むヘッダ情報を構築し得る。ビデオエンコーダ20は、CU、PU、及びTUのサイズを含むビデオブロック情報、並びにイントラモード予測に関する動きベクトル情報を示すシンタックス要素をエントロピー符号化してもよい。エントロピー符号化ユニット56は、シンタックス要素をエントロピー符号化するために、シンタックス要素を1つ以上のバイナリビットとして2値化し、コンテキストモデルに従って各バイナリビットのコンテキストを選択することによってCABACを実行してもよい。

【0074】

0081

逆量子化ユニット58及び逆変換ユニット60は、それぞれ逆量子化及び逆変換を適用して、残差ブロックを、参照フレームの参照ブロックとして後で使用できるように画素領域に再構成する。加算ユニット62は、再構成された残差ブロックを、動き補償ユニット44によって生成された予測ブロックに加算して、参照フレームメモリ64に記憶される参照ブロックを生成する。参照ブロックは、後続のビデオフレーム中のブロックをインター予測する(inter-predict)ために、動き推定ユニット42及び動き補償ユニット44によって参照ブロックとして使用され得る。

【0075】

0082

図3A及び図3Bは、第1のサイズを有する変換単位内の第2のサイズを有する保持される係数ブロックのそれぞれ方形領域及び矩形領域を示す概念図である。図3A及び図3Bの概念図において、それぞれ異なるボックスは、変換が施された後のTU内の変換係数

を表す。ハッチングされた領域内の係数は、保持される変換係数を備え、ハッチングされた領域に含まれない係数（即ち、白又は斜線のボックス）は、2次元変換時にゼロ化された係数を備える。

【0076】

0083

上述のように、本開示の技法は、2次元変換の各方向の後に第1のサイズのTUに含まれる変換係数のうちの高周波数の変換係数のサブセットをゼロ化することによって、より大きいTUサイズ、例えば 32×32 から 128×128 までの中間バッファリング要件を低減させる。このようにして、図2の変換ユニット52は、TUの第1のサイズよりも小さい第2のサイズを有する保持される係数ブロックをTU内に生成することができる。

【0077】

0084

ゼロ化プロセスは、TU内の変換係数のサブセットの値をゼロに等しく設定することを備える。ゼロ化された変換係数は、計算されることも破棄されることもなく、その代わり、ゼロ化された変換係数は、単にゼロに等しく設定され、記憶すべき値も符号化すべき値も有さない。本開示によれば、ゼロ化された変換係数は一般に、TU内の保持される変換係数と比較して高周波数の変換係数である。高周波数の変換係数は、符号化すべきビデオブロックと予測ブロックとの非常に小さい画素差分に相当する残差ビデオデータを表す。従って、高周波数の変換係数に含まれる残差ビデオデータを非常に少なくすることができ、各値をゼロに等しく設定しても、復号後のビデオ品質に対する影響は無視できる程度の影響である。

【0078】

0085

図3Aは、 16×16 の第1のサイズを有するTU70と、TU70内の、 8×8 の第2のサイズの最終的な方形領域を有する保持される係数ブロック74とを示す。保持される係数ブロック74のサイズ及び形状は、コード化プロセスのコード化複雑度要件に基づいて選択されてもよい。この例では、図2の変換ユニット52は、本来 16×16 のサイズを有するTU70内に 8×8 のサイズの方形領域を有する保持される係数ブロック74を生成するように構成されてもよい。変換ユニット52は、保持される係数ブロック74を生成するために、2次元変換の各方向、即ち行方向及び列方向を適用した後より周波数の高い係数のうちの2分の1をゼロ化してもよい。他の場合には、変換ユニット52は、コード化プロセスのコード化複雑度要件に応じてゼロ化する係数の割合を高くするか又は低くするように構成されてもよい。

【0079】

0086

まず、変換ユニット52は、TU70内の残差ビデオデータの各行に1次元変換を施し、変換から出力される中間変換係数のサブセット（この場合は2分の1）をゼロ化することができる。図3Aの図示の例では、保持される中間変換係数は、TU70の本来の 16×16 のサイズの2分の1に等しい 16×8 のサイズの矩形領域を有する保持される中間係数ブロック73（即ち、TU70内の斜線のブロック）に含まれる。ゼロ化後のサブセット（即ち、TU70内の白いブロック）は、TU70の保持される中間係数ブロック73内の係数よりも高い周波数値を有する係数を含んでもよい。図示の例では、変換ユニット52は、TU70の各行における8つの最高周波数値を有する係数のうちの2分の1をゼロ化する。このゼロ化プロセスでは、TU70内に 16×8 の矩形領域を有する保持される中間係数ブロック73が得られる。他の例では、保持される中間係数ブロック73の領域は異なるサイズ又は形状を備えてもよい。変換ユニット52は、 16×16 のTU70内の全ての係数のうちで最も高い周波数値を有する係数のうちの2分の1をゼロ化してもよい。このゼロ化プロセスによって、TU70の左上角部に三角形領域を有する保持される中間係数ブロックが得られる。

【0080】

0087

ビデオエンコーダ 20 は、第 1 の方向における変換から出力される中間変換係数のうちの 2 分の 1 をゼロ化することによって、第 2 の方向に変換を適用する前に、保持される中間係数ブロック 73 内に各係数をバッファリングするだけでよい。ゼロ化後のサブセット（即ち、TU 70 内の白いブロック）内の係数は、記憶すべき値も、変換すべき値も、符号化すべき値も有さない。このようにして、各技法は、2 次元変換を実行する際の中間バッファリング要件を低減させることができる。このことは、HEVC 規格に関して提案されているより大きい変換単位サイズ、例えば 32×32 から 128×128 までのサイズに特に有用である。

【0081】

0088

バッファリングに続いて、変換ユニット 52 は、保持される中間係数ブロック 73 内の保持される中間変換係数の各列に 1 次元変換を施し、変換から出力される変換係数のサブセット（この場合は 2 分の 1）をゼロ化してもよい。図 3A の図示の例では、保持される中間変換係数は、TU 70 の本来の 16×16 のサイズの 4 分の 1 に等しい 8×8 のサイズの方形領域を有する保持される係数ブロック 74（即ち、TU 70 内のハッチングされたブロック）に含まれる。ゼロ化後のサブセット（即ち、保持される中間係数ブロック 73 内のハッチングされていない方形領域）は、TU 70 の保持される係数ブロック 74 内の係数よりも高い周波数値を有する係数を含んでもよい。図示の例では、変換ユニット 52 は、保持される中間係数ブロック 73 の各列における 8 つの最高周波数値を有する係数のうちの 2 分の 1 をゼロ化する。他の例では、変換ユニット 52 は、 16×8 の保持される中間係数ブロック 73 内の全ての係数のうちで最も高い周波数値を有する係数のうちの 2 分の 1 をゼロ化してもよい。いずれの場合も、このゼロ化プロセスでは、TU 70 内に 8×8 の方形領域を有する保持される係数ブロック 74 が得られる。

【0082】

0089

図 3B は、 16×16 の第 1 のサイズを有する TU 76 と、TU 76 内の、 4×16 の第 2 のサイズの最終的な矩形領域を有する保持される係数ブロック 78 とを示す。保持される係数ブロック 78 のサイズ及び形状は、コード化プロセスのコード化複雑度要件に基づいて選択されてもよい。より具体的には、保持される係数ブロック 78 の最終的な矩形領域は、イントラコード化モード、スキャンパターン、及び保持される係数ブロック 78 に関する最後の有効係数の位置のうちの少なくとも 1 つに基づいて選択されてもよい。

【0083】

0090

図 3B の図示の例では、変換ユニット 52 は、本来 16×16 のサイズを有する TU 76 内に 4×16 のサイズの矩形領域を有する保持される係数ブロック 78 を生成するように構成されてもよい。変換ユニット 52 は、最初に、保持される係数ブロック 78 の最終的な矩形領域の最も短い辺（例えば、列）の方向において TU 76 に適用されてもよい。変換ユニット 52 は、保持される係数ブロック 78 を生成するために、2 次元変換を第 1 の方向（例えば、列方向）に施した後で係数のうちの 4 分の 3 をゼロ化してもよい。他の場合には、変換ユニット 52 は、コード化プロセスのコード化複雑度要件に応じてゼロ化する係数の割合を高くするか又は低くするように構成されてもよい。

【0084】

0091

保持される係数ブロック用に矩形が選択されると、各技法では、まず矩形領域の短い方の辺（即ち、保持される変換係数がより少ない方）の方向に 1 次元変換を施すことによって中間バッファリング要件をさらに低減させることができる。このようにして、ビデオエンコーダ 20 は、矩形領域の長い方の辺の方向に 1 次元変換を施す前に中間変換係数のうちの 2 分の 1 よりも少ない数をバッファリングすることができる。図 3B の図示の例では、保持される係数ブロック 78 の最終的な矩形領域の高さ (H) は、矩形領域の幅 (W)

よりも小さく、従って、垂直方向に保持される中間変換係数の方が少ない。従って、変換ユニット 52 はまず、TU76 の各行に変換が施される前にビデオエンコーダ20 がバッファリングする中間変換係数の数が 2 分の 1 よりも少なくなる（この場合には 4 分の 1）ように TU76 の各列に変換を施してもよい。

【0085】

0092

より具体的には、変換ユニット 52 は、TU76 内の残差ビデオデータの各列に 1 次元変換を施し、変換から出力される中間変換係数のサブセット（この場合は 4 分の 3）をゼロ化してもよい。図 3B の図示の例では、保持される中間変換係数は、TU76 の本来の 16×16 のサイズの 4 分の 1 に等しい 4×16 の矩形領域を有する保持される中間係数ブロック 77（即ち、TU76 内の斜線のブロック）に含まれる。ゼロ化後のサブセット（即ち、TU76 内の白いブロック）は、TU76 の保持される中間係数ブロック 77 内の係数よりも高い周波数値を有する係数を含んでもよい。図示の例では、変換ユニット 52 は、TU76 の各列における 12 個の最高周波数値を有する係数のうちの 4 分の 3 をゼロ化する。このゼロ化プロセスでは、TU76 内に 4×16 の矩形領域を有する保持される中間係数ブロック 77 が得られる。

【0086】

0093

ビデオエンコーダ20 は、第 1 の方向における変換から出力される中間変換係数のうちの 4 分の 3 をゼロ化することによって、第 2 の方向に変換を適用する前に、保持される中間係数ブロック 77 内に各係数をバッファリングするだけでよい。ゼロ化後のサブセット（即ち、TU76 内の白いブロック）内の係数は、記憶すべき値も、変換すべき値も、符号化すべき値も有さない。このようにして、各技法は、2 次元変換を実行する際の中間バッファリング要件を低減させることができる。このことは、HEVC 規格に関して提案されているより大きい変換単位サイズ、例えば 32×32 から 128×128 までのサイズに特に有用である。

【0087】

0094

バッファリング後、変換ユニット 52 は、保持される係数ブロック 77 内の係数の行に 1 次元変換を施してもよい。この例では、TU76 はすでに、 16×16 の本来のサイズの 4 分の 1 までゼロ化されているので、変換ユニット 52 は、変換から出力されるどの変換係数もゼロ化しなくてよい。図 3A の図示の例では、保持される変換係数は、保持される中間係数ブロック 77 と同じ 4×16 の矩形領域を有する保持される係数ブロック 78（即ち、TU76 内のハッチングされたブロック）に含まれる。このゼロ化プロセスでは、TU70 内に 4×16 の矩形領域を有する保持される係数ブロック 78 が得られる。

【0088】

0095

図 4 は、ジョイントコンテキストモデルを使用してビデオ係数をエントロピー復号するための技法を実施することのできる例示的なビデオデコーダを示すブロック図である。図 4 の例では、ビデオデコーダ 30 は、エントロピー復号ユニット 80 と、予測ユニット 81 と、逆量子化ユニット 86 と、逆変換ユニット 88 と、加算ユニット 90 と、参照フレームメモリ 92 とを含む。予測ユニット 81 は、動き補償ユニット 82 と、イントラ予測ユニット 84 とを含む。ビデオデコーダ 30 は、幾つかの例では、ビデオエンコーダ 20（図 2）に関して説明した符号化パスに一般的に相互関係にある復号パスを実行し得る。

【0089】

0096

復号プロセス中に、ビデオデコーダ 30 は、符号化されたビデオフレーム又はビデオスライスと、ビデオエンコーダ 20 などのビデオエンコーダからの符号化情報を表すシンタクス要素とを含む符号化されたビデオビットストリームを受信する。エントロピー復号ユニット 80 は、ビットストリームを復号して TU 内に量子化された変換係数を生成する

。例えば、エントロピー復号ユニット 80 は、CABAC、CAVLC、又は別のエントロピー符号化技法のようなコンテキスト適応型エントロピー復号を実行し得る。エントロピー復号ユニット 80 は、本開示で説明する技法を使用して TU 内の有効係数の位置を示す有意性マップ、即ち 1 と 0 のマップを含むシンタックス要素をエントロピー復号してもよい。エントロピー復号ユニット 80 は、動きベクトル及び他の予測シンタックス要素をエントロピー復号してもよい。

【0090】

0097

エントロピー復号ユニット 80 は、コンテキスト適応型エントロピー復号を実行するために、例えば、事前に復号された近接する係数の値が非ゼロであるかどうかに関係し得るコンテキストモデルに従って、ビットストリーム内に表された符号化された各係数にコンテキストを割り当てる。エントロピー復号ユニット 80 はまた、コンテキストモデル内の割り当てられたコンテキストに関連する符号化された係数用の符号化プロセスを決定する。次いで、エントロピー復号ユニット 80 は、割り当てられたコンテキストに基づいて係数をエントロピー復号してもよい。CABAC の場合、エントロピー復号ユニット 80 は、コンテキストモデル内の割り当てられたコンテキストに関連する符号化された係数の値（例えば、0 又は 1）の確率推定値を求めてよい。エントロピー復号ユニット 80 は次いで、係数の復号された実際の値に基づいてコンテキストモデル内の割り当てられたコンテキストに関連する確率推定値を更新する。

【0091】

0098

従来、ビデオデコーダ 30 は、実装されるビデオ圧縮規格によって支持される TU の様々なサイズの各々に別個のコンテキストモデルを維持しなければならない。HEVC 規格の場合、ビデオコード化効率を向上させるために、例えば、 32×32 から 128×128 までの追加の変換単位サイズが提案されているが、追加の TU サイズによって、追加の変換単位サイズの各々にコンテキストモデルを維持するためにメモリ及び計算要件も増大する。

【0092】

0099

本開示の技法によれば、エントロピー復号ユニット 80 は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用してビデオ係数のエントロピー復号を実行するように構成されてもよい。各技法について、主として CABAC のエントロピー復号技法に関して説明する。しかしながら、場合によっては、各技法が他のコンテキスト適応型エントロピーコード化技法に適用されてもよい。ジョイントコンテキストモデルを 2 つ又はそれ以上のサイズの変換単位間で共有すると、ビデオデコーダ 30 上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させることができる。さらに、ジョイントコンテキストモデルを共有すると、ビデオスライスの開始時に全てのコンテキストモデルをリセットすることを含む、ビデオデコーダ 30 上にコンテキストモデルを維持することについての計算コストを削減することもできる。CABAC の場合、各技法は、係数の実際のコード化された値に基づいてコンテキストモデルの確率推定値を頻繁に更新することについての計算コストを削減することもできる。

【0093】

0100

本開示によれば、エントロピー復号ユニット 80 は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを維持し、ジョイントコンテキストモデルに従って 1 つの変換単位に関連する係数のコンテキストを選択してもよい。次いで、エントロピー復号ユニット 80 は、選択されたコンテキストに基づいて変換単位内の有効係数をエントロピー復号してもよい。一例として、ジョイントコンテキストモデルは、 32×32 の第 1 のサイズを有する変換単位と 16×16 の第 2 のサイズを有する変換単位との間で共有されてもよい。場合によっては、2 つよりも多くのサイズの変換単位が同じ

ジョイントコンテキストモデルを共有してもよい。一例では、ジョイントコンテキストモデルは、変換単位の有意性マップのジョイントコンテキストモデルであってもよい。他の例では、ジョイントコンテキストモデルを他のコード化情報又はシンタックス要素に関連付けてもよい。ジョイントコンテキストモデルを使用するCABAC復号プロセスについて図6に関してより詳しく説明する。

【0094】

0101

一例では、エントロピー復号ユニット80は、保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有するTUと本来第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを維持してもよい。場合によっては、保持される係数ブロックは、第2のサイズと等しいサイズを有してもよい。例えば、この場合、エントロピー復号ユニット80は、ジョイントコンテキストモデルに従って、本来第1のサイズを有するTU内の第2のサイズを有する保持される係数ブロックの係数のコンテキストを選択してもよい。エントロピー復号ユニット80は次いで、選択されたコンテキストに基づいて、符号化された有効係数を第1のサイズを有するTU内の保持される係数ブロックに算術的に復号する。CABACの場合、エントロピー復号ユニット80はまた、係数の復号された実際の値に基づいてジョイントコンテキストモデル内の選択されたコンテキストに関連する確率推定値を更新する。

【0095】

0102

別の例では、エントロピー復号ユニット80は、第1のサイズを有する第1のTUと第2のサイズを有する第2のTUとによって共有されるジョイントコンテキストモデルを維持してもよい。この例では、エントロピー復号ユニット80は、ジョイントコンテキストモデルに従って、第1のサイズと第2のサイズのうちの一方のTUに関連する符号化された係数のコンテキストを選択してもよい。次いで、エントロピー復号ユニット80は、選択されたコンテキストに基づいて、符号化された有効係数をTUに復号する。CABACの場合、エントロピー復号ユニット80は次いで、係数の復号された実際の値に基づいてジョイントコンテキストモデル内の選択されたコンテキストに関連する確率推定値を更新してもよい。いずれの場合も、エントロピー復号ユニット80は、第1のサイズ又は第2のサイズのいずれかを有するTU内の復号された量子化変換係数を逆量子化ユニット86に転送する。

【0096】

0103

逆量子化ユニット86は、上述のようにエントロピー復号ユニット80によってTUに復号された量子化変換係数を逆量子化、即ち脱量子化する。逆量子化プロセスは、各ビデオブロック又はCUについてビデオエンコーダ20によって計算される量子化パラメータQPを使用して量子化の程度を求め、同様に、適用すべき逆量子化の程度を求めることが得る。逆変換ユニット88は、逆変換、例えば、逆DCT、逆整数変換、逆ウェーブレット変換、又は概念的に同様の逆変換プロセスをTU内の変換係数に適用して、画素領域において残差ビデオデータを生成する。

【0097】

0104

幾つかの例では、逆変換ユニット88は2次元固定小数点変換を備え得る。逆変換ユニット88は、まずTU内の変換係数の行に1次元変換を施し、次いでTU内の変換係数の列に1次元逆変換を施すか、又はこの動作を逆の順序で実行することによって、TUに2次元変換を施すことができる。ビデオデコーダ30は、第1の方向の1次元逆変換をTU内の各変換係数に施した後、中間残差データを、第2の方向への1次元逆変換を施せるようにバッファリングする。上述のように、HEVC規格では、ビデオコード化効率を向上させるためにより大きい変換単位サイズ、例えば 32×32 から 128×128 までのサイズが提案されている。しかしながら、TUサイズを大きくすると、2次元変換の中間バ

ツファーリング要件も増大する。

【0098】

0105

本開示で説明する各技法は、TUサイズを大きくする中間バッファリング要件を低減させるために、図2のビデオエンコーダ20によって、TUに含まれる変換係数のうちの高周波数の変換係数のサブセットをゼロ化することを含んでもよい。TU内のゼロ化された変換係数は、単にゼロに等しく設定され、記憶すべき値も、変換すべき値も、符号化すべき値も有さない。従って、エントロピー復号ユニット80は、通常第1のサイズを有するTU内の第2のサイズを有する保持される係数ブロックに関連する符号化された係数を表す符号化されたビットストリームを受信する。エントロピー復号ユニット80は、各係数を第1のサイズを有するTU内の保持される係数ブロックに復号する。その場合、TUは、第2のサイズの保持される係数ブロック内の係数とTU内の残りの係数を表すゼロとを含む。

【0099】

0106

このように、ビデオエンコーダ20において変換係数をゼロ化するプロセスは、ビデオデコーダ30において逆変換を実行する際にTUサイズを大きくする中間バッファリング要件を低減させることができる。一例として、逆変換ユニット88は、第1の方向、即ち行方向の1次元領域逆変換を 32×32 のサイズを有するTU内の 16×16 のサイズを有する保持される係数ブロック内の変換係数に施してもよい。行の逆変換の後で、ビデオデコーダ30は、TUの2分の1、即ち 32×16 の係数のみを備える保持される係数ブロック内の係数から変換された中間残差データをバッファリングするだけでよい。次いで、逆変換ユニット88は、第2の方向、例えば列方向の1次元逆変換をTU内の中間残差データに施してもよい。このようにして、逆変換ユニット88は、 16×16 のサイズを有する保持される係数ブロック内の残差データを含め、TU内の残りの残差データを表すゼロを加算することによって、本来 32×32 のサイズを有するTUを生成してもよい。

【0100】

0107

エントロピー復号ユニット80はまた、予測ユニット81に復号された動きベクトルと他の予測シンタックス要素とを転送する。ビデオデコーダ30は、ビデオ予測単位レベル、ビデオコード化単位レベル、ビデオスライスレベル、ビデオフレームレベル及び/又はビデオシーケンスレベルにおいてシンタックス要素を受信し得る。ビデオフレームがイントラコード化されたフレームとしてコード化されると、予測ユニット81のイントラ予測ユニット84は、現在のフレームの事前に復号されたブロックからのデータに基づいて現在のビデオフレームのビデオブロック用の予測データを生成する。ビデオフレームがインターフレームとしてコード化されると、予測ユニット81の動き補償ユニット82は、エントロピー復号ユニット80から受信された復号された動きベクトルに基づいて現在のビデオフレームのビデオブロックの予測ブロックを生成する。予測ブロックは、参照フレームメモリ92に記憶された参照フレームの1つ以上の参照ブロックに関して生成されてもよい。

【0101】

0108

動き補償ユニット82は、動きベクトルとその他の予測シンタックスとを構文解析することによって、復号すべきビデオブロックについての予測情報を求め、この予測情報を使用して、復号されている現在のビデオブロックのための予測ブロックを生成する。例えば、動き補償ユニット82は、受信されたシンタックス要素の幾つかを使用して、現在のフレームを符号化するために使用されるCUのサイズと、フレームの各CUがどのように分割されるかを記述する分割情報と、各分割がどのように符号化されるかを示すモード（例えば、イントラ予測又はインター予測）と、インター予測スライスタイル（例えば、Bスライス、Pスライス、又はGPBスライス）と、参照フレームリスト構成コマンドと、参

照フレームに適用される補間フィルタと、フレームの各ビデオブロックの動きベクトルと、動きベクトルに関するビデオパラメータ値と、現在のビデオフレームを復号するためのその他の情報を求める。

【0102】

ビデオデコーダ30は、逆変換ユニット88から得た残差ブロックに、動き補償ユニット82によって生成された対応する予測ブロックを加算することによって、復号ビデオブロックを形成する。加算ユニット90は、この加算演算を実行する1つ以上の構成要素を表す。必要に応じて、ブロックキネスアーティファクトを除去するために、デブロッキングフィルタを適用して、復号ブロックをフィルタ処理することもできる。復号されたビデオブロックは次いで、後で行われる動き補償のために参照フレームの参照ブロックを供給する参照フレームメモリ92に記憶される。参照フレームメモリ92はまた、図1の表示装置32などの表示装置上に表示される復号ビデオを生成する。

【0103】

図5は、ジョイントコンテキストモデルに従ってビデオ係数のコンテキストを選択するように構成された図2のエントロピー符号化ユニット56の例を示すブロック図である。エントロピー符号化ユニット56は、コンテキストモデル化ユニット94と、算術符号化ユニット96と、ジョイントコンテキストモデルストア98とを含む。上述のように、本開示の各技法は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用してビデオ係数のエントロピー符号化を実行することを対象としている。具体的に言うと、本明細書では各技法についてCABAC符号化プロセスに関して説明する。

【0104】

エントロピー符号化ユニット56は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルをジョイントコンテキストモデルストア98内に維持する。一例として、ジョイントコンテキストモデルストア98は、第2のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有するTUと本来第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。別の例として、ジョイントコンテキストモデルストア98は、第1のサイズを有するTU及び第2のサイズを有するTUによって共有されるジョイントコンテキストモデルを記憶してもよい。場合によっては、第1のサイズは32×32を備えてもよく、第2のサイズは16×16を備えてもよい。

【0105】

コンテキストモデル化ユニット94は、エントロピー符号化されるベクトルとしてスキヤンされた第1のサイズ又は第2のサイズのいずれかを有するTUに関連する変換係数を受信する。次いで、コンテキストモデル化ユニット94は、ジョイントコンテキストモデルに従って、TUの事前に符号化された近接する係数の値に基づいてTUの各係数にコンテキストを割り当てる。より具体的には、コンテキストモデル化ユニット94は、事前に符号化された近接する係数の値が非ゼロであるかどうかに応じてコンテキストを割り当てることができる。割り当てられるコンテキストは、ジョイントコンテキストモデル内のコンテキストインデックス、例えばコンテキスト(i)(i=0,1,2,、N)を指してもよい。

【0106】

係数にコンテキストが割り当てられた後、コンテキストモデル化ユニット94は、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する係数の値(例えば、0又は1)の確率推定値を求めてよい。それぞれの異なるコンテキストインデックスが、そのコンテキストインデックスを有する係数の値の確率推定値に関連付けられる。CABACの場合にコンテキストモデル化ユニット94によって実行される確率推定は、有限状態機械(FSM)を使用してテーブルドリブン推定器(table-driven estimator)に基づいてよい。FSMは、各コンテキストについて、過去のコンテキスト値を追跡し、所与の係数が0又は1の値を有する確率の最良推定値として現在の状態を供給することに

よって、関連する確率推定値を維持する。例えば、確率状態が0～127の範囲である場合、状態0は、係数が値0を有する確率が0.9999であることを意味してもよく、状態127は、係数が値0を有する確率が0.0001であることを意味してもよい。

【0107】

算術符号化ユニット96は、割り当てられたコンテキストに関連する係数の求められた確率推定値に基づいて係数を算術的に符号化する。このようにして、算術符号化ユニット96は、ジョイントコンテキストモデルに従って、第1のサイズ又は第2のサイズのいずれかを有するTUに関連する算術的に符号化された係数を表す符号化されたビットストリームを生成する。

【0108】

0115

符号化後、算術符号化ユニット96は、係数の実際の符号化された値をコンテキストモデル化ユニット94に送り返し、ジョイントコンテキストモデルストア98内において、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する確率推定値を更新する。コンテキストモデル化ユニット94は、確率状態間を遷移することによってジョイントコンテキストモデル内の割り当てられたコンテキストの確率更新を実行する。例えば、係数の実際のコード化された値が0である場合、より低い状態に遷移することによって係数値が0に等しい確率を高くしてもよい。ジョイントコンテキストモデルの確率推定値を頻繁に更新して係数の実際の符号化された値を反映することによって、ジョイントコンテキストモデル内の同じコンテキストに割り当てられる将来の係数の確率推定値の精度を高めることができ、それによって、算術符号化ユニット96によるビット符号化をさらに低減させることができる。

【0109】

第1の例では、ジョイントコンテキストモデルストア98は、第2のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有するTUと本来第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。例えば、ジョイントコンテキストモデルストア98は、本来 32×32 のサイズを有するTU内の 16×16 のサイズを有する保持される係数ブロックと本来 16×16 のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。

【0110】

32×32 のサイズを有する第1のTU内の 16×16 のサイズを有する保持される係数ブロック内の第1の係数に、 16×16 のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む 32×32 のサイズを有するTUと 16×16 のサイズを有するTUとによって共有されるジョイントコンテキストモデル内のコンテキスト(5)を割り当ててもよい。次いで、コンテキストモデル化ユニット94は、ジョイントコンテキストモデル内の割り当てられたコンテキスト(5)に関連する第1の係数の値の確率推定値を求め、第1の係数の実際の符号化された値に基づいてジョイントコンテキストモデル内の割り当てられたコンテキスト(5)に関連する確率推定値を更新する。 16×16 のサイズを有する第2のTU内の第2の係数に、第1のTU内の保持される係数ブロック内の係数と同じジョイントコンテキストモデル内のコンテキスト(5)を割り当ててもよい。次いで、コンテキストモデル化ユニット94は、ジョイントコンテキストモデル内の同じ割り当てられたコンテキスト(5)に関連する第2の係数の値の確率推定値を求め、第2の係数の実際の符号化された値に基づいてジョイントコンテキストモデル内の同じ割り当てられたコンテキスト(5)に関連する確率推定値を更新する。

【0111】

第2の例では、ジョイントコンテキストモデルストア98は、第1サイズを有するTUと第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。例えば、ジョイントコンテキストモデルストア98は、 32×32 のサイズを有するTUと 16×16 のサイズを有するTUとによって共有されるジョイントコンテ

キストモデルを記憶してもよい。 32×32 のサイズを有する第 1 の TU 内の第 1 の係数に、 32×32 のサイズを有する TU と 16×16 のサイズを有する TU とによって共有されるジョイントコンテキストモデル内のコンテキスト (5) を割り当ててもよい。次いで、コンテキストモデル化ユニット 94 は、ジョイントコンテキストモデル内の割り当てられたコンテキスト (5) に関する第 1 の係数の値の確率推定値を求め、第 1 の係数の実際の符号化された値に基づいてジョイントコンテキストモデル内の割り当てられたコンテキスト (5) に関する確率推定値を更新する。 16×16 のサイズを有する第 2 の TU 内の第 2 の係数に、第 1 の TU 内の係数と同じジョイントコンテキストモデル内のコンテキスト (5) を割り当ててもよい。次いで、コンテキストモデル化ユニット 94 は、ジョイントコンテキストモデル内の同じ割り当てられたコンテキスト (5) に関する第 2 の係数の値の確率推定値を求め、第 2 の係数の実際の符号化された値に基づいてジョイントコンテキストモデル内の同じ割り当てられたコンテキスト (5) に関する確率推定値を更新する。

【0112】

図 6 は、ジョイントコンテキストモデルに従ってビデオ係数のコンテキストを選択するように構成されたエントロピー復号ユニット 80 の例を示すブロック図である。エントロピー復号ユニット 80 は、算術符号化ユニット 102 と、コンテキストモデル化ユニット 104 と、ジョイントコンテキストモデルストア 106 とを含む。上述のように、本開示の各技法は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルを使用してビデオ係数のエントロピー復号を実行することを対象としている。具体的に言うと、本明細書では各技法について CABAC 復号プロセスに関して説明する。エントロピー復号ユニット 80 は、図 5 のエントロピー符号化ユニット 56 の動作と本質的に対称的な形で動作することができる。

【0113】

エントロピー復号ユニット 80 は、異なるサイズを有する変換単位同士の間で共有されるジョイントコンテキストモデルをジョイントコンテキストモデルストア 106 内に維持する。ジョイントコンテキストモデルストア 106 内に記憶されるジョイントコンテキストモデルは、図 5 のエントロピー符号化ユニット 56 のジョイントコンテキストモデルストア 98 に記憶されるジョイントコンテキストモデルと実質的に同様である。一例として、ジョイントコンテキストモデルストア 106 は、第 2 のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第 1 のサイズを有する TU と本来第 2 のサイズを有する TU とによって共有されるジョイントコンテキストモデルを記憶してもよい。別の例として、ジョイントコンテキストモデルストア 106 は、第 1 のサイズを有する TU と第 2 のサイズを有する TU とによって共有されるジョイントコンテキストモデルを記憶してもよい。場合によっては、第 1 のサイズは 32×32 を備えてもよく、第 2 のサイズは 16×16 を備えてもよい。

【0114】

算術復号ユニット 102 は、第 1 のサイズ又は第 2 のサイズのいずれかを有する TU に関連する符号化された変換係数を表す符号化されたビットストリームを受信する。算術復号ユニット 102 は、ビットストリームに含まれる第 1 の係数を復号する。次いで、コンテキストモデル化ユニット 104 は、第 1 の復号された係数の値に基づいてビットストリームに含まれる後続の符号化された係数にコンテキストを割り当てる。同様に、コンテキストモデル化ユニット 104 は、ジョイントコンテキストモデルに従って、TU の事前に復号された近接する係数の値に基づいてビットストリームに含まれる符号化された各係数にコンテキストを割り当てる。より具体的には、コンテキストモデル化ユニット 104 は、事前に復号された近接する係数の値が非ゼロであるかどうかに応じてコンテキストを割り当てることができる。割り当てられるコンテキストは、ジョイントコンテキストモデル内のコンテキストインデックスを指してもよい。

【0115】

符号化された係数にコンテキストが割り当てられた後、コンテキストモデル化ユニット

104は、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する符号化された係数の値（例えば、0又は1）の確率推定値を求めてよい。それぞれの異なるコンテキストインデックスが確率推定値に関連付けられる。コンテキストモデル化ユニット104は、符号化された係数の求められた確率推定値を算術復号ユニット102に送り返す。次いで、算術復号ユニット102は、割り当てられたコンテキストに関連する係数の求められた確率推定値に基づいて係数を算術的に復号する。このようにして、算術復号ユニット102は、ジョイントコンテキストモデルに従って、第1のサイズ又は第2のサイズのいずれかを有するTU内に、復号された変換係数を生成する。

【0116】

復号後、算術復号ユニット102は、係数の復号された実際の値をコンテキストモデル化ユニット104に送り返し、ジョイントコンテキストモデルストア106内において、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する確率推定値を更新する。ジョイントコンテキストモデルの確率推定値を頻繁に更新して係数の復号された実際の値を反映することによって、ジョイントコンテキストモデル内の同じコンテキストに割り当てられる将来の係数の確率推定値の精度を高めることができ、それによって、算術復号ユニット102によるビット復号をさらに低減させることができる。

【0117】

第1の例では、ジョイントコンテキストモデルストア106は、第2のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有するTUと本来第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。例えば、ジョイントコンテキストモデルストア106は、本来 32×32 のサイズを有するTU内の 16×16 のサイズを有する保持される係数ブロックと本来 16×16 のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。

【0118】

32×32 のサイズを有する第1のTU内の 16×16 のサイズを有する保持される係数ブロックに関連する第1の符号化された係数に、 16×16 のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む 32×32 のサイズを有するTUと 16×16 のサイズを有するTUとによって共有されるジョイントコンテキストモデル内のコンテキスト（5）を割り当ててもよい。次いで、コンテキストモデル化ユニット104は、ジョイントコンテキストモデル内の割り当てられたコンテキスト（5）に関連する第1の符号化された係数の値の確率推定値を求め、第1の係数の復号された実際の値に基づいてジョイントコンテキストモデル内の割り当てられたコンテキスト（5）に関連する確率推定値を更新する。 16×16 のサイズを有する第2のTUに関連する第2の符号化された係数に、第1のTU内の保持される係数ブロックに関連する第1の符号化された係数と同じジョイントコンテキストモデル内のコンテキスト（5）を割り当ててもよい。コンテキストモデル化ユニット104は次いで、ジョイントコンテキストモデル内の同じ割り当てられたコンテキスト（5）に関連する第2の符号化された係数の値の確率推定値を求め、第2の係数の復号された実際の値に基づいてジョイントコンテキストモデル内の同じ割り当てられたコンテキスト（5）に関連する確率推定値を更新する。

【0119】

第2の例では、ジョイントコンテキストモデルストア106は、第1サイズを有するTUと第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。例えば、ジョイントコンテキストモデルストア106は、 32×32 のサイズを有するTUと 16×16 のサイズを有するTUとによって共有されるジョイントコンテキストモデルを記憶してもよい。 32×32 のサイズを有する第1のTUに関連する第1の符号化された係数に、 32×32 のサイズを有するTUと 16×16 のサイズを有するTUとによって共有されるジョイントコンテキストモデル内のコンテキスト（5）を割り当ててもよい。次いで、コンテキストモデル化ユニット104は、ジョイントコンテキストモデル内の割り当てられたコンテキスト（5）に関連する第1の符号化された係数

の値の確率推定値を求め、第1の係数の復号された実際の値に基づいてジョイントコンテキストモデル内の割り当てられたコンテキスト(5)に関連する確率推定値を更新する。16×16のサイズを有する第2のTUに関連する第2の符号化された係数に、第1のTUに関連する第1の符号化された係数と同じジョイントコンテキストモデル内のコンテキスト(5)を割り当ててもよい。コンテキストモデル化ユニット104は次いで、ジョイントコンテキストモデル内の同じ割り当てられたコンテキスト(5)に関連する第2の符号化された係数の値の確率推定値を求め、第2の係数の復号された実際の値に基づいてジョイントコンテキストモデル内の同じ割り当てられたコンテキスト(5)に関連する確率推定値を更新する。

【0120】

図7は、第1のサイズを有する第1の変換単位と第2のサイズを有する第2の変換単位との間で共有されるジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化しかつエントロピー復号する例示的な動作を示すフローチャートである。図示の動作について、図2のビデオエンコーダ20内の図5のエントロピー符号化ユニット56、及び図3のビデオデコーダ30内の図6のエントロピー復号ユニット80を参照して説明するが、他の機器が同様の技法を実施してもよい。

【0121】

0128

図示の動作では、ビデオエンコーダ20内のエントロピー符号化ユニット56及びビデオデコーダ30内のエントロピー符号化ユニット80は、第1のサイズを有するTUと第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを維持してもよい。この場合、第1のサイズ、例えば32×32を有する第1のTUに含まれる係数は、第1のTU内の高周波数係数をゼロ化しない場合でも第2のサイズ、例えば16×16を有する第2のTUに含まれる係数と同様の確率統計を有してもよい。このことは、高周波数係数が、エントロピーコード化のための近接する係数の確率統計に対する影響が無視できるほど少ない残差ビデオデータを表すために可能になる。

【0122】

ある場合には、ビデオエンコーダ20は、変換ユニット52を使用して残差ビデオデータを、第1のサイズを有するTU内の変換係数に変換してもよい(120)。別の場合には、ビデオエンコーダ20は、変換ユニット52を使用して残差ビデオデータを、第2のサイズを有するTU内の変換係数に変換してもよい(121)。本開示の各技法は、TUが第1のサイズを有するかそれとも第2のサイズを有するかにかかわらず、エントロピー符号化ユニット56が同じジョイントコンテキストモデルに従ってTU内の係数をエントロピー符号化するのを可能にする。従って、これらの技法は、ビデオエンコーダ20上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオエンコーダ20上にコンテキストモデルを維持することについての計算コストを削減することができる。

【0123】

エントロピー符号化ユニット56内のコンテキストモデル化ユニット94は、第1のサイズと第2のサイズの両方を有するTUによって共有されるジョイントコンテキストモデルに従ってTU内の各係数のコンテキストを選択する(122)。より具体的には、コンテキストモデル化ユニット94は、ジョイントコンテキストモデルに従って、TUの事前に符号化された近接する係数の値に基づいてTUの所与の係数にコンテキストを割り当てる。コンテキストモデル化ユニット94は次いで、ジョイントコンテキストモデルリストア98において、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する係数の値(例えば、0又は1)の確率推定値を求めてよい。次に、算術符号化ユニット96は、係数の選択されたコンテキストに基づいて係数を算術的に符号化する(124)。

【0124】

符号化後、算術符号化ユニット96は、係数の実際の符号化された値をコンテキストモ

デル化ユニット94に送り返す。次いで、コンテキストモデル化ユニット94は、第1のサイズ又は第2のサイズのいずれかを有するTU内の係数の実際の符号化された値に基づいてジョイントコンテキストモデルの確率推定値を更新してもよい(126)。ビデオエンコーダ20は、第1のサイズ又は第2のサイズのいずれかを有するTUに関連する符号化された係数を表すビットストリームをビデオデコーダ30に送信する(128)。

【0125】

ビデオデコーダ30は、第1のサイズ又は第2のサイズのいずれかを有するTUに関連する符号化された変換係数を表す符号化されたビットストリームを受信してもよい(130)。本開示の各技法は、TUが第1のサイズを有するかそれとも第2のサイズを有するかにかかわらず、エントロピー復号ユニット80が同じジョイントコンテキストモデルに基づいてTUに関連する係数をエントロピー復号するのを可能にする。従って、これらの技法は、ビデオデコーダ30上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオデコーダ30上にコンテキストモデルを維持することについての計算コストを削減することができる。

【0126】

エントロピー復号ユニット80内のコンテキストモデル化ユニット104は、第1のサイズと第2のサイズの両方を有するTUによって共有されるジョイントコンテキストモデルに従ってTU内の各係数のコンテキストを選択する(132)。より具体的には、コンテキストモデル化ユニット104は、ジョイントコンテキストモデルに従って、TUの事前に符号化された近接する係数の値に基づいてTUに関連する後続の符号化された係数にコンテキストを割り当ててもよい。コンテキストモデル化ユニット104は次いで、ジョイントコンテキストモデルストア106内において、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する符号化された係数の値(例えば、0又は1)の確率推定値を求めてよい。コンテキストモデル化ユニット104は、符号化された係数の選択されたコンテキストに関連する確率推定値を算術復号ユニット102に送り返す。次に、算術符号化ユニット102は、選択されたコンテキストに基づいて、符号化された係数を第1のサイズ又は第2のサイズのいずれかを有するTUに算術的に復号する(134)。

【0127】

復号後、算術復号ユニット102は、係数の実際の符号化された値をコンテキストモデル化ユニット104に送り返す。次いで、コンテキストモデル化ユニット104は、第1のサイズ又は第2のサイズのいずれかを有するTU内の係数の復号された実際の値に基づいてジョイントコンテキストモデルの確率推定値を更新してもよい(136)。ある場合には、ビデオデコーダ30は、逆変換ユニット88を使用して、第1のサイズを有するTU内の係数を残差ビデオデータに逆変換してもよい(138)。別の場合には、ビデオデコーダ30は、逆変換ユニット88を使用して、第2のサイズを有するTU内の係数を残差ビデオデータに逆変換してもよい(139)。

【0128】

図8は、第2のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第1のサイズを有する第1の変換単位と第2のサイズを有する第2の変換単位との間で共有されるジョイントコンテキストモデルを使用してビデオ係数をエントロピー符号化しつつエントロピー復号する例示的な動作を示すフローチャートである。図示の動作について、図2のビデオエンコーダ20内の図4のエントロピー符号化ユニット56、及び図3のビデオデコーダ30内の図5のエントロピー復号ユニット80を参照して説明するが、他の機器が同様の技法を実施してもよい。

【0129】

図示の動作では、ビデオエンコーダ20内のエントロピー符号化ユニット56及びビデオデコーダ30内のエントロピー符号化ユニット80は、第1のサイズを有するTUと第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルを維持してもよい。この場合、第1のサイズ、例えば32×32のサイズを有する第1のTU内に第

2のサイズ、例えば 16×16 のサイズを有する保持される係数ブロックに含まれる係数は、第2のサイズ、例えば 16×16 のサイズを有する第2のTUに含まれる係数と同様の確率統計を有してもよい。

【0130】

ある場合には、ビデオエンコーダ20は、変換ユニット52を使用して残差ビデオデータを、第1のサイズを有するTU内の変換係数に変換してもよい(140)。ビデオエンコーダ20は、変換後に第1のTUに含まれる係数のサブセットをゼロ化して、第2のサイズを有する保持される係数ブロックを第1のTU内に生成する(141)。ゼロ化された変換係数のサブセットは通常、保持される係数ブロック内の係数と比較して高周波数の変換係数を含む。高周波数の変換係数に含まれる残差ビデオデータを非常に少なくすることができ、各値をゼロに等しく設定しても、復号後のビデオ品質に対する影響は無視できる程度の影響である。別の場合には、ビデオエンコーダ20は、変換ユニット52を使用して残差ビデオデータを、第2のサイズを有するTU内の変換係数に変換してもよい(142)。

【0131】

本開示の各技法は、TUが第1のサイズを有するかそれとも第2のサイズを有するかにかかわらず、エントロピー符号化ユニット56が同じジョイントコンテキストモデルに従ってTU内の保持される係数をエントロピー符号化するのを可能にする。従って、これらの技法は、ビデオエンコーダ20上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオエンコーダ20上にコンテキストモデルを維持することについての計算コストを削減することができる。

【0132】

エントロピー符号化ユニット56内のコンテキストモデル化ユニット94は、第2のサイズにゼロ化された係数を含む第1のサイズを有するTUと第2のサイズを有するTUとによって共有されるジョイントコンテキストモデルに従ってTU内の保持される各係数のコンテキストを選択する(144)。より具体的には、コンテキストモデル化ユニット94は、ジョイントコンテキストモデルに従って、保持される係数ブロックの事前に符号化された近接する係数の値に基づいて第1のTUの保持される係数ブロック内の所与の係数にコンテキストを割り当てる。コンテキストモデル化ユニット94は次いで、ジョイントコンテキストモデルストア98内において、ジョイントコンテキストモデル内の割り当られたコンテキストに関連する係数の値(例えば、0又は1)の確率推定値を求めてよい。次に、算術符号化ユニット96は、係数の選択されたコンテキストに基づいて係数を算術的に符号化する(146)。

【0133】

符号化後、算術符号化ユニット96は、係数の実際の符号化された値をコンテキストモデル化ユニット94に送り返す。次いで、コンテキストモデル化ユニット94は、第1のサイズのTU内の第2のサイズの保持される係数ブロック内又は本来第2のサイズを有するTU内の係数の実際の符号化された値に基づいてジョイントコンテキストモデルの確率推定値を更新してもよい(148)。ビデオエンコーダ20は、第1のサイズのTU内の保持される係数ブロック又は第2のサイズのTUに関連する符号化された係数を表すビットストリームをビデオデコーダ30に送信する(150)。

【0134】

次いで、ビデオデコーダ30は、第1のサイズのTU内の第2のサイズの保持される係数ブロック又は本来第2のサイズを有するTUに関連する符号化された係数を表すビットストリームを受信してもよい(152)。本開示の各技法は、TUが第1のサイズを有するかそれとも第2のサイズを有するかにかかわらず、エントロピー復号ユニット80が同じジョイントコンテキストモデルに基づいてTUに関連する係数をエントロピー復号するのを可能にする。従って、これらの技法は、ビデオデコーダ30上にコンテキスト及び確率を記憶するのに必要なメモリの量を低減させ、ビデオデコーダ30上にコンテキストモデルを維持することについての計算コストを削減することができる。

【 0 1 3 5 】

エントロピー復号ユニット 80 内のコンテキストモデル化ユニット 104 は、第 2 のサイズを有する保持される係数ブロックを生成するためにゼロ化された係数を含む第 1 のサイズを有する TU と第 2 のサイズを有する TU とによって共有されるジョイントコンテキストモデルに従って TU に関連する各係数のコンテキストを選択する (154)。より具体的には、コンテキストモデル化ユニット 104 は、ジョイントコンテキストモデルに従って、保持される係数ブロックの事前に復号された近接する係数の値に基づいて第 1 の TU の保持される係数ブロックに関連する後続の符号化された係数にコンテキストを割り当ててもよい。コンテキストモデル化ユニット 104 は次いで、ジョイントコンテキストモデルストア 106 内において、ジョイントコンテキストモデル内の割り当てられたコンテキストに関連する符号化された係数の値 (例えば、0 又は 1) の確率推定値を求めてよい。コンテキストモデル化ユニット 104 は、符号化された係数の選択されたコンテキストに関連する求められた確率を算術符号化ユニット 102 に送り返す。次に、算術符号化ユニット 102 は、選択されたコンテキストに基づいて、符号化された係数を第 1 のサイズの TU 内の保持される係数ブロック又は第 2 のサイズの TU に算術的に復号する (156)。

【 0 1 3 6 】

復号後、算術復号ユニット 102 は、係数の復号された実際の値をコンテキストモデル化ユニット 104 に送り返す。次いで、コンテキストモデル化ユニット 104 は、第 1 のサイズの TU 内の第 2 のサイズの保持される係数ブロック内又は本来第 2 のサイズを有する TU 内の係数の復号された実際の値に基づいてジョイントコンテキストモデルの確率推定値を更新してもよい (158)。ある場合には、ビデオデコーダ 30 は、逆変換ユニット 88 を使用して、第 1 のサイズを有する TU 内の第 2 のサイズを有する保持される係数ブロックの係数を残差ビデオデータに逆変換してもよい (160)。このようにして、逆変換ユニット 88 は、第 2 のサイズを有する保持される係数ブロック内の残差データを含め、TU 内の残りの残差データを表すゼロを加算することによって、第 1 のサイズを有する TU を生成してもよい。別の場合には、ビデオデコーダ 30 は、逆変換ユニット 88 を使用して、第 2 のサイズを有する TU 内の係数を残差ビデオデータに逆変換してもよい (162)。

【 0 1 3 7 】

1 つ以上の例では、説明した機能は、ハードウェア、ソフトウェア、ファームウェア、又はそれらの任意の組合せで実装され得る。各機能は、ソフトウェアで実装された場合、1 つ以上の命令又はコードとしてコンピュータ可読媒体上に記憶されるか、若しくはコンピュータ可読媒体を介して送信され、ハードウェアベースの処理ユニットによって実行され得る。コンピュータ可読媒体は、データ記憶媒体などの有形媒体、又は例えば、通信プロトコルに従って、ある場所から別の場所へのコンピュータプログラムの転送を可能にする任意の媒体を含む通信媒体に対応するコンピュータ可読記憶媒体を含み得る。このようにして、コンピュータ可読媒体は、概して、(1) 非一時的である有形コンピュータ可読記憶媒体、又は(2) 信号又は搬送波などの通信媒体に対応し得る。データ記憶媒体は、本開示で説明した技法の実装のための命令、コード及び / 又はデータ構造を取り出すために 1 つ以上のコンピュータ若しくは 1 つ以上のプロセッサによってアクセスされ得る任意の利用可能な媒体であり得る。コンピュータプログラム製品はコンピュータ可読媒体を含み得る。

【 0 1 3 8 】

限定ではなく例として、そのようなコンピュータ可読記憶媒体は、RAM、ROM、EEPROM、CD-ROM 又は他の光ディスクストレージ、磁気ディスクストレージ、又は他の磁気ストレージ機器、フラッシュメモリ、若しくは命令又はデータ構造の形態の所望のプログラムコードを記憶するために使用され得、コンピュータによってアクセスされ得る、任意の他の媒体を備えることができる。さらに、あらゆる接続をコンピュータ可読媒体と呼ぶことも妥当である。例えば、命令が、同軸ケーブル、光ファイバーケーブル、

ツイストペア、デジタル加入者回線 (D S L) 、又は赤外線、無線、及びマイクロ波などのワイヤレス技術を使用して、ウェブサイト、サーバ、又は他のリモートソースから送信される場合、同軸ケーブル、光ファイバーケーブル、ツイストペア、D S L、又は赤外線、無線、及びマイクロ波などのワイヤレス技術は、媒体の定義に含まれる。但し、コンピュータ可読記憶媒体及びデータ記憶媒体は、接続、搬送波、信号、又は他の一時媒体を含まないが、代わりに非一時的有形記憶媒体を対象とすることを理解されたい。本明細書で使用するディスク (disk) 及びディスク (disc) は、コンパクトディスク (disc) (C D) 、レーザディスク (disc) 、光ディスク (disc) 、デジタル多用途ディスク (disc) (D V D) 、フロッピー (登録商標) ディスク (disk) 及びブルーレイディスク (disc) を含み、ディスク (disk) は、通常、データを磁気的に再生し、ディスク (disc) は、データをレーザで光学的に再生する。上記の組合せもコンピュータ可読媒体の範囲内に含めるべきである。

【 0 1 3 9 】

命令は、1つ以上のデジタル信号プロセッサ (D S P) などの1つ以上のプロセッサ、汎用マイクロプロセッサ、特定用途向け集積回路 (A S I C) 、フィールドプログラマブル論理アレイ (F P G A) 、あるいは他の等価な集積回路又はディスクリート論理回路によって実行され得る。従って、本明細書で使用する「プロセッサ」という用語は、前述の構造、又は本明細書で説明する技法の実装に好適な他の構造のいずれかを指す。さらに、幾つかの態様では、本明細書で説明した機能は、符号化及び復号のために構成された専用のハードウェア及び/又はソフトウェアモジュール内に設けられ得、あるいは複合コードックに組み込まれ得る。また、各技法は、1つ以上の回路又は論理要素中に十分に実装され得る。

【 0 1 4 0 】

本開示の技法は、ワイヤレスハンドセット、集積回路 (I C) 又はI Cのセット (例えば、チップセット) を含む、多種多様な機器又は装置において実施され得る。本開示では、開示する技法を実行するように構成された機器の機能的態様を強調するために様々な構成要素、モジュール、又はユニットについて説明したが、それらの構成要素、モジュール、又はユニットを、必ずしも異なるハードウェアユニットによって実現する必要はない。むしろ、上記で説明したように、様々なユニットは、好適なソフトウェア及び/又はファームウェアとともに、上記で説明したように1つ以上のプロセッサを含む、コーデックハードウェアユニットとして組み合わせられるか、又は相互動作ハードウェアユニットの集合によって実現され得る。