

(19)日本国特許庁(JP)

(12)公表特許公報(A)

(11)公表番号
特表2025-511538
(P2025-511538A)

(43)公表日 令和7年4月16日(2025.4.16)

(51)国際特許分類
H 0 4 N 19/85 (2014.01)

F I
H 0 4 N 19/85

テーマコード (参考)
5 C 1 5 9

審査請求 未請求 予備審査請求 未請求 (全87頁)

(21)出願番号	特願2024-554996(P2024-554996)	(71)出願人	000001007 キャノン株式会社 東京都大田区下丸子3丁目30番2号
(86)(22)出願日	令和5年4月11日(2023.4.11)	(74)代理人	110003281 弁理士法人大塚国際特許事務所
(85)翻訳文提出日	令和6年11月7日(2024.11.7)	(72)発明者	ロゼワーン, クリストファー ジェームズ オーストラリア国 2 1 1 3 ニューサウスウェールズ州 タラベラロード パーク エステイト5 ビルディング エー キャノン オーストラリア プロプライエタリー リミテッド内
(86)国際出願番号	PCT/AU2023/050291	(72)発明者	グエン, ティ ホン ニュン オーストラリア国 2 1 1 3 ニューサウスウェールズ州 タラベラロード パーク エステイト5 ビルディング エー キャノン オーストラリア プロプライエタリー リミテッド内
(87)国際公開番号	WO2023/197031		
(87)国際公開日	令和5年10月19日(2023.10.19)		
(31)優先権主張番号	2022202471		
(32)優先日	令和4年4月13日(2022.4.13)		
(33)優先権主張国・地域又は機関	オーストラリア(AU)		
(81)指定国・地域	AP(BW,CV,GH,GM,KE,LR,LS,MW,MZ,NA,RW,SD,SL,ST,SZ,TZ,UG,ZM,ZW),EA(AM,AZ,BY,KG,KZ,RU,TJ,TM),EP(AL,AT,BE,BG,CH,CY,CZ,DE,DK,EE,ES,FI,FR,GB,GR,HR,HU,IE,IS,IT,LT,LU,LV)		

(54)【発明の名称】 テンソルを符号化および復号するための方法、装置、およびシステム

(57)【要約】

テンソルをビットストリームに符号化するシステムおよび方法であって、テンソルはチャンネル次元を有する。本方法は、テンソルの平均特徴を生成することと、平均特徴はチャンネル次元にわたるテンソルの平均であり、生成された平均特徴を量子化して量子化平均特徴マップを生成することと、量子化平均特徴マップの再構成を逆量子化して平均特徴の第2表現を生成することを含む。本方法は、平均特徴マップの第2表現およびテンソルのチャンネル次元を使用して平均特徴係数を決定することと、少なくとも量子化平均特徴マップおよび平均特徴係数の量子化バージョンをビットストリームに符号化することと、をさらに含む。

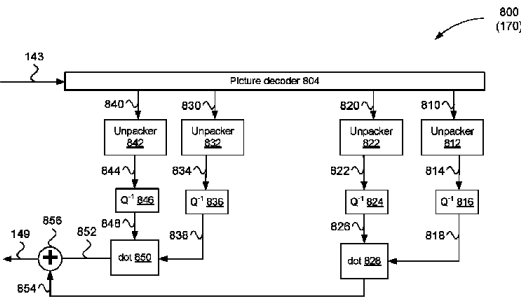


Fig. 8

【特許請求の範囲】

【請求項 1】

テンソルをビットストリームに符号化する方法であって、前記テンソルはチャンネル次元を有し、前記方法は、

前記テンソルの平均特徴を生成することと、前記平均特徴は、前記チャンネル次元にわたる前記テンソルの平均であり、

前記生成された平均特徴を量子化して量子化平均特徴マップを生成することと、

前記量子化平均特徴マップの再構成を逆量子化して前記平均特徴の第 2 表現を生成することと、

前記平均特徴マップの前記第 2 表現および前記テンソルの前記チャンネル次元を使用して平均特徴係数を決定することと、

少なくとも前記量子化平均特徴マップおよび前記平均特徴係数の量子化バージョンを前記ビットストリームに符号化することと

を含む方法。

【請求項 2】

前記符号化ステップは、VVC 符号化を使用して実施され、前記量子化平均特徴マップの前記再構成では、VVC 符号化を使用することによる損失がモデル化される、請求項 1 に記載の方法。

【請求項 3】

前記符号化ステップは、HEVC 符号化を使用して実施され、前記量子化平均特徴マップの前記再構成では、HEVC 符号化を使用することによる損失がモデル化される、請求項 1 に記載の方法。

【請求項 4】

量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法であって、前記方法は、

前記ビットストリームから、前記量子化機能を使用して生成される量子化平均特徴を復号することと、

前記量子化平均特徴を逆量子化して、前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成される逆量子化平均特徴を導出することと、

前記ビットストリームから平均特徴係数を復号することと、

少なくとも前記逆量子化平均特徴および前記平均特徴係数を使用して前記特徴マップを復号することと

を含む方法。

【請求項 5】

テンソルをビットストリームに符号化する方法であって、前記方法は、

基底ベクトルのセットを量子化して量子化基底ベクトルを生成することと、

前記量子化基底ベクトルの再構成を逆量子化して前記基底ベクトルの第 2 表現を生成することと、

前記基底ベクトルの前記第 2 表現および前記テンソルの前記チャンネル次元を使用して基底ベクトル係数を決定することと、

少なくとも前記基底ベクトル係数および前記基底ベクトルの量子化バージョンを前記ビットストリームに符号化することと

を含む方法。

【請求項 6】

前記符号化ステップは、VVC 符号化を使用して実施され、前記基底ベクトルモデルの前記再構成は、VVC 符号化を使用することによる損失をモデル化する、請求項 1 に記載の方法。

【請求項 7】

前記符号化ステップは、HEVC 符号化を使用して実施され、前記基底ベクトルモデルの前記再構成は、HEVC 符号化を使用することによる損失をモデル化する、請求項 1 に記

10

20

30

40

50

載の方法。

【請求項 8】

量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法であって、前記方法は、

前記量子化機能を使用して生成された量子化基底ベクトルのセットを前記ビットストリームから復号することと、

前記量子化基底ベクトルを逆量子化して逆量子化基底ベクトルを導出することと、前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成される逆量子化平均特徴、

前記ビットストリームから基底ベクトル係数を復号することと、

少なくとも前記逆量子化基底ベクトルおよび前記基底ベクトル係数を使用して前記特徴マップを復号することと、を含む方法。

10

【請求項 9】

テンソルをビットストリームに符号化するためのエンコーダであって、前記テンソルはチャンネル次元を有し、前記エンコーダは、

前記テンソルの平均特徴を生成し、前記平均特徴は、前記チャンネル次元にわたる前記テンソルの平均であり、

前記生成された平均特徴を量子化して量子化平均特徴マップを生成し、

前記量子化平均特徴マップの再構成を逆量子化して前記平均特徴の第 2 表現を生成し、

前記平均特徴マップの前記第 2 表現および前記テンソルの前記チャンネル次元を使用して平均特徴係数を決定し、

20

少なくとも前記量子化平均特徴マップおよび前記平均特徴係数の量子化バージョンを前記ビットストリームに符号化する

ように構成されているエンコーダ。

【請求項 10】

テンソルをビットストリームに符号化する方法を実行するためのプログラムを格納する非一時的コンピュータ可読記憶媒体であって、前記方法は、

基底ベクトルのセットを量子化して量子化基底ベクトルを生成することと、

前記量子化基底ベクトルの再構成を逆量子化して前記基底ベクトルの第 2 表現を生成することと、

30

前記基底ベクトルの前記第 2 表現および前記テンソルの前記チャンネル次元を使用して基底ベクトル係数を決定することと、

少なくとも前記基底ベクトル係数および前記基底ベクトルの量子化バージョンを前記ビットストリームに符号化することと

を含む非一時的コンピュータ可読記憶媒体。

【請求項 11】

システムであって、

メモリと、

テンソルをビットストリームに符号化する方法を実装するために、前記メモリに記憶されたコードを実行するように構成され、前記テンソルはチャンネル次元を有する、プロセッサと

40

を有し、前記方法は、

前記テンソルの平均特徴を生成することと、前記平均特徴は、前記チャンネル次元にわたる前記テンソルの平均であり、

前記生成された平均特徴を量子化して量子化平均特徴マップを生成することと、

前記量子化平均特徴マップの再構成を逆量子化して前記平均特徴の第 2 表現を生成することと、

前記平均特徴マップの前記第 2 表現および前記テンソルの前記チャンネル次元を使用して平均特徴係数を決定することと、

少なくとも前記量子化平均特徴マップおよび前記平均特徴係数の量子化バージョンを前

50

記ビットストリームに符号化することと
を含むシステム。

【請求項 1 2】

量子化機能を使用して符号化されたビットストリームから特徴マップを復号するためのデ
コーダであって、前記デコーダは、

前記ビットストリームから、前記量子化機能を使用して生成される量子化平均特徴を復
号し、

前記量子化平均特徴を逆量子化して、前記ビットストリームを符号化するために使用さ
れる前記量子化機能の逆を使用して生成される逆量子化平均特徴を導出し、

前記ビットストリームから平均特徴係数を復号し、

少なくとも前記逆量子化平均特徴および前記平均特徴係数を使用して前記特徴マップを
復号する

ように構成されているデコーダ。

【請求項 1 3】

量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法を実
行するためのプログラムを格納する非一時的コンピュータ可読記憶媒体であって、前記方
法は、

前記量子化機能を使用して生成された量子化基底ベクトルのセットを前記ビットストリ
ームから復号することと、

前記量子化基底ベクトルを逆量子化して逆量子化基底ベクトルを導出することと、前記
逆量子化平均特徴は前記ビットストリームを符号化するために使用される前記量子化機能
の逆を使用して生成され、

前記ビットストリームから基底ベクトル係数を復号することと、

少なくとも前記逆量子化基底ベクトルおよび前記基底ベクトル係数を使用して前記特徴
マップを復号することと、を含む非一時的コンピュータ可読記憶媒体。

【請求項 1 4】

システムであって、

メモリと、

量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法を
実装するために前記メモリ上に記憶されたコードを実行するように構成されるプロセッサ
と

を有し、前記方法は、

前記ビットストリームから、前記量子化機能を使用して生成される量子化平均特徴を復
号することと、

前記量子化平均特徴を逆量子化して、前記ビットストリームを符号化するために使用さ
れる前記量子化機能の逆を使用して生成される逆量子化平均特徴を導出することと、

前記ビットストリームから平均特徴係数を復号することと、

少なくとも前記逆量子化平均特徴および前記平均特徴係数を使用して前記特徴マップを
復号することと

を含むシステム。

【発明の詳細な説明】

【技術分野】

【0001】

関連出願への言及

本出願は2022年4月13日に提出されたオーストラリア国特許出願第202220
2471号の出願日の35U.S.C. § 119に基づく利益を主張し、その全体が本明細
書に完全に記載されているかのように、参照により本明細書に組み込まれる

本発明は一般に、デジタルビデオ信号処理に関し、特に、畳み込みニューラルネットワ
ークからテンソルを符号化および復号するための方法、装置、およびシステムに関する。

本発明はまた、ビデオ圧縮技術を使用して畳み込みニューラルネットワークからテンソル

10

20

30

40

50

を符号化および復号するためのコンピュータプログラムを記録したコンピュータ可読媒体を含むコンピュータプログラム製品に関する。

【背景技術】

【0002】

ビデオ圧縮は、ビデオデータの送信および記憶のためのアプリケーションを含む、多くのアプリケーションをサポートするために使用されるユビキタス技術である。多くのビデオコーディング規格が開発されており、他のものは現在開発中である。ビデオコーディング標準化における最近の発展は、「Joint Video Experts Team」(JVET)と呼ばれるグループの形成につながった。Joint Video Experts Team (JVET) は、2つの標準設定組織 (Standards Setting Organisations) (SSO)、つまり、「Video Coding Experts Group」(VCEG)としても知られる、International Telecommunication Union (ITU) の Telecommunication Standardisation Sector (ITU-T) の Study Group 16、Question 6 (SG16/Q6) と、「Moving Picture Experts Group」(MPEG)としても知られる、国際標準化機構／国際電気標準会議合同技術委員会 1／小委員会 29／Working Group 11 (ISO/IEC JTC1/SC29/WG11) と、のメンバを含む。

10

【0003】

Joint Video Experts Team (JVET) は、「versatile video coding」(VVC) と呼ばれるビデオ圧縮規格を開発した。

【0004】

20

畳み込みニューラルネットワーク (CNN) は、とりわけ、マシンビジョンを伴うユースケースに対処する新興技術である。CNN は典型的には全結合レイヤを含む畳み込みレイヤなどの多くのレイヤを含み、データは、「テンソル」の形態で1つのレイヤから次のレイヤに渡される。各レイヤの重みは、トレーニング段階で決定され、ここで、非常に大量のトレーニングデータがCNNを通過し、決定された結果がトレーニングデータに関連するグラウンドトゥースと比較される。確率的勾配降下などのネットワーク重みを更新するためのプロセスは、ネットワークが所望のレベルの精度で実行するまで、ネットワーク重みを反復的に精緻化するために適用される。畳み込みステージが1より大きい「ストライド」を有する場合、畳み込みからの出力テンソルは、対応する入力テンソルよりも低い空間分解能を有する。「max pooling」(または「Max pool」)などの演算も、入力テンソルと比較して出力テンソルの空間サイズを縮小する。最大プーリングは入力テンソルをデータサンプルのグループ(たとえば、データサンプルの2×2グループ)に分割することによって出力テンソルを生成し、各グループから、出力テンソル内の対応する値のための出力として最大値を選択する。入力を用いてCNNを実行し、入力を出力に漸進的に変換するプロセスは、一般に「推論」と呼ばれる。

30

【0005】

一般に、テンソルは4つの次元、すなわち、バッチ、チャネル、高さおよび幅を有する。ビデオデータを推論するときのサイズ「1」の第1次元「バッチ」は、一度に1つのフレームがCNNを通過することを示す。ネットワークをトレーニングするとき、所定の「バッチサイズ」に従って、ネットワークの重みが更新される前に複数のフレームがネットワークを通過するように、バッチ次元の値を増加させることができる。マルチフレームビデオは、所与のビデオのフレームの数に従ってサイズが増加されたバッチ次元を有する単一のテンソルとして通過され得る。しかしながら、メモリ消費およびアクセスに関する実地的な考慮事項のために、ビデオデータの推論は、典型的にはフレーム単位で実行される。「チャネル」次元は、所与のテンソルに対する同時 (concurrent) の「特徴マップ」の数を示す。高さおよび幅の次元は、CNNの特定のステージにおける特徴マップのサイズを示す。チャネルカウントは、ネットワークアーキテクチャに従ってCNNを介して変化する。また、特徴マップサイズは、特定のネットワークレイヤで発生するサブサンプリングに応じて変化する。

40

【0006】

50

CNNの第1レイヤへの入力は、典型的には第1レイヤへのテンソル入力の次元との互換性のためにリサイズされる、画像またはビデオフレームである。テンソルの次元はCNNアーキテクチャに依存し、一般に、入力幅および高さに関連するいくつかの次元と、さらなる「チャンネル」次元と、を有する。

【0007】

チャンネル次元に基づいてテンソルをスライスすることはいわゆる「特徴マップ」のセットをもたらし、これはテンソルの各スライスが対応する入力画像と何らかの関係を有し、様々なエッジタイプなどの特性を捕捉するためである。ネットワークへの入力からさらに離れたレイヤでは、プロパティがより抽象的であり得る。CNNの「タスク性能」は、特定の入力を用いてタスクを実行する際のCNNの結果を、一般に人間によって準備され、「正しい」結果を示すとみなされる、提供されたグラウンドトゥールズと比較することによって測定される。

【0008】

ネットワークポロジが決定されると、ネットワーク重みは、より多くの訓練データが利用可能になるにつれて時間更新され得る。また、CNNの一部を再トレーニングし、ネットワークの他の部分（複数可）における重みを変更しないままにすることも可能である。CNNの全体的な複雑さは高くなる傾向があり、比較的多数の積和演算が実行され、多数の中間テンソルがメモリに書き込まれ、メモリから読み出される。いくつかのアプリケーションでは、CNNが完全に「クラウド」に実装され、高価で高価な処理能力が必要となる。他のアプリケーションでは、CNNがカメラまたは携帯電話などのエッジデバイスに実装され、柔軟性は低くなるが、処理負荷はより分散される。新しいアーキテクチャは、ネットワークを部分に分割することを含み、そのうちの1つはエッジデバイスで実行され、別の1つはクラウドで実行される。そのような分散ネットワークアーキテクチャは、「協調インテリジェンス」と呼ばれることがある。協調インテリジェンスは、ネットワークの第1部分からの部分的な結果を、いくつかの異なる第2部分、おそらく、各部分は異なるタスクに対して最適化される、と共に、再使用することなどの利点を提供する。協調インテリジェンスアーキテクチャは、WANなどのネットワークを介した送信のために、テンソルデータの効率的な圧縮の必要性を導入する。テンソルデータは、クラウド内のサーバ上に、または様々なタスクのために後で使用される部分的に処理されたデータとしてデバイス内に圧縮された形態で記憶され得る。

【0009】

VVCは特に、ビデオフォーマットが（たとえば、より高い解像度およびより高いフレームレートを有する）能力を増加させるにつれて、ますます高い圧縮性能に対する継続的な需要に対処し、帯域幅コストが比較的高いWANを介したサービス配信に対する増大する市場需要に対処することが予想される。VVCは、現代のシリコンプロセスにおいて実施可能であり、達成された性能と実施コストとの間の許容可能なトレードオフを提供する。実装コストはたとえば、シリコンエリア、CPUプロセッサロード、メモリ利用率、および帯域幅のうちの1つまたは複数に関して考慮され得る。VVC規格の汎用性の一部は、ビデオデータを圧縮するために利用可能なツールの幅広い選択、ならびにVVCが適している広範囲のアプリケーションにある。

【0010】

ビデオデータは、画像データのフレームのシーケンスを含み、各フレームは、1つまたは複数のカラーチャンネルを含む。一般に、1つの一次カラーチャンネルおよび2つの二次カラーチャンネルが必要とされる。一次カラーチャンネルは一般に「ルマ」チャンネルと呼ばれ、二次カラーチャンネルは一般に「クロマ」チャンネルと呼ばれる。ビデオデータは典型的にはRGB（赤－緑－青）色空間で表示されるが、RGB色空間は3つのそれぞれの構成要素の間に高い相関度を有する。エンコーダまたはデコーダによって見られるビデオデータ表現はしばしば、YCbCrなどの色空間を使用している。YCbCrは、伝達関数に従って「ルマ」にマッピングされた輝度を、Y（一次）チャンネルに集中させ、CbおよびCr（二次）チャンネルに彩度を集中させる。非相関YCbCr信号の使用により、ルマチャネ

10

20

30

40

50

ルの統計は、クロマチャネルの統計と著しく異なる。主要な差異は、量子化の後、クロマチャネルが対応するルマチャネルブロックの係数と比較して、所与のブロックのための比較的少数の有意係数を含むことである。さらに、CbおよびCrチャネルは、「4:2:0クロマフォーマット」として知られる、ルマチャネルと比較してより低いレート、例えば、水平方向に半分および垂直方向に半分で空間的にサンプリング（サブサンプリング）されてもよい。4:2:0クロマフォーマットは、インターネットビデオストリーミング、ブロードキャストテレビジョン、Blu-RayTMディスクへの保存など、「コンシューマ」アプリケーションで一般的に使用される。ルマサンプルのみが存在する場合、結果として生じるモノクロフレームは、「4:0:0クロマフォーマット」を使用すると言われる。

10

【0011】

VVC規格は、「ブロックベース」アーキテクチャを規定し、フレームは最初に、「コーディングツリーユニット」(CTU)として知られる領域の正方形アレイに分割される。一般に、CTUは、 128×128 ルマサンプルのような比較的大きな面積を占める。VVC規格を使用するときの他の可能なCTUサイズは、 32×32 および 64×64 である。ただし、各フレームの右端および下端のCTUは、面積がより小さくなり得、暗黙の分割はCBがフレーム内に留まることを確実にすることを生じる。各CTUに関連付けられているのは、ルマチャネルおよびクロマチャネルの両方のための「コーディングツリー」(「共有ツリー」)、またはルマチャネルおよびクロマチャネルのそれぞれのための別個のツリーである。コーディングツリーは、CTUのエリアの、「コーディングブロック」(CB)とも呼ばれるブロックのセットへの分解を定義する。共有ツリーが使用中であるとき、単一のコーディングツリーはルマチャネルおよびクロマチャネルの両方のためのブロックを指定し、その場合、コロケートされたコーディングブロックのコレクションは「コーディングユニット」(CU)と呼ばれる(すなわち、各カラーチャネルのためのコーディングブロックを有する各CU)。CBは、特定の順序で符号化または復号するために処理される。4:2:0クロマフォーマットの使用の結果として、 128×128 ルマサンプルエリアのためのルマコーディングツリーを有するCTUは、 128×128 ルマサンプルエリアと並置された 64×64 クロマサンプルエリアのための対応するクロマコーディングツリーを有する。単一のコーディングツリーがルマチャネルおよびクロマチャネルのために使用されているとき、所与のエリアのためのコロケートされたブロックのコレクションは概して、「ユニット」、たとえば、上記のCU、ならびに「予測ユニット」(PU)および「変換ユニット」(TU)と呼ばれる。4:2:0クロマフォーマットビデオデータのカラーチャネルにまたがるCUを有する単一のツリーは、クロマブロックを、対応するルマブロックの幅および高さの半分にする。所与のエリアのために別個のコーディングツリーが使用されるとき、上述のCB、ならびに「予測ブロック」(PB)および「変換ブロック」(TB)が使用される。

20

30

【0012】

「ユニット」と「ブロック」との間の上記の区別にもかかわらず、「ブロック」という用語は、すべてのカラーチャネルに動作が適用されるフレームのエリアまたは領域の総称として使用され得る。

40

【0013】

CUごとに、フレームデータの対応する領域のコンテンツ(サンプル値)の予測ユニット(PU)が生成される(「予測ユニット」)。さらに、予測と、エンコーダへの入力で見られる領域の内容と、の間の差分(または「空間領域」残差)の表現が形成される。各カラーチャネルにおける差分は、残差係数のシーケンスとして変換され、コーディングされ得、所与のCUのための1つまたは複数のTUを形成する。適用される変換は、残差値の各ブロックに適用される、離散コサイン変換(DCT)または他の変換であり得る。変換は分離可能に適用される(すなわち、2次元変換は、2つのパスで実行される)。ブロックは最初に、ブロック内のサンプルの各行に1次元変換を適用することによって変換される。次いで、部分結果は、部分結果の各列に1次元変換を適用することによって変換さ

50

れ、残差サンプルを実質的に非相関化する変換係数の最終ブロックを生成する。様々なサイズの変換は、矩形形状ブロックの変換を含むVVC規格によってサポートされ、各サイド寸法は2のべき乗である。変換係数は、ビットストリームへのエントロピー符号化のために量子化される。

【0014】

VVCは、イントラフレーム予測およびインターフレーム予測を特徴とする。イントラフレーム予測は、フレーム内のデータサンプルの現在のブロックの予測を生成するために使用されているフレーム内の以前に処理されたサンプルの使用を伴う。インターフレーム予測は、以前に復号されたフレームから取得されたサンプルのブロックを使用して、フレーム中のサンプルの現在のブロックの予測を生成することを伴う。以前に復号されたフレームから取得されたサンプルのブロックは、しばしばフィルタリングが適用された動きベクトルに従って、現在のブロックの空間位置からオフセットされる。イントラフレーム予測ブロックは、(i) 均一サンプル値(「DCイントラ予測」)、(ii) オフセットおよび水平および垂直勾配を有するプレーン(「平面イントラ予測」)、(iii) 特定の方向に適用された隣接サンプルを有するブロックの母集団(「角度イントラ予測」)、または(iv) 隣接サンプルおよび選択された行列係数を使用した行列乗算の結果であり得る。予測されたブロックと対応する入力サンプルとの間のさらなる不一致は、「残差」をビットストリームに符号化することによって、ある程度補正され得る。残差は一般に、空間領域から周波数領域に変換されて、「一次変換領域において残差係数を形成し、これは、「二次変換領域」において残差係数を生成するために、「二次変換」のアプリケーションによってさらに変換され得る。残差係数は量子化パラメータに従って量子化され、デコーダで生成されたサンプルの再構成の精度の損失をもたらすが、ビットストリームにおけるビットレートの低減を伴う。ピクチャのシーケンスは、イントラ予測を使用するピクチャおよびイントラ予測またはインター予測を使用するピクチャの指定された構造に従って符号化され得、符号化順序において先行するピクチャに対する指定された依存性は、表示順序または配信順序とは異なり得る。「ランダムアクセス」構成は、周期的なイントラピクチャをもたらし、デコーダおよびビットストリームの復号を開始するエントリポイントを形成する。ランダムアクセス構成における他のピクチャは一般に、インター予測を使用して、指定された深度の階層構造に従って、表示または配信順序で現在のピクチャの前後のピクチャからそれらのコンテンツを予測する。現在のピクチャを予測するために、表示順において現在のピクチャの後のピクチャの使用は、所与のピクチャの復号と、所与のピクチャの表示(およびバッファからの除去)との間のある程度のピクチャバッファリングおよび遅延が必要である。

【発明の概要】

【0015】

本発明の目的は、既存の構成の1つまたは複数の欠点を実質的に克服するか、または少なくとも改善することである。

【0016】

本開示の態様は、テンソルをビットストリームに符号化する方法を提供し、前記テンソルはチャンネル次元を有し、前記方法は、前記テンソルの平均特徴を生成することと、前記平均特徴は、前記チャンネル次元にわたる前記テンソルの平均であり、前記生成された平均特徴を量子化して量子化平均特徴マップを生成することと、前記量子化平均特徴マップの再構成を逆量子化して前記平均特徴の第2表現を生成することと、前記平均特徴マップの前記第2表現および前記テンソルの前記チャンネル次元を使用して平均特徴係数を決定することと、少なくとも前記量子化平均特徴マップおよび前記平均特徴係数の量子化バージョンを前記ビットストリームに符号化することを含む。

【0017】

本開示の他の態様は、量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法を提供し、前記方法は、前記ビットストリームから、前記量子化機能を使用して生成される量子化平均特徴を復号することと、前記量子化平均特徴を逆量子化

10

20

30

40

50

して、前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成される逆量子化平均特徴を導出することと、前記ビットストリームから平均特徴係数を復号することと、少なくとも前記逆量子化平均特徴および前記平均特徴係数を使用して前記特徴マップを復号することを含む。

【0018】

本開示の他の態様は、特徴マップをビットストリームに符号化する方法を提供し、前記テンソルはチャンネル次元を有し、前記方法は、基底ベクトルのセットを量子化して量子化基底ベクトルを生成することと、前記量子化基底ベクトルの再構成を逆量子化して前記基底ベクトルの第2表現を生成することと、前記基底ベクトルの前記第2表現および前記テンソルの前記チャンネル次元を使用して基底ベクトル係数を決定することと、少なくとも前記基底ベクトル係数および前記基底ベクトルの量子化バージョンを前記ビットストリームに符号化することを含む。

10

【0019】

本開示の他の態様は、量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法を提供し、前記方法は、前記量子化機能を使用して生成された量子化基底ベクトルのセットを前記ビットストリームから復号することと、前記量子化基底ベクトルを逆量子化して逆量子化基底ベクトルを導出することと、前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成される逆量子化平均特徴、前記ビットストリームから基底ベクトル係数を復号することと、少なくとも前記逆量子化基底ベクトルおよび前記基底ベクトル係数を使用して前記特徴マップを復号することと、を含む。

20

【0020】

本開示の他の態様は、テンソルをビットストリームに符号化するためのエンコーダを提供し、前記テンソルはチャンネル次元を有し、前記エンコーダは、前記テンソルの平均特徴を生成し、前記平均特徴は、前記チャンネル次元にわたる前記テンソルの平均であり、前記生成された平均特徴を量子化して量子化平均特徴マップを生成し、前記量子化平均特徴マップの再構成を逆量子化して前記平均特徴の第2表現を生成し、前記平均特徴マップの前記第2表現および前記テンソルの前記チャンネル次元を使用して平均特徴係数を決定し、少なくとも前記量子化平均特徴マップおよび前記平均特徴係数の量子化バージョンを前記ビットストリームに符号化するように構成されている。

30

【0021】

本開示の他の態様は、テンソルをビットストリームに符号化する方法を実行するためのプログラムを格納する非一時的コンピュータ可読記憶媒体を提供し、前記方法は、基底ベクトルのセットを量子化して量子化基底ベクトルを生成することと、前記量子化基底ベクトルの再構成を逆量子化して前記基底ベクトルの第2表現を生成することと、前記基底ベクトルの前記第2表現および前記テンソルの前記チャンネル次元を使用して基底ベクトル係数を決定することと、少なくとも前記基底ベクトル係数および前記基底ベクトルの量子化バージョンを前記ビットストリームに符号化することを含む。

【0022】

本開示の他の態様は、システムを提供し、メモリと、テンソルをビットストリームに符号化する方法を実装するために、前記メモリに記憶されたコードを実行するように構成され、前記テンソルはチャンネル次元を有する、プロセッサとを有し、前記方法は、前記テンソルの平均特徴を生成することと、前記平均特徴は、前記チャンネル次元にわたる前記テンソルの平均であり、前記生成された平均特徴を量子化して量子化平均特徴マップを生成することと、前記量子化平均特徴マップの再構成を逆量子化して前記平均特徴の第2表現を生成することと、前記平均特徴マップの前記第2表現および前記テンソルの前記チャンネル次元を使用して平均特徴係数を決定することと、少なくとも前記量子化平均特徴マップおよび前記平均特徴係数の量子化バージョンを前記ビットストリームに符号化することを含む。

40

【0023】

50

本開示の他の態様は、量子化機能を使用して符号化されたビットストリームから特徴マップを復号するためのデコーダを提供し、前記デコーダは、前記ビットストリームから、前記量子化機能を使用して生成される量子化平均特徴を復号し、前記量子化平均特徴を逆量子化して、前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成される逆量子化平均特徴を導出し、前記ビットストリームから平均特徴係数を復号し、少なくとも前記逆量子化平均特徴および前記平均特徴係数を使用して前記特徴マップを復号するように構成されている。

【0024】

本開示の他の態様は、量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法を実行するためのプログラムを格納する非一時的コンピュータ可読記憶媒体を提供し、前記方法は、前記量子化機能を使用して生成された量子化基底ベクトルのセットを前記ビットストリームから復号することと、前記量子化基底ベクトルを逆量子化して逆量子化基底ベクトルを導出することと、前記逆量子化平均特徴は前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成され、前記ビットストリームから基底ベクトル係数を復号することと、少なくとも前記逆量子化基底ベクトルおよび前記基底ベクトル係数を使用して前記特徴マップを復号することと、を含む。

【0025】

本開示の他の態様は、システムを提供し、メモリと、量子化機能を使用して符号化されたビットストリームから特徴マップを復号する方法を実装するために前記メモリ上に記憶されたコードを実行するように構成されるプロセッサとを有し、前記方法は、前記ビットストリームから、前記量子化機能を使用して生成される量子化平均特徴を復号することと、前記量子化平均特徴を逆量子化して、前記ビットストリームを符号化するために使用される前記量子化機能の逆を使用して生成される逆量子化平均特徴を導出することと、前記ビットストリームから平均特徴係数を復号することと、少なくとも前記逆量子化平均特徴および前記平均特徴係数を使用して前記特徴マップを復号することとを含む。

【0026】

他の態様も開示される。

【図面の簡単な説明】

【0027】

次に、本発明の少なくとも1つの実施形態を、以下の図面および付録を参照して説明する。

【図1】図1は、分散マシンタスクシステムを示す概略ブロック図である。

【図2A】図2Aは、図1の分散マシンタスクシステムが実施され得る汎用コンピュータシステムの概略ブロック図を形成する。

【図2B】図2Bは、図1の分散マシンタスクシステムが実施され得る汎用コンピュータシステムの概略ブロック図を形成する。

【図3A】図3Aは、CNNのバックボーン部分の機能モジュールを示す概略ブロック図である。

【図3B】図3Bは、図3Aの残差ブロックを示す概略ブロック図である。

【図3C】図3Cは、図3Aの残差ユニットを示す概略ブロック図である。

【図3D】図3Dは、図3AのCBLモジュールを示す概略ブロック図である。

【図4】図4は、CNNの代替バックボーン部分の機能モジュールを示す概略ブロック図である。

【図5】図5は、分散マシンタスクシステムの一部としてのチャンネル間非相関ベースのテンソルエンコーダを示す概略ブロック図である。

【図6】図6は、チャンネル間非相関ベースのテンソルエンコーダの一部として、特徴マップパッカ、バックフレームエンコーダ、およびアンパッカを示す概略ブロック図である。

【図7】図7は、ビデオエンコーダの機能モジュールを示す概略ブロック図である。

【図8】図8は、分散マシンタスクシステムの一部としてのチャンネル間非相関ベースのテンソルデコーダを示す概略ブロック図である。

【図 9】図 9 は、ビデオデコーダの機能モジュールを示す概略ブロック図である。

【図 10 A】図 10 A は、CNN のヘッド部を示す概略ブロック図である。

【図 10 B】図 10 B は、図 10 A のアップスケーラモジュールを示す概略ブロック図である。

【図 10 C】図 10 C は、図 10 A の検出モジュールを示す概略ブロック図である。

【図 11】図 11 は、CNN の代替ヘッド部を示す概略ブロック図である。

【図 12 A】図 12 A は、チャンネル間非相関テンソルデータを保持するためのサブピクチャの配置を示す概略ブロック図である。

【図 12 B】図 12 B は、チャンネル間非相関テンソルデータを保持するためのサブピクチャの配置を示す概略ブロック図である。

【図 13】図 13 は、符号化されたチャンネル間非相関特徴マップおよび関連するメタデータを保持するビットストリームを示す概略ブロック図である。

【図 14】図 14 は、CNN の第 1 部分を実行し、非相関化し、結果として得られる特徴マップを符号化するための方法を示す。

【図 15】図 15 は、ビットストリームを復号し、非相関特徴マップを再構成し、CNN の第 2 部分を実行するための方法を示す。

【図 16 A】図 16 A は、2 つの連続するデータセットおよび関連する固有ベクトルを示す。

【図 16 B】図 16 B は、2 つの連続するデータセットおよび関連する固有ベクトルを示す。

【図 17】図 17 は、CNN の第 1 部分から生成されたテンソルに対する連続する非相関演算から生じる基底ベクトルを調整するための方法を示す。

【図 18 A】図 18 A は、分解を用いて圧縮する前に FPN 内の異なるレイヤからのテンソルを結合するためのテンソル低減モジュールを示す概略ブロック図である。

【図 18 B】図 18 B は、解凍後に結合テンソルから異なる FPN レイヤのテンソルを抽出するためのテンソル分離モジュールを示す概略ブロック図である。

【図 19】図 19 は、CNN のマルチレイヤ第 1 部分からの別のテンソルとの非相関のために CNN のマルチレイヤ第 1 部分からテンソルを再サンプリングする方法を示す。

【図 20】図 20 は、CNN の第 2 部分について、ビットストリームから再構成されたテンソルを元の解像度に再サンプリングするための方法を示す。

【図 21】図 21 は、CNN の第 1 部分からテンソルを符号化するための基底ベクトルを決定するための方法を示す。

【図 22】図 22 は、CNN の第 1 部分からテンソルを符号化するために使用される基底ベクトルを復号する方法を示す。

【図 23】図 23 は、ネットワーク内の分割点における異なるレイヤのテンソルと、テンソルコンバイナから符号化テンソルを再生するための結果として生じるオプションと、の関係を示す概略ブロック図である。

【0028】

付録 A は、ビットストリームにおける基底ベクトルパッキングに関連するメタデータを表すための SEI メッセージフォーマットおよび関連するセマンティクスを示す。

【発明を実施するための形態】

【0029】

添付の図面のいずれか 1 つまたは複数において、同じ参照符号を有するステップおよび／または特徴が参照される場合、それらのステップおよび／または特徴は反対の意図がない限り、本説明の目的のために、同じ機能または動作を有する。

【0030】

分散マシンタスクシステムは、中間圧縮データを生成するネットワークカメラまたはスマートフォンなどのエッジデバイスを含むことができる。分散マシンタスクシステムはまた、タスク結果を生成するために中間圧縮データについて動作する、サーバファームベース（「クラウド」）アプリケーションなどの最終デバイスを含み得る。タスク結果の例に

10

20

30

40

50

は、オブジェクト検出およびインスタンスセグメンテーションタスクのパフォーマンスを示すために使用されるmAP（平均精度）、および複数のビデオフレームにわたってオブジェクトを追跡する精度を測定するために使用されるMOTA（複数オブジェクト追跡精度）が含まれる。他のタスク結果の例は、人間の姿勢推定および行動認識を含む。加えて、エッジデバイス機能は、クラウドにおいて実施されてもよく、中間圧縮データは潜在的に、必要に応じて複数の異なるタスクのために、後の処理のために記憶されてもよい。

【0031】

中間圧縮データの便利な形態は、高性能圧縮規格およびその実装の利用可能性のために、圧縮ビデオビットストリームである。ビデオ圧縮規格は、典型的には平面アレイに配置された、10ビットなど、いくつかの所与のビット深度の整数サンプルに対して動作する。カラービデオは、用途に応じて、例えば色成分Y、Cb、Cr、またはR、G、Bに対応する3つの平面アレイを有する。CNNは、典型的にはテンソルの形成で浮動小数点データに対して動作する。テンソルは一般に、CNNが動作する入力ビデオデータと比較して、比較的小さい空間次元数を有する。テンソルは一般に、カラービデオデータに典型的な3つのチャンネルよりも多くのチャンネルを有する。

【0032】

テンソルは典型的には、以下の寸法：フレーム、チャンネル、高さ、および幅を有する。例えば、寸法[1、256、76、136]のテンソルは、各々がサイズ136×76の256個の特徴マップ（チャンネル）を含むと言われる。ビデオデータの場合、推論は、典型的には複数のフレームを含むテンソルを使用するのではなく、一度に1フレーム（フレーム値1）で実行される。

【0033】

VVCは、ピクチャの複数のサブピクチャへの分割をサポートし、その各々は、独立して符号化され、独立して復号され得る。1つのアプローチでは、各サブピクチャが1つの「スライス」として、または符号化されたCTUの連続するシーケンスとして符号化される。「タイル」メカニズムもまた、ピクチャをいくつかの独立して復号可能な領域に分割するために利用可能である。サブピクチャは、それぞれのサブピクチャとして符号化されたCTUの様々な矩形セットを用いて、多少柔軟な方法で指定され得る。サブピクチャ次元の柔軟な定義は、1つのピクチャ内の異なるエリアを必要とするタイプのデータを効率的に保持することを可能にし、大きな「未使用」エリア、すなわち、テンソルデータの再構成のために使用されないフレームのエリアを回避する。

【0034】

図1は、分散マシンタスクシステム100の機能モジュールを示す概略ブロック図である。複数のシステムにわたってマシンタスクを分配するという概念は、「協調インテリジェンス（collaborative intelligence）」（CI）と呼ばれることがある。特定のニューラルネットワークを2つの部分に分割するには、ネットワーク内の「分割点」を指定する必要がある。入力レイヤから分割点までのネットワーク内のレイヤは、第1デバイスにおいて実行され、結果として得られる中間テンソル（複数可）が圧縮される。分割点からネットワークの終わりまでのレイヤは、第1デバイスからの解凍テンソルを使用して実行される。分割点において、圧縮される必要がある1つ以上のテンソルが存在し得る。テンソルの次元は同じであってもよいし、異なってもよい。「フィーチャピラミッドネットワーク（feature pyramid network）」（FPN）が使用されている場合、所与のレイヤが前のレイヤの幅および高さの半分であるように、レイヤが幅および高さに関連することが一般的である。FPNアーキテクチャはまた、すべての交互レイヤ上で生じる幅および高さの半減を定義し得る。いくつかのアーキテクチャでは、同じ幅および高さの複数のテンソルが見られる。レイヤ間のチャンネルカウントは、変化してもよく、同じであってもよい。したがって、現代のCNNにおいて使用される様々なネットワークトポロジに適用可能な圧縮方法は、広範囲のシナリオへの適用に有益である。

【0035】

システム１００は、特徴マップを非相関化し、パッキングし、量子化して、符号化データから特徴マップを符号化し、復号するための平面フレームにするための方法を実施するために使用され得る。システム１００は、関連するオーバーヘッドデータの計算負荷が許容可能であり、復号された特徴マップについてのタスク性能がビットストリームのビットレートの変化に耐性があるように使用することができる。システム１００はまた、テンソルの量子化表現がビットを不必要に消費しないように使用され得、ビットは、タスク性能に関して相応の利益を提供しない。

【００３６】

システム１００は、符号化ビデオビットストリーム１２１の形成でCNNバックボーン１１４から符号化テンソルデータ１１５aを生成するためのソースデバイス１１０を含む。システム１００はまた、符号化ビデオビットストリーム１４３の形成でテンソルデータを復号するための宛先デバイス１４０を含む。通信チャンネル１３０は、ソースデバイス１１０から宛先デバイス１４０に符号化ビデオビットストリーム１２１を通信するために使用される。いくつかの構成では、ソースデバイス１１０および宛先デバイス１４０がそれぞれの携帯電話ハンドセット（たとえば、「スマートフォン」）またはネットワークカメラおよびクラウドアプリケーションのいずれかまたは両方を備え得る。通信チャンネル１３０は、イーサネット（登録商標）などの有線接続、またはワイドエリアネットワーク（WAN）を介する接続、またはアドホック接続を介する接続を含む、Wi-Fiまたは5Gなどのワイヤレス接続であり得る。さらに、ソースデバイス１１０および宛先デバイス１４０は、符号化ビデオデータがファイルサーバまたはメモリ中のハードディスクドライブなど、いくつかのコンピュータ可読記憶媒体上でキャプチャされるアプリケーションを備え得る。

【００３７】

図１に示すように、ソースデバイス１１０は、ビデオソース１１２と、CNNバックボーン１１４と、テンソルコンバイナ１６２と、主成分分析（PCA）エンコーダ１６０と、送信機１２２とを含む。ビデオソース１１２は、典型的には画像キャプチャセンサ、非一時的記録媒体上に記憶された以前にキャプチャされたビデオシーケンス、または遠隔画像キャプチャセンサからのビデオフィードなど、キャプチャされたビデオフレームデータ（１１３として示される）のソースを備える。ビデオソース１１２はまた、コンピュータグラフィックスカードの出力であってもよく、例えば、オペレーティングシステムのビデオ出力、およびコンピューティングデバイス（例えば、タブレットコンピュータ）上で実行される様々なアプリケーションを表示する。ビデオソース１１２として画像キャプチャセンサを含み得るソースデバイス１１０の例は、スマートフォン、ビデオカムコーダ、プロフェッショナルビデオカメラ、およびネットワークビデオカメラを含む。ビデオソース１１２は、独立した画像を生成してもよく、または時間的に連続した画像、すなわちビデオを生成してもよい。

【００３８】

CNNバックボーン１１４は、ビデオフレームデータ１１３を受信し、CNNの「バックボーン」に対応するレイヤなど、全体的なCNNの特定のレイヤを実行し、テンソル１１５aを出力する。CNNのバックボーンレイヤは例えば、ビデオフレームデータ１１３によって表される入力画像の異なる空間スケールに対応する複数のテンソルを出力として生成することができる。異なる空間スケールは、「特徴ピラミッドネットワーク(feature pyramid network)」(FPN)アーキテクチャと呼ばれ得る。FPNは例えば、システム１００によって「YOLOv3」ネットワーク（オブジェクト検出に使用される）が実行され、空間分解能およびチャンネルカウントが変化する場合、テンソル１１５aとしてバックボーン１１４から出力される３つのレイヤに対応する３つのテンソルをもたらすことができる。システム１００が「高速RCNN X101-FPN」（オブジェクト検出のために使用される）または「マスクRCNN X101-FPN」（インスタンスセグメンテーションのために使用される）などのネットワークを実行しているとき、テンソル１１５aは、５つのレイヤP2～P5のテンソルを含む。テンソルコンバイナ１６２は

、1つのレイヤについてテンソルを再サンプリングし、得られたテンソルを別のレイヤ、たとえば、F P N中の隣接するレイヤのテンソルと連結し、結合テンソル115を生成することによって、2つのレイヤを結合し得る。P C Aエンコーダ160は、テンソルコンバイナ162から出力されるテンソル115を受信する。レイヤが個別に非相関化されるレイヤである場合に必要となる基底ベクトルよりも少ない基底ベクトルを使用して表される結合レイヤについて、十分なインターレイヤ相関 (inter-layer correlation) が存在する場合に、レイヤの結合は適している。インターレイヤ相関の程度は、ネットワーク自体および提供される入力データの特性である。インターレイヤ相関の程度は、各レイヤのテンソルが別々に個別に非相関化される場合の基底ベクトルの数の合計と比較して、結合テンソルの基底ベクトルの総数の低減を可能にする。たとえば、通常、レイヤごとに25個の基底ベクトルを使用し、2つのレイヤを連結する場合、連結されたレイヤによって必要とされる基底ベクトルの数は、50未満に設定され得る。使用される基底ベクトルの数が、基底ベクトルの説明された分散 (the explained variance) に適応的である構成は、そのようなインターレイヤ相関を利用することによって基底ベクトルカウントの低減を達成することができ、図21および図22を参照して論じられる。インターレイヤ相関が入力データによって影響を受ける限り、テンソルコンバイナ162は、基底ベクトルエリアが最小化されるようにレイヤを結合するかどうかを選択することができる。ネットワークがある程度のインターレイヤ相関を有すると理解される場合、固定された構成が適用されてもよく、例えば、F P Nの最も分解されていないレイヤをダウンサンプリングし、その結果を、非相関ステップ (すなわち、1480) においてF P Nの第2の最も分解されていないレイヤと組み合わせる。レイヤの結合が適切または必要でない場合、テンソルコンバイナ162はいかなる動作も実行せず、テンソル115はテンソル115aに等しい。

10

20

30

40

【0039】

P C Aエンコーダ160は、C N Nバックボーン114によって出力される全体的なC N Nの1つまたは複数の内部レイヤを符号化するように動作する。P C Aエンコーダ160は、まず、主成分分析 (P C A) などの方法を使用して、それぞれのテンソルのチャンネルにわたって、テンソルのD C正規化バージョンを非相関化する。テンソルのD C正規化バージョンは、テンソルの各チャンネルから「平均特徴」を除去することによって生成される。平均特徴は、テンソルのすべてのチャンネルから均一に除去 (減算) されてもよく、または平均特徴は、順変換の適用によってチャンネル依存方式で除去されてもよく、その結果、テンソルの各チャンネルについて対応する「平均係数」が得られる。P C Aから得られる固有ベクトルのサブセットは、符号化されるテンソルのチャンネルを表すために「基底ベクトル」 (または交換可能に、「構成要素」という用語が基底ベクトルを指すために使用され得る) として使用される。テンソルの各チャンネルは、係数を生成するために基底ベクトルを使用して変換される。基底ベクトル、平均特徴マップ、および関連する係数は、量子化され、ビデオフレームにパックされる。ビデオフレームは、ビデオエンコーダを使用して符号化され、ビットストリーム121として出力される。ビットストリーム121は通信チャンネル130を介した送信のために送信機122に供給されるか、またはビットストリーム121は後で使用するために記憶装置132に書き込まれる。

40

【0040】

ソースデバイス110は、C N Nバックボーン114のための特定のネットワークをサポートする。しかしながら、宛先デバイス140は、ヘッドC N N150のためのいくつかのネットワークのうちの1つを使用し得る。ヘッドC N N150のためのいくつかのネットワークのうちの1つを使用する際に、パックされた特徴マップの形態の部分的に処理されたデータは、C N Nバックボーン114の動作を再度実行する必要なしに、様々なタスクを実行する際に後で使用するために記憶され得る。

【0041】

ビットストリーム121は、送信機122によって、符号化ビデオデータ (または「符号化ビデオ情報」) として通信チャンネル130を介して送信される。ビットストリーム1

50

21は、いくつかの実装形態では記憶メモリ132に記憶することができ、記憶装置132は、後に通信チャンネル130を介して（または通信チャンネル130を介した送信の代わりに）送信されるまで、「フラッシュ」メモリまたはハードディスクドライブなどの非一時的記憶デバイスである。例えば、符号化ビデオデータは、ビデオ分析アプリケーションのためのワイドエリアネットワーク（WAN）を介して顧客に要求に応じて供給され得る。

【0042】

宛先デバイス140は、受信機142と、PCAデコーダ170と、テンソルセパレータ172と、CNNヘッド150と、CNNタスク結果バッファ152と、を含む。受信機142は、通信チャンネル130から符号化ビデオデータを受信し、ビデオビットストリーム143をPCAデコーダ170に渡す。PCAデコーダ170は、復号されたテンソル149aを出力し、テンソルセパレータ172に供給される。テンソルセパレータは、テンソルコンバイナ162の逆演算を実行して、抽出されたテンソル149を生成する。抽出されたテンソル149は、CNNヘッド150に渡される。レイヤの結合が適切でないか、または必要とされないアーキテクチャでは、テンソルセパレータ172がいかなる動作も実行せず、テンソル149はテンソル149aに等しい。CNNヘッド150は、CNNバックボーン114で開始されたタスクの後のレイヤを実行して、タスク結果151を生成する。タスク結果151は、タスク結果バッファ152に格納される。タスク結果バッファ152の内容は例えば、グラフィカルユーザインターフェースを介してユーザに提示されてもよく、またはいくつかのアクションがタスク結果に基づいて決定される分析アプリケーションに提供されてもよく、タスク結果は集約されたタスク結果の要約レベルのユーザへの提示を含み得る。ソースデバイス110および宛先デバイス140の各々の機能性は、単一のデバイスにおいて実施されることも可能であり、その例は、携帯電話ハンドセット、タブレットコンピュータ、およびクラウドアプリケーションを含む。

【0043】

上述の例示的なデバイスにもかかわらず、ソースデバイス110および宛先デバイス140の各々は、一般にハードウェア構成要素とソフトウェア構成要素との組合せを通して、汎用コンピューティングシステム内で構成され得る。図2Aは、コンピュータモジュール201、キーボード202とマウスポインタデバイス203とスキャナ226と、ビデオソース112として構成され得るカメラ227と、マイクロフォン280となどの入力デバイスと、プリンタ215、表示デバイス214、およびラウドスピーカ217を含む出力デバイスと、を含む、そのようなコンピュータシステム200を示す。外部変調器-復調器（Modem）トランシーバデバイス216は、コネクション221を介して通信ネットワーク220との間で通信するためにコンピュータモジュール201によって使用され得る。通信チャンネル130を表し得る通信ネットワーク220は、インターネット、セルラー電気通信ネットワーク、またはプライベートWANなどの（WAN）であり得る。コネクション221が電話回線である場合、モデム216は、従来の「ダイヤルアップ」モデムであってもよい。代替として、コネクション221が大容量（たとえば、ケーブルまたは光）コネクションである場合、モデム216はブロードバンドモデムであり得る。ワイヤレスモデムはまた、通信ネットワーク220へのワイヤレス接続のために使用され得る。トランシーバデバイス216は、送信機122および受信機142の機能を提供することができ、通信チャンネル130は、コネクション221において実施され得る。

【0044】

コンピュータモジュール201は、典型的には少なくとも1つのプロセッサユニット205と、メモリユニット206とを含む。例えば、メモリユニット206は、半導体RAM（random access memory）及び半導体ROM（read only memory）を有することができる。コンピュータモジュール201はまた、ビデオディスプレイ214に結合するオーディオビデオインターフェース207、ラウドスピーカ217およびマイクロフォン280と、キーボード202、マウス203、スキャナ226、カメラ227、およびオプションでジョイスティックまたは他のヒューマンイン

ターフェースデバイス（図示せず）に結合するI/Oインターフェース213と、外部モデム216およびプリンタ215のためのインターフェース208とを含む、いくつかの入力／出力（I/O）インターフェースを含む。オーディオビデオインターフェース207からコンピュータモニタ214への信号は一般に、コンピュータグラフィックスカードの出力である。いくつかの実装形態では、モデム216がコンピュータモジュール201内に、たとえばインターフェース208内に組み込まれ得る。コンピュータモジュール201はまた、ローカルエリアネットワーク（LAN）として知られるローカルエリア通信ネットワーク222へのコネクション223を介してコンピュータシステム200の結合を可能にするローカルネットワークインターフェース211を有する。図2Aに示されるように、ローカル通信ネットワーク222はまた、コネクション224を介してワイドネットワーク220に結合することができ、これは、典型的にはいわゆる「ファイアウォール」デバイスまたは同様の機能のデバイスを含む。ローカルネットワークインターフェース211は、EthernetTM回路カード、BluetoothTMワイヤレス構成、またはIEEE802.11ワイヤレス構成を有するが、インターフェース211のために多数の他のタイプのインターフェースが実施され得る。ローカルネットワークインターフェース211はまた、送信機122および受信機142の機能を提供することができ、通信チャネル130はまた、ローカル通信ネットワーク222において実施され得る。

10

【0045】

I/Oインターフェース208および213は、シリアルコネクティビティおよびパラレルコネクティビティのいずれかまたは両方を提供することができ、前者は典型的にはユニバーサルシリアルバス（USB）規格に従って実装され、対応するUSBコネクタ（図示せず）を有する。記憶デバイス209が提供され、典型的にはハードディスクドライブ（HDD）210を含む。また、フロッピーディスク（登録商標）ドライブや磁気テープドライブ（図示せず）などの他の記憶デバイスを用いてもよい。光ディスクドライブ212は、典型的にはデータの不揮発性ソースとして機能するように設けられる。例えば、光ディスク（例えば、CD-ROM、DVD、Blu-ray DiscTM）、USB-RAM、ポータブル、外部ハードドライブ、およびフロッピーディスクなどのポータブルメモリデバイスを、コンピュータシステム200への適切なデータのソースとして使用することができる。典型的には、HDD210、光ドライブ212、ネットワーク220および222のいずれも、ビデオソース112として、またはディスプレイ214を介して再生のために記憶される復号されたビデオデータの宛先として動作するように構成されてもよい。システム100のソースデバイス110および宛先デバイス140は、コンピュータシステム200において実施され得る。

20

30

【0046】

コンピュータモジュール201の構成要素205～213は、典型的には、相互接続されたバス204を介して、当業者に知られているコンピュータシステム200の従来の動作モードをもたらすように通信する。たとえば、プロセッサ205は、コネクション218を使用してシステムバス204に結合される。同様に、メモリ206および光ディスクドライブ212は、コネクション219によってシステムバス204に結合される。説明された構成を実施することができるコンピュータの例は、IBM-PCおよび互換機、SunSPARCstations、AppleMacTM、または同様のコンピュータシステムを含む。

40

【0047】

適切または望ましい場合、PCAエンコーダ160およびPCAデコーダ170、ならびに以下で説明する方法は、コンピュータシステム200を使用して実装され得る。特に、PCAエンコーダ160、PCAデコーダ170、および説明される方法は、コンピュータシステム200内で実行可能な1つまたは複数のソフトウェアアプリケーションプログラム233として実装され得る。特に、PCAエンコーダ160、PCAデコーダ170、および説明される方法のステップは、コンピュータシステム200内で実行されるソ

50

ソフトウェア 233 内の命令 231 (図 2 B 参照) によって実行される。ソフトウェア命令 231 は、それぞれが 1 つまたは複数の特定のタスクを実行するための 1 つまたは複数のコードモジュールとして形成され得る。ソフトウェアはまた、2 つの別個の部分に分割され得、第 1 部分および対応するコードモジュールは説明された方法を実行し、第 2 部分および対応するコードモジュールは、第 1 部分とユーザとの間のユーザインターフェースを管理する。

【0048】

ソフトウェアはたとえば、以下で説明する記憶デバイスを含むコンピュータ可読媒体に記憶され得る。ソフトウェアは、コンピュータ可読媒体からコンピュータシステム 200 にロードされ、次いで、コンピュータシステム 200 によって実行される。コンピュータ可読媒体に記録されたそのようなコンピュータプログラムまたはソフトウェアを有するコンピュータ可読媒体は、コンピュータプログラム製品である。コンピュータシステム 200 におけるコンピュータプログラム製品の使用は、好ましくはソースデバイス 110 および宛先デバイス 140 を実装するための有利な装置と、説明された方法と、をもたらし。

【0049】

ソフトウェア 233 は、典型的には、HDD 210 またはメモリ 206 に格納される。ソフトウェアは、コンピュータ可読媒体からコンピュータシステム 200 にロードされ、コンピュータシステム 200 によって実行される。したがって、たとえば、ソフトウェア 233 は光ディスクドライブ 212 によって読み取られる光学的に読み取り可能なディスク記憶媒体 (たとえば、CD-ROM) 225 に記憶され得る。

【0050】

いくつかの例では、アプリケーションプログラム 233 が 1 つまたは複数の CD-ROM 225 上で符号化されてユーザに供給され、対応するドライブ 212 を介して読み出されてもよく、あるいはネットワーク 220 または 222 からユーザによって読み出されてもよい。さらに、ソフトウェアは、他のコンピュータ可読媒体からコンピュータシステム 200 にロードすることもできる。コンピュータ可読記憶媒体は実行および／または処理のために、記録された命令および／またはデータをコンピュータシステム 200 に提供する任意の非一時的有形記憶媒体を指す。このような記憶媒体の例は、フロッピーディスク、磁気テープ、CD-ROM、DVD、Blu-ray DiscTM、ハードディスクドライブ、ROM 又は集積回路、USB メモリ、光磁気ディスク、又は PCMCIA カード等のコンピュータ可読カード、などを含み、このようなデバイスはコンピュータモジュール 201 の内部又は外付けである。コンピュータモジュール 201 へのソフトウェア、アプリケーションプログラム、命令、および／またはビデオデータもしくは符号化ビデオデータの提供に参加することもできる一時的または非有形のコンピュータ可読伝送媒体の例は、無線または赤外線伝送チャネル、ならびに別のコンピュータまたはネットワーク化されたデバイスへのネットワークコネクション、ならびに電子メール送信およびウェブサイトなどに記録された情報を含むインターネットまたはイントラネットを含む。

【0051】

アプリケーションプログラム 233 の第 2 部分および上述の対応するコードモジュールは、ディスプレイ 214 上にレンダリングされるかまたは表される 1 つまたは複数のグラフィカルユーザインターフェース (GUI) を実施するために実行され得る。典型的にはキーボード 202 およびマウス 203 の操作を通して、コンピュータシステム 200 およびアプリケーションのユーザは、機能的に適応可能な方法でインターフェースを操作して、GUI に関連するアプリケーションに制御コマンドおよび／または入力を提供することができる。ラウドスピーカ 217 を介して出力されるスピーチプロンプトと、マイクロフォン 280 を介して入力されるユーザ音声コマンドとを利用するオーディオインターフェースなど、機能的に適応可能なユーザインターフェースの他の形態も実装され得る。

【0052】

図 2 B は、プロセッサ 205 および「メモリ」234 の詳細な概略ブロック図である。メモリ 234 は、図 2 A のコンピュータモジュール 201 によってアクセス可能な全ての

10

20

30

40

50

メモリモジュール（記憶デバイス２０９及び半導体メモリ２０６を含む）の論理的集合を表す。

【００５３】

コンピュータモジュール２０１が最初に電源投入されると、パワーオンセルフテスト（POST）プログラム２５０が実行される。POSTプログラム２５０は、典型的には図２Ａの半導体メモリ２０６のROM２４９に格納される。ソフトウェアを記憶するROM２４９などのハードウェアデバイスは、ファームウェアと呼ばれることがある。POSTプログラム２５０は適切な機能を保証するためにコンピュータモジュール２０１内のハードウェアを検査し、典型的には、正しい動作のために、プロセッサ２０５、メモリ２３４（２０９、２０６）、および同様に典型的にはROM２４９に記憶された基本入出力システムソフトウェア（BIOS）モジュール２５１をチェックする。POSTプログラム２５０が正常に実行されると、BIOS２５１は、図２Ａのハードディスクドライブ２１０を起動する。ハードディスクドライブ２１０の起動により、ハードディスクドライブ２１０に常駐するブートストラップローダプログラム２５２に、プロセッサ２０５を介して実行させる。これにより、オペレーティングシステム２５３がRAMメモリ２０６にロードされ、オペレーティングシステム２５３が動作を開始する。オペレーティングシステム２５３はプロセッサ２０５によって実行可能なシステムレベルアプリケーションであり、プロセッサ管理、メモリ管理、デバイス管理、ストレージ管理、ソフトウェアアプリケーションインターフェース、および汎用ユーザインターフェースを含む、様々な高レベルの機能を果たす。

10

20

【００５４】

オペレーティングシステム２５３は、メモリ２３４（２０９、２０６）を管理して、コンピュータモジュール２０１上で実行される各プロセスまたはアプリケーションが別のプロセスに割り振られたメモリと衝突することなく実行するのに十分なメモリを有することを保証する。さらに、図２Ａのコンピュータシステム２００において利用可能な異なるタイプのメモリは、各プロセスが効果的に実行され得るように、適切に使用される必要がある。したがって、集約メモリ２３４は、メモリの特定のセグメントがどのように割り振られるかを示すことを意図するものではなく（特に明記しない限り）、むしろ、コンピュータシステム２００によってアクセス可能なメモリの一般的なビュー、およびそのようなメモリがどのように使用されるかを提供することを意図するものである。

30

【００５５】

図２Ｂに示されるように、プロセッサ２０５は、制御部２３９と、算術論理部（ALU）２４０と、キャッシュメモリと呼ばれることもあるローカルまたは内部メモリ２４８とを含む、いくつかの機能モジュールを含む。キャッシュメモリ２４８は、典型的にはレジスタセクション内にいくつかの記憶レジスタ２４４～２４６を含む。１つ以上の内部バス２４１は、これらの機能モジュールを機能的に相互接続する。プロセッサ２０５はまた、典型的には、コネクション２１８を使用して、システムバス２０４を介して外部デバイスと通信するための１つまたは複数のインターフェース２４２を有する。メモリ２３４は、コネクション２１９を使用してバス２０４に結合される。

【００５６】

アプリケーションプログラム２３３は、条件付き分岐およびループ命令を含むことができる一連の命令２３１を含む。また、プログラム２３３は、プログラム２３３の実行に用いられるデータ２３２を含んでもよい。命令２３１およびデータ２３２は、それぞれ、メモリロケーション２２８、２２９、２３０および２３５、２３６、２３７に記憶される。命令２３１およびメモリロケーション２２８～２３０の相対的なサイズに応じて、特定の命令はメモリロケーション２３０に示される命令によって示されるように、単一のメモリロケーションに記憶され得る。代替的に、命令はメモリロケーション２２８および２２９に示される命令セグメントによって示されるように、それぞれが別個のメモリロケーションに記憶されるいくつかの部分にセグメント化され得る。

40

【００５７】

50

一般に、プロセッサ 205 には、その中で実行される命令のセットが与えられる。プロセッサ 205 は、プロセッサ 205 が別の命令セットを実行することによって反応する後続の入力を待つ。各入力は、すべて図 2A に示される、入力デバイス 202、203 のうちの 1 つまたは複数によって生成されたデータ、ネットワーク 220、202 のうちの 1 つにわたって外部ソースから受信されたデータ、記憶デバイス 206、209 のうちの 1 つから取り出されたデータ、または、対応するリーダ 212 に挿入された記憶媒体 225 から取り出されたデータを含む、いくつかのソースのうちの 1 つまたは複数から提供され得る。命令のセットの実行は、場合によってはデータの出力をもたらし得る。実行はまた、データまたは変数をメモリ 234 に記憶することを伴い得る。

【0058】

PCA エンコーダ 160、PCA デコーダ 170、および説明する方法は、メモリ 234 中の対応するメモリロケーション 255、256、257 に記憶される、入力変数 254 を使用し得る。PCA エンコーダ 160、PCA デコーダ 170、および説明される方法は、メモリ 234 中の対応するメモリロケーション 262、263、264 に記憶される、出力変数 261 を生成する。中間変数 258 は、メモリロケーション 259、260、266、および 267 に記憶され得る。

【0059】

図 2B のプロセッサ 205 を参照すると、レジスタ 244、245、246、算術論理ユニット (ALU) 240、および制御部 239 は、プログラム 233 を構成する命令セット内の命令ごとに「フェッチ、デコード、および実行」サイクルを実行するために必要なマイクロオペレーションのシーケンスを実行するように協働する。各フェッチ、デコード、および実行サイクルは：

メモリロケーション 228、229、230 から命令 231 をフェッチまたは読み出すフェッチ動作；

制御部 239 が、どの命令がフェッチされたかを判定するデコード動作；および
制御部 239 および／または ALU 240 が命令を実行する実行動作と
を含む。

【0060】

その後、次の命令のためのさらなるフェッチ、デコード、および実行サイクルが実行され得る。同様に、格納サイクルは、制御部 239 が値をメモリロケーション 232 に格納または書き込むことによって実行されてもよい。

【0061】

説明される図 15 ～ 22 の方法における各ステップまたはサブプロセスは、プログラム 233 の 1 つまたは複数のセグメントに関連付けられ、典型的にはプロセッサ 205 内のレジスタセクション 244、245、246、ALU 240、および制御部 239 によって実行され、プログラム 233 の言及されたセグメントのための命令セット内のすべての命令について、フェッチ、デコード、および実行サイクルを実行するように協働する。

【0062】

図 3A は、CNN バックボーン 114 の実装として機能し得る、CNN のバックボーン部分 310 の機能モジュールを示す概略ブロック図 300 である。バックボーン部分 114 は、「DarkNet-53」と呼ばれることもあるが、異なるバックボーンも可能であり、その結果、フレームごとにテンソル 115 のレイヤの数および次元が異なる。

【0063】

図 3A に示すように、ビデオデータ 113 は、リサイザモジュール 304 に渡される。リサイザモジュール 304 は、ビデオデータ 113 の各フレームを CNN バックボーン 310 による処理に適した解像度にリサイズし、リサイズされたフレームデータ 312 を生成する。ビデオデータ 113 の解像度が既に CNN バックボーン 310 に適している場合、リサイザモジュール 304 の動作は必要とされない。リサイズされたフレームデータ 312 は、畳み込みバッチ正規化リーキー修正線形 (CBL) モジュール (a convolutional batch normalisation leaky recti

10

20

30

40

50

f i e d l i n e a r (C B L) m o d u l e) 3 1 4 に渡され、テンソル 3 1 6 を生成する。C B L 3 1 4 は、図 3 D に示されるように、C B L モジュール 3 6 0 を参照して説明されるようなモジュールを含む。

【 0 0 6 4 】

C B L モジュール 3 6 0 は、入力として、リサイズされたフレームデータ 3 1 2 のテンソル 3 6 1 をとる。テンソル 3 6 1 は、畳み込みレイヤ 3 6 2 に渡されてテンソル 3 6 3 を生成する。畳み込みレイヤ 3 6 2 が 1 のストライドを有する場合、テンソル 3 6 3 はテンソル 3 6 1 と同じ空間ディメンジョンを有する。畳み込みレイヤ 3 6 2 が 2 などのより大きいストライドを有する場合、テンソル 3 6 3 はテンソル 3 6 1 と比較してより小さい空間ディメンジョンを有し、例えば、2 のストライドの幅および高さが半分になる。ストライドにかかわらず、テンソル 3 6 3 のチャンネルディメンジョンのサイズは、特定の C B L ブロックのテンソル 3 6 1 のチャンネルディメンジョンと比較して変化し得る。テンソル 3 6 3 は、テンソル 3 6 5 を出力するバッチ正規化モジュール 3 6 4 に渡される。バッチ正規化モジュール 3 6 4 は、入力テンソル 3 6 3 を正規化し、スケーリング係数およびオフセット値を適用して、出力テンソル 3 6 5 を生成する。スケーリング係数およびオフセット値は、トレーニングプロセスから導出される。テンソル 3 6 5 は、リーキー修正線形活性化 (「L e a k y R e L U」) モジュール 3 6 6 に渡され、テンソル 3 6 7 を生成する。モジュール 3 6 6 は、「リーキー」活性化関数を提供し、テンソル内の正の値が通過され、負の値の大きさが、例えば、それらの以前の値の 0 . 1 X に大幅に低減される。

【 0 0 6 5 】

テンソル 3 1 6 は、C B L ブロック 3 1 4 から、内部的に 1 個の残差ユニット、2 個の残差ユニット、および 8 個の残差ユニットの連結を含む 1 + 2 + 8 モジュールなどの残差ブロックモジュール 3 2 0 に渡される。

【 0 0 6 6 】

残差ブロックは、図 3 B に示される R e s B l o c k 3 4 0 を参照して説明される。R e s B l o c k 3 4 0 は、テンソル 3 4 1 を受信する。テンソル 3 4 1 は、ゼロパディングモジュール 3 4 2 によってゼロパディングされ、テンソル 3 4 3 を生成する。テンソル 3 4 3 は、C B L モジュール 3 4 4 に渡されてテンソル 3 4 5 を生成する。テンソル 3 4 5 は、残差ユニット 3 4 6 に渡される。残差ユニット 3 4 6 は、一連の連結された残差ユニットを含む。残差ユニット 3 4 6 の最後の残差ユニットは、テンソル 3 4 7 を出力する。

【 0 0 6 7 】

図 3 C に示されるように、残差ユニットは、R e s U n i t 3 5 0 を参照して説明される。R e s U n i t 3 5 0 は、テンソル 3 5 1 を入力とする。テンソル 3 5 1 は、C B L モジュール 3 5 2 に渡されてテンソル 3 5 3 を生成する。テンソル 3 5 3 は、第 2 C B L ユニット 3 5 4 に渡され、テンソル 3 5 5 を生成する。加算モジュール 3 5 6 は、テンソル 3 5 5 をテンソル 3 5 1 と合計してテンソル 3 5 7 を生成する。加算モジュール 3 5 6 は、入力テンソル 3 5 1 が出力テンソル 3 5 7 に実質的に影響を与えるので、「ショートカット」と呼ばれることもある。トレーニングされていないネットワークの場合、R e s U n i t 3 5 0 は、パススルーテンソルに作用する。トレーニングが実行されると、C B L モジュール 3 5 2 および 3 5 4 は、トレーニングデータおよびグラウンドトゥルースデータに従ってテンソル 3 5 7 をテンソル 3 5 1 から離すように動作する。

【 0 0 6 8 】

図 3 A に戻って、R e s 1 1 モジュール 3 2 0 は、テンソル 3 2 2 を出力する。テンソル 3 2 2 は、バックボーンモジュール 3 1 0 からレイヤの 1 つとして出力され、R e s 8 モジュール 3 2 4 にも提供される。R e s 8 モジュール 3 2 4 は、8 個の残差ユニット (すなわち、3 5 0) を含む残差ブロック (すなわち、3 4 0) である。R e s 8 モジュール 3 2 4 は、テンソル 3 2 6 を生成する。テンソル 3 2 6 は、R e s 4 モジュール 3 2 8 に渡され、レイヤの 1 つとしてバックボーンモジュール 3 1 0 から出力される。R e s 4 モジュールは 4 つの残差ユニット (すなわち、3 5 0) を含む残差ブロック (すなわち、

340)である。Res4モジュール328は、テンソル329を生成する。テンソル329は、バックボーンモジュール310からレイヤの1つとして出力される。まとめて、レイヤテンソル322、326、および329がテンソル115として出力される。バックボーンCNN310は入力として、解像度1088×608のビデオフレームをとり、3つのレイヤに対応する3つのテンソルを生成することができ、[1、256、76、136]、[1、512、38、68]、[1、1024、19、34]の次元を有する。3つのレイヤに対応する3つのテンソル115の別の例は、[1、512、34、19]、[1、256、68、38]、[1、128、136、76]であってもよく、これらはそれぞれ、CNN310内の第75の特徴マップ、第90の特徴マップ、および第105の特徴マップで分割される。分離点は、CNN310に依存する。

10

【0069】

Res11 320、Res8 324、およびRes4 328の各々は、ResBlock340と同様に動作する。CBL314、CBL344及びCBL354のそれぞれは、CBL360と同様に動作する。

【0070】

図4は、CNNバックボーン114の実装として機能し得る、CNNの代替バックボーン部分400の機能モジュールを示す概略ブロック図である。バックボーン部分400は、特徴ピラミッドネットワーク(「ResNet FPN」)を有する残差ネットワークを実装し、CNNバックボーン114の代替バージョンである。フレームデータ113は入力され、テンソル409、413、417、421、425を介して、ステムネットワーク408、res2モジュール412、res3モジュール416、res4モジュール420、res5モジュール424、および最大プールモジュール428を通過し、最大プールモジュール428は、出力としてP6テンソル429を生成する。

20

【0071】

ステムネットワーク408は、2のストライドを有する7×7畳み込みと、最大プーリング演算とを含む。res2モジュール412、res3モジュール416、res4モジュール420、およびres5モジュール424は、LeakyReLUアクティブ化などの畳み込み演算を実行する。各モジュール412、416、420および424は、また、2のストライド設定を介して、処理されたテンソルの解像度の1つの半分を実行する。テンソル413、417、421、および425の各々は、それぞれ、1×1の横方向畳み込みモジュール446、444、442、および440のうちの1つに渡される。モジュール446、444、442、および440は、それぞれテンソル447、445、443、441を生成する。テンソル441は、3×3出力畳み込みモジュール470に渡され、出力テンソルP5 471を生成する。テンソル441はまた、アップサンプリングされたテンソル451を生成するために、アップサンプラモジュール450に渡される。合計モジュール460はテンソル443および451を合計してテンソル461を生成し、これは、アップサンプラモジュール452および3×3横方向畳み込みモジュール472に渡される。モジュール472は、P4テンソル473を出力する。アップサンプラモジュール452は、アップサンプリングされたテンソル453を生成する。合計モジュール462は、テンソル445および453を合計してテンソル463を生成し、これは、3×3横方向畳み込みモジュール474およびアップサンプラモジュール454に渡される。モジュール474は、P3テンソル475を出力する。アップサンプラモジュール454は、アップサンプリングされたテンソル455を出力する。合計モジュール464は、テンソル447および455を合計してテンソル465を生成し、これは、3×3横方向畳み込みモジュール476に渡される。モジュール476は、P2テンソル477を出力する。アップサンプラモジュール450、452、および454は、低い計算量のために最近傍補間を使用する。テンソル429、471、473、475、および477は、CNNバックボーン400の出力テンソル115を形成する。図4は高速RCNNネットワークアーキテクチャの特定のバックボーン部分(「Pレイヤスプリットポイント」)を示すが、バックボーンとヘッドへの異なる分割が可能である。テンソル413、4

30

40

50

17、421、および425の結果でネットワークを分割することは、「res」分割点と呼ばれる。テンソル409でネットワークを分割することは、「ステム」分割点と呼ばれる。テンソル447、445、443、および441でネットワークを分割することは、「レイヤ」分割点と呼ばれる。

【0072】

図5は、分散マシンタスクシステム100の一部としてのエンコーダ160（または「PCAエンコーダ」）の実装形態に対応するチャンネル間非相関ベースのテンソルエンコーダを示す概略ブロック図500である。PCAエンコーダ160は、入力テンソルの再構成されたバージョンがPCAデコーダ170によって最小限の忠実度の損失で生成されるように、入力テンソル115の次元数を低減するように動作する。フレームデータ113と比較して、テンソル115はより低い空間分解能を有するが、より多くのチャンネルを有する。たとえば、RGBまたはYCbCrフォーマットのフレームデータ113は3つのチャンネルを有し、テンソルは256個のチャンネルを有し得る。結果として、テンソル115のチャンネルをビデオフレームに直接パックすることは、符号化される必要がある比較的大きいサイズのフレームをもたらす。テンソルのチャンネルは内容は様々であるが、ある程度のチャンネル間相関を示す。フレームにパックされるとき、チャンネル間相関はパッキングフォーマットが特徴マップ間の特定の空間オフセットをもたらすので、ビデオエンコーダを使用して活用することが困難であり、ビデオエンコーダは一般に、活用することができない。さらに、利用のためにチャンネル間相関を明らかにするビデオエンコーダであっても、そうするためのインター予測またはイントラブロックコピーの使用は、別のチャンネルからの1つのチャンネルの予測に対応する。チャンネル間の対応は予測モードをテストし、動きベクトルまたはブロックベクトルを探索するときに、チャンネルのペア間の相関がビデオエンコーダにおいて発見され得る程度まで繰り返される。

【0073】

主成分分析（PCA）または同等の方法を使用するチャンネル間非相関のアプローチが、PCAエンコーダ160において使用される。PCAは、「固有空間」を定義する1組のベクトル（「主成分」または「固有ベクトル」）を生成する。PCAの出力は、一次元ベクトルである。説明される構成では、出力1次元ベクトルが、PCAが実行されるテンソルの幅および高さの次元数に対応する2次元形態に変換されると理解される。各固有ベクトルは、固有空間を定義する他の固有ベクトルに対して直交する。直交性は、各固有ベクトルが特徴マップを表現する際に行う寄与に冗長性がないことを意味する。入力テンソルからのそのようなベクトルのセットの導出は、ベクトルのセットが入力テンソルよりも小さい場合、「分解」または次元削減と呼ばれることがある。固有空間は、テンソルのチャンネルを主成分のスカラ倍数で表現できる空間である。主成分の順序付けは、成分における最大分散量が先行する成分によって捕捉されていないテンソルにおける残りの未説明分散の最大量を捕捉するようなものである。したがって、ベクトルのセットの最初のN個の成分を選択することは、ここで、Nはテンソルのチャンネルカウントよりも小さく、オリジナルのテンソルと比較して、デコーダにおいてテンソルを再構成（または「投影」）するときに、精度の低下を最小限に抑えながら、次元が削減された空間でテンソルを表現することを可能にする。1つの空間から別の空間へのテンソルの変換はドット積演算を使用して達成され、入力1次元アレイとして適切に配置され、必要に応じて転置される。

【0074】

主成分分析のための方法は「特異値分解（singular value decomposition）」（SVD）を含み、PCAエンコーダ160の分解機能（以下のモジュール548に関連して説明する）として働くことができる。548においてPCAエンコーダ160において使用され得る分解の他の方法は、QR分解を含み、それは、a Gram-Schmidt process、Householder transformations、またはGivens rotations、またはLU（下位）decomposition、またはCholesky decompositionを使用して実行され得る。各分解方法で使用する値Nは、テンソル115内のチャンネルの数よりも実質的に小さくすることができる。たとえば、テンソル115内の256個のチャネ

ルを用いて、最初の25個の主成分のみを使用して、PCAデコーダ170はCNNヘッド150がロスレス近くで動作するために十分に高い忠実度の再構成されたテンソルを生成することができ、すなわち、CNNバックボーン114とCNNヘッド150との間に非可逆圧縮がないか、または無視できるようにすることができる。

【0075】

主成分または基底ベクトルは非可逆圧縮を使用して符号化され、いくつかの実施形態ではエンコーダが基底ベクトルの非可逆表現を使用する必要性をもたらす。エンコーダにおける基底ベクトルの非可逆表現の使用は、係数を生成するために使用される順変換が復号されたテンソルのチャンネルに基底ベクトルを投影し戻すために使用される逆変換に比べて、より反転可能であることを保証する。分解モジュール548はまた、一般に、ゼロセンタリングされたデータ上での動作を必要とし、またはそれから利益を得て、結果としてより正確な基底ベクトルを生成する。主成分または基底ベクトルは原点に対して相対的であるので、分解されるテンソルも原点の周りに中心を置かなければならず、すなわちゼロ中心でなければならない。ゼロ中心テンソルの使用は直交であり、各連続基底ベクトルがテンソルにおける最大量の未説明分散を正確に示す特性を有する基底ベクトルの生成を可能にする。連続する基底ベクトルがそれぞれ、テンソルにおける最大量の未説明分散を説明するとき、所与の程度までテンソルを説明するために、低減された数の基底ベクトルが必要とされる。言い換えれば、ゼロ中心テンソルの分解は最大の次元数の低減を可能にし、したがって最大の圧縮効率を可能にする。

【0076】

テンソルエンコーダ500はテンソル115を受け取り、テンソル115を抽出平均モジュール510に入力する。分解モジュール548がゼロ中心データに対して動作することを可能にするために、抽出平均モジュール510は、テンソルのチャンネル次元にわたって平均を実行することによって平均特徴マップ511を生成するように動作する。平均特徴マップ511のバージョン、非可逆平均特徴マップ522は、導出された非可逆係数538を使用してドット積モジュール540によって順変換されて、チャンネルごとの平均特徴マップ542を生成する。モジュール240で実施されるドット積は特徴マップの「平坦化された」バージョンに対して実行され、すなわち、2次元アレイから1次元アレイに再形成される。ドット積が特徴マップおよび基底ベクトルに対して実行される場合、特徴マップおよび基底ベクトルの両方が一方が水平方向に、他方が垂直方向に平坦化される。本書の残りの部分ではドット積が(2次元)特徴マップまたは他のデータに対して実行されると言われる場合、1次元表現への、および1次元表現からの再形成は暗黙的であると理解される。チャンネルごとの平均特徴マップ542は、減算器544によってテンソル内の各チャンネルから差し引かれ、DC正規化テンソル546を生成する。

【0077】

サンプル領域における非可逆符号化の潜在的な使用、および、整数サンプル領域への／からの浮動小数点領域からの量子化、から生じる不一致を考慮するために、PCAエンコーダ160は非可逆平均特徴マップ522を生成するために、平均特徴マップ511を量子化し、符号化し、復号し、逆量子化する。平均特徴マップ511は、量子化器モジュール512によって量子化されて、整数平均特徴マップ513を生成する。整数平均特徴マップ513は、サブピクチャエンコーダ514に渡される。サブピクチャエンコーダ514は、特徴マップ513をサブピクチャにパックする。サブピクチャは、ビットストリーム部分515として符号化される。ビットストリーム部分515は、テンソルのチャンネルにわたって平均化された平均特徴マップ511の圧縮表現に対応する。サブピクチャエンコーダ514はまた、再構成された整数平均特徴マップ516を出力し、これは、PCAデコーダ170内で生成されたパックされていないデコードされた平均特徴マップ844に対応し、両方とも、平均特徴マップ511の非可逆表現である。再構成された整数平均特徴マップ516は、逆量子化器モジュール520によって浮動小数点領域に戻されて、非可逆平均特徴マップ522を生成する。非可逆平均特徴マップ522は、整数平均特徴マップ516の浮動小数点表現を提供する。

【0078】

振幅がテンソル115内のチャンネルごとに変化するので、平均特徴マップ522が除去される度合いは、非可逆平均特徴マップ522とテンソル115のそれぞれのチャンネルとのドット積を実行するドット積モジュール524によって生成される「平均係数」を使用して変化させることができる。ドット積モジュールは、テンソル115のチャンネルごとに1つの平均係数を含む平均係数のセット526を生成する。平均係数526は、量子化器モジュール528によってサンプル領域に量子化されて、整数平均係数530を生成する。平均係数530は、サブピクチャにパックされ、サブピクチャエンコーダ532によって符号化されて、ビットストリーム部分533および非可逆サブピクチャ表現534を生成する。ビットストリーム部分533は、平均テンソル値511に関する係数526に対応する。復号された平均係数を表す非可逆サブピクチャ表現534は、逆量子化器モジュール536に供給される。逆量子化器モジュール536は再構成された平均係数538、すなわち、平均係数526の非可逆浮動小数点バージョンを出力する。したがって、ゼロ平均化テンソル546は、PCAデコーダ170において導出されるゼロ平均化テンソル852に対応する。

10

【0079】

損失の原因には、量子化および非可逆符号化が含まれる。平均係数に関する任意の損失に対するチャンネルごとの平均特徴マップ542の生成の感度(sensitivity)に起因して、サブピクチャエンコーダ532は、サブピクチャエンコーダ514に必要とされるよりも高い品質レベル(より小さい量子化ステップサイズまたはより低い量子化パラメータ)で動作する。サブピクチャエンコーダ532は、ビットストリーム部分533においてより多くのビットが費やされるという犠牲を払って、より高い品質のために可逆モードで動作され得る。HEVCなどの規格は、変換および量子化処理をバイパスすることによって、可逆演算を局所的に提供するための手段として、CUのための「変換量子化バイパス」モードを提供する。VV Cなどの規格は代わりに、量子化プロセスが可逆プロセスになるように十分に低いQPを使用し、変換からの損失を回避するために、変換スキップモードが独立して選択され得る。可逆符号化のそのような局所適用は「混合」非可逆／可逆符号化を可能にし、可逆符号化は、パックされた係数を含むサブピクチャの領域のためなど、ピクチャの高感度部分(highly sensitive portions)において使用される。平均またはDC除去の結果は、分解モジュール548における基底ベクトル導出により適したゼロ中心テンソル、すなわちDC正規化テンソル546である。

20

30

【0080】

分解モジュールは、基底ベクトルを生成するためにテンソル546のチャンネルにわたって分解機能を実行するように動作し、基底ベクトルは、成分の数と表現の精度との間の許容可能なトレードオフを有するテンソルを表すことが可能な成分である。分解のための様々な機構が、分解モジュール548において使用され得る。単一値分解(SVD)は、分解モジュール548での使用に適した、一般的で計算的に高速な分解方法の1つである。非可逆符号化がソースデバイス110から宛先デバイス140に伝達される必要がある異なるタイプのデータ(平均特徴、係数、基底ベクトル)のために使用され得るにもかかわらず、データのサンプル表現の使用は浮動小数点領域から整数領域への量子化を必要とし、その範囲は、ビデオ符号化および復号処理のビット深度によって制約される。量子化データのための単一のコヒーレントコンテナを生成するために、画像フレームは、複数のサブピクチャから構成され得、その各々は独立して符号化されているサブピクチャと、PCAエンコーダ160において利用可能にされる再構成バージョン(すなわち、対応するデコーダにおいて生成されるようなサブピクチャのバージョン)である。以下、図13A及び図13Bを参照して、サブピクチャの配置について説明する。逆量子化モジュール520および536を使用する結果として、非可逆符号化による圧縮損失が、システム500によって実施されるテンソル符号化プロセスにおいて考慮される。

40

【0081】

分解モジュール548は、基底ベクトル550のセットを生成する。基底ベクトル55

50

0のセットのサイズは一般に、テンソル115のチャンネルの数の一部である。基底ベクトル550は、テンソル115が忠実度の最小損失で近似的に表現され得る新しい部分空間を定義する。テンソルを分解するときにチャンネルがあるより少ない基底ベクトルを生成することは、「次元削減」と呼ばれる。次元削減は、テンソル115のチャンネルを表現することに最も重要でない寄与をする基底ベクトルを破棄するか、または生成しないことによって、非可逆圧縮を達成するための手段を提供する。例えば、255または25個の基底ベクトルの入力チャンネルカウントの約10%が、分解モジュール548によって生成され得る。基底ベクトルはテンソル115内の特徴マップを表現することができる固有空間を形成し、次元、したがって、ピクチャまたはサブピクチャ内の符号化されたエリアが大幅に低減される。基底ベクトル550は、量子化器552によって量子化されて、整数基底ベクトルテンソル554を生成する。量子化器552は、基底ベクトル550を使用してテンソルのための（整数）係数を生成するように動作する。整数基底ベクトルテンソル554は、サブピクチャにパックされ、サブピクチャエンコーダ556によって符号化される。サブピクチャエンコーダ556は、ビットストリーム部分557を出力し、再構成されたサブピクチャをアンパックした結果として、再構成された整数テンソル558を出力する。ビットストリーム部分557は、モジュール548によるテンソル546の分解によって生成された成分550に対応する。逆量子化器モジュール560は、再構成された整数テンソル558を逆量子化して、再構成された基底ベクトルテンソル562を生成する。

10

20

30

40

50

【0082】

順変換は、ドット積モジュール564によって実行され、ゼロ平均テンソル546の各チャンネルと、再構成基底ベクトルテンソル562内の基底ベクトルの各々とのドット積を生成して、係数のセット566を生成する。ゼロ平均テンソル546内の256個のチャンネルおよびテンソル562内の25個（25個）の成分を用いて、セット566として出力される $256 \times 25 = 6400$ 個の係数が存在する。係数566は、量子化器モジュール568によって浮動小数点領域から整数領域に量子化される。量子化器モジュール568は、係数サンプル570のセットを出力する。係数サンプル570は、サブピクチャにパックされ、サブピクチャエンコーダ572によってビットストリーム部分573に符号化される。ビットストリーム部分573は、成分550に関連する係数566に対応する。ビットストリーム部分515、533、557、および573は4つのサブピクチャに対応し、サブピクチャビットストリームコンバイナ580によって結合されて、符号化ビットストリーム121を生成する。PCAデコーダ170に見られる再構成された係数818の忠実度に対するタスク性能の感度（sensitivity）のために、サブピクチャエンコーダ572において使用される品質レベルは基底ベクトルを符号化するために、すなわち、サブピクチャエンコーダ556によって使用される品質レベルよりも相対的に高い。

【0083】

PCAエンコーダ160の構成では、分解モジュール548が可変数の基底ベクトル550を出力するように動作可能であり、その結果、少なくともテンソル115の最小分散量は最大数の成分Mの限界まで、基底ベクトル550によって表され、例えば、256個のチャンネルを有するテンソルについては25である。「固有値」は、基底ベクトル550内の各ベクトルに関連付けられ、それぞれの基底ベクトルを使用することによってどれだけの分散が適応されるか、または「説明される」かを示す。各連続する基底ベクトルが、テンソル115内の最大量の未説明の分散を説明するので、基底ベクトル550の固有値は基底ベクトル550内のベクトルにわたって進行するとき、大きさが減少している。最小分散閾値、85%、90%、または95%、98%、または他の値などを確立することができる。最大値Nは第1のN番目の固有値の累積和が最小分散閾値未満であり、Nが分解モジュール548によって生成された基底ベクトルの数以下である場合に選択され得る。あるいは、最小値Nが第1のN番目の固有値の累積和が最小分散閾値よりも大きく、Nが分解モジュール548によって生成された基底ベクトルの数以下である場合に選択され得る。N個の基底ベクトルのサブセットが基底ベクトル550として出力される。基底ベ

クトル 5 5 0 を（図 1 2 A に関連して説明したように）サブピクチャ 1 2 1 6 にパッキングし、ビットストリーム部分 5 5 7 に符号化するためのモジュール 5 5 2、5 5 6 はいくつかの構成では最初の N 個の基底ベクトルを符号化し、残りのベクトル（存在する場合）が順変換に使用されないようにし、サブピクチャ 1 2 1 6 にパッキングすることができる。その中でより小さい分散を有するテンソル 1 1 5 の場合、選択された値 N は、2 5 などの固定しきい値よりも低くなり得る。N 個の基底ベクトルを選択した結果、サブピクチャ 1 2 1 6 にパックする基底ベクトルのエリアおよびサブピクチャ 1 2 1 4 にパックする係数の数（図 1 2 A に関連して説明したように）が低減され、ビットストリーム 1 2 1 の符号化サイズが低減され得る。ネットワークの分割点が複数のテンソルを含む場合、たとえば、F P N が使用される場合、値 N は、F P N 中の各レイヤについて独立して導出され得、C N N バックボーン 1 1 4 からの F P N の各レイヤについて生成されたテンソルに含まれる複雑さ（したがって、分散）の差への適応を可能にする。圧縮されている各テンソルについて決定された値 N は、P C A エンコーダ 1 6 0 によってビットストリーム 1 2 1 に符号化され、P C A デコーダ 1 7 0 が復号ピクチャから正しい数の基底ベクトルをアンパックすることを可能にする。P C A デコーダ 1 7 0 は、基底ベクトルテンソルカウントを決定するために、アンパックされた数の基底ベクトル N を使用し、サブピクチャ 1 2 1 6 におけるパッキング位置を確立するために、最悪のケースの基底ベクトルカウント M を使用する。P C A デコーダ 1 7 0 は新しい値 N が復号されるときに N を更新しながら、現在および後続のフレームのために復号された値 N を使用して、ビットストリーム 1 4 3 から復号された基底ベクトルとして最初の N 個の位置における基底ベクトルをアンパックすることができる。

10

20

【0 0 8 4】

損失を低減するために、平均係数 5 3 0 および係数サンプル 5 7 0 などの感度データは、可逆的に符号化され得る。平均係数 5 3 0 および係数サンプル 5 7 0 は、V V C とは異なる符号化を使用して符号化され得る。ビデオサンプルとして符号化するために、5 3 0 および 5 7 0 をそれぞれのサブピクチャにパッキングする代わりに、それらは、数のリストのための代替の一般的な方式を使用して符号化され得る。1 つの方式は「D e e p C A B A C」であり、それによって、低い大きさをクラスタ化された大きさを有する数のシーケンスと、多くのゼロ値の数（たとえば、ガウス分布または同様の分布）とが、V V C または H E V C などの規格における残差サンプルに使用されるものと同様の二値化方式とともに、コンテキスト適応型バイナリ算術符号化を使用して効率的に符号化される。整数領域における係数のための可逆符号化または非可逆符号化の使用にかかわらず、浮動小数点領域から整数領域への変換は（浮動小数点値がビットストリーム 1 2 1 において表されない限り、あまり効率的ではない）他のやり方で不可避であるいくつかの損失を導入する。係数 5 3 0 および 5 7 0 が D e e p C A B A C など、V V C とは別個の方式を使用してビットストリーム 1 2 1 中で表される場合、D e e p C A B A C ビットストリームは、「ユーザデータ S E I メッセージ」または他の専用 S E I メッセージなど、S E I メッセージを使用してフレームに関連付けられ得る。各ピクチャ符号化基底ベクトルについて、関連する S E I メッセージは係数および基底ベクトルからの特徴マップの再構成を可能にするために、ピクチャに付随する。

30

40

【0 0 8 5】

量子化器 5 1 2、5 2 8、5 5 2、および 5 6 8 は概して、すべて浮動小数点値を、ビデオフレームのビット深度、たとえば、1 0 ビット符号化が使用されるときに 0 ~ 1 0 2 3 によって与えられるものに対応する整数サンプル範囲に量子化する。サンプルへの量子化は閉範囲の値への量子化を意味するが、V V C に対する代替の圧縮方式が使用される場合、開（非制限）範囲への量子化が可能である。たとえば、サブピクチャエンコーダ 5 3 2 および 5 7 2 が D e e p C A B A C などのアルゴリズムを使用する場合、整数値を値の範囲に適合させる必要はない。浮動小数点値をサンプル範囲にフィッティングするとき、量子化器 5 1 2、5 2 8、5 5 2、および 5 6 8 はそれぞれ、様々な粒度に設定され得るそれぞれの「量子化範囲」を使用して動作する。各それぞれの量子化器の量子化範囲は、

50

テンソルが量子化されるたびに決定され得る。各決定された量子化範囲はPCAデコーダ170が整数値を逆量子化して浮動小数点領域の正しい範囲に戻すことができるように、ビットストリーム121においてシグナリングされる必要がある。

【0086】

ビットストリーム115は、CNNバックボーン114およびCNNヘッド150によって形成されるニューラルネットワークのステージから取得される。したがって、テンソル546は、CNNバックボーン114およびCNNヘッド150によって形成されるニューラルネットワークのステージからのものでもある。図5は1つのテンソル115のPCAベースの圧縮を示しているが、特徴ピラミッドネットワークが存在する点でCNNが分離される場合、圧縮を必要とする複数のテンソルが存在する。FPNが使用されるとき、PCAエンコーダ160は、FPNの各テンソルに対して動作する。サブピクチャエンコーダ514、532、556、および572はFPNのテンソル間で共有され得、パッキングおよびアンパッキングはFPN中の各テンソルに対応するデータのためにそれぞれのサブピクチャ中のエリアを予約することが可能である。FPNが使用される場合、FPNの各レイヤを処理するとき、量子化器512、528、552、および568、ならびに逆量子化器520、536、560のために別個の量子化範囲が使用され得る。別個の量子化範囲は、図13を参照して説明したSEIメッセージにおいてPCAデコーダ170によって使用するためにビットストリーム121において符号化される。

【0087】

PCAエンコーダ160の別の構成では、量子化器512、528、552、および568によって決定された量子化範囲は、それぞれの量子化器の前の量子化範囲に設定され、現在のPCA符号化動作のために量子化されているデータ（データ511、526、550、および566）の範囲に適応するために、必要に応じて増加される。量子化範囲を設定すると、ピクチャからピクチャへのメモリ効果が生じ、これにより、テンソル圧縮動作のシーケンスにわたって量子化範囲が「拡大」する。テンソルのシーケンスにわたって増大する量子化範囲は、基底ベクトルなどのデータの大部分のサンプル値を変更する特定のテンソルに見られる外れ値に対する感度を低下させる。テンソルのシーケンスにわたって増大する量子化範囲は、連続するテンソル圧縮動作のパックされたピクチャがビデオシーケンスとして見られるときに観察され得る「ちらつき」効果を抑制することができる。パックされたピクチャにおけるちらつきの抑制はエンコーダが圧縮された形態で効率的に検出および表現することが困難である輝度シフトを低減することによって、ビデオエンコーダ（たとえば、エンコーダ514、532、556、および572）を使用する圧縮を容易にする。ビデオエンコーダがランダムアクセス構成におけるイントラピクチャなどの周期的リフレッシュを使用する場合、量子化範囲は、そのようなピクチャの現在のテンソルのみから決定され得る。ランダムアクセスシーケンスの場合、非イントラピクチャは、前のピクチャから基底ベクトルなどのコンテンツを予測するためにインター予測符号化ツールを使用し得、連続するフレーム中の基底ベクトルを表すために使用されるサンプル値の分布の相対的安定性に起因して、より高い圧縮効率を達成する。

【0088】

PCAエンコーダ160およびPCAデコーダ170の別の構成では、FPNが使用されるとき、レイヤ間の基底ベクトルのための係数の量子化範囲（QR）間に関係が確立される。CNNバックボーン114およびCNNヘッド150がFasterRCNNまたはMaskRCNNネットワークを形成する場合、レイヤNの基底ベクトル係数のQRは、レイヤN+1のQRの2倍に設定され、以下同様である。QRを前のレイヤのQRの2倍に設定することは、テンソル圧縮のために2つのレイヤが単一のレイヤに連結される場合、例えば、ダウンサンプリングされたP2およびP3が連結されてレイヤNを形成し、P4がレイヤN+1を形成する場合を含む。基底ベクトル係数について1つのQRがシグナリングされる場合、各レイヤについてのQRは、レイヤインデックスに基づいて導出される。CNNバックボーン114およびCNNヘッド150がYOLOv3ネットワークまたは類似のものを形成するとき、解像度ベースの順序付けは、レイヤN+1がレイヤ

NのQRの2倍を有するように反転される。ビットストリーム121中のシグナリングがFPNが使用されているときに、基底ベクトルのための係数の逆量子化のために、PCAデコーダ170中でどの順序付けが使用されるべきかを示す。

【0089】

図6は、エンコーダアーキテクチャの概略ブロック図600である。エンコーダアーキテクチャ600は、特徴マップパッカ610と、パックスドフレームエンコーダ614と、アンパッカ620とを含む。サブピクチャエンコーダ514、532、556、および572は、アーキテクチャ600のインスタンスとして実装される。パッカ610は所与のチャンネルカウント、幅、および高さ次元を有し、整数値を含む、すなわちすでに量子化されたテンソル608を受信し、たとえば、サブピクチャエンコーダ514は、整数平均特徴マップ513を受信する。パッカ610は、受信したテンソルをサンプルの2D平面アレイにパッキングする。概して、説明される構成では、テンソル608のそれぞれのチャンネルの特徴マップが左から右および上から下の方法で、サブピクチャフレーム612に格納される。サブピクチャフレーム612は、特徴マップサイズとサブピクチャフレーム612の次元との間の不一致に起因するパッキングにおけるギャップの許容を含む、テンソル608のチャンネルを保持するのに十分なサイズである必要がある。図7を参照して、一般にVVCエンコーダとして実装されるパックスドフレームエンコーダ614について説明する。エンコーダ614は、それぞれのサブピクチャに対応する符号化ビットストリーム部分616を生成する。例えば、サブピクチャエンコーダ514は、符号化ビットストリーム部分515を出力する。また、エンコーダ614は、ビットストリーム部分616を復号する際に再生された非可逆バージョンに対応する、再構成(reconstructed)フレーム618を出力する。再構成フレーム618は平均特徴マップ513の再構成を表し、符号化に起因して引き起こされる損失は、モデル化または表現される。損失は、エンコーダ614で使用される特定の符号化方法によって被るものなどの符号化損失を反映する。図7に示す例では、エンコーダ614がVVCエンコーダである。別の実装形態では、エンコーダ614が異なる符号化損失を被るHEVCコードであり得る。JPEG LSまたはDeepCABACなどの無損失コーデックがエンコーダ614のために使用され得、その場合、再構成フレーム618はサブピクチャフレーム612に等しい。エンコーダ614のためにHEVCまたはVVCが使用される場合、それらは、係数のための可逆符号化を実行するように構成され得る。

【0090】

再構成フレーム618は、アンパッカ620に渡され、アンパッカ620は特徴マップを抽出して、テンソル608と同じ次元数を有する再構成テンソル622を生成する。モジュール600の動作の結果として、PCAエンコーダ160は、PCAデコーダ170に見られるバージョンに対応する特徴マップ(または係数)のバージョンを使用することが許可され、したがって、損失源が考慮されない場合よりも高いレベルの忠実度で動作する。サブピクチャエンコーダ514、532、556、および572は、ループフィルタリングが一般に、復号されたピクチャの人間の消費のために最適化されるので、内部的にもサブピクチャ境界にわたってもループフィルタリングを無効にするように構成される。システム100は、ビデオ圧縮を手段として使用して、CNNバックボーン114からCNNヘッド150に伝搬される必要がある中間テンソルデータに対して実行される次元縮小から得られるデータを効率的に表す。

【0091】

図7は、ビデオエンコーダ614の機能モジュールを示す概略ブロック図700である。ビデオエンコーダ614は、全体的なピクチャを定義するサブピクチャのセットの中で1つのサブピクチャを符号化する。すべてのサブピクチャが1つの符号化パスにおいて1つのピクチャとして符号化されることが可能であるが、1つの符号化パスの使用はパイプラインにおける非可逆符号化を考慮するためのPCAエンコーダ160におけるフィードバックが可能ではないことを意味する。一般に、データは、固定サイズのサブブロックへのブロックの分割など、サンプルまたは係数のグループ内のビデオエンコーダ614内の

機能モジュール間を通過するか、またはアレイとして通過する。ビデオエンコーダ 6 1 4 は、図 2 A および 2 B に示されるように、汎用コンピュータシステム 2 0 0 を使用して実装され得、様々な機能モジュールは、コンピュータシステム 2 0 0 内の専用ハードウェアによって、ハードディスクドライブ 2 0 5 上に常駐し、プロセッサ 2 0 5 によってその実行において制御されるソフトウェアアプリケーションプログラム 2 3 3 の 1 つまたは複数のソフトウェアコードモジュールなど、コンピュータシステム 2 0 0 内で実行可能なソフトウェアによって実装され得る。代替的に、ビデオエンコーダ 6 1 4 は、専用ハードウェアとコンピュータシステム 2 0 0 内で実行可能なソフトウェアとの組合せによって実装され得る。ビデオエンコーダ 6 1 4 および説明された方法は代替的に、説明された方法の機能またはサブ機能を実行する 1 つまたは複数の集積回路などの専用ハードウェアで実装され得る。そのような専用ハードウェアは、グラフィックプロセッシングユニット (GPU)、デジタルシグナルプロセッサ (DSP)、特定用途向け標準製品 (ASSP)、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)、または 1 つもしくは複数のマイクロプロセッサおよび関連するメモリを含み得る。特に、ビデオエンコーダ 6 1 4 は、各々がソフトウェアアプリケーションプログラム 2 3 3 の 1 つまたは複数のソフトウェアコードモジュールとして実装され得るモジュール 7 1 0 ~ 7 9 0 を含む。

10

【0092】

図 7 のビデオエンコーダ 6 1 4 は汎用ビデオコーディング (VVC) ビデオ符号化パイプラインの一例であるが、本明細書で説明する処理ステージを実行するために他のビデオコーデックも使用され得る。例えば、HEVC を使用することができる。説明する例は、符号化データのビットストリームを生成する。他のコーデックが使用された場合、いくつかの実装形態は、フレームフォーマットなどの異なるフォーマットにデータをパックし得る。ビデオエンコーダ 6 1 4 は、サブピクチャの一連のフレームなど、サブピクチャフレームデータ 6 1 2 を受信し、各フレームは、1 つまたは複数のカラーチャネルを含む。フレームデータ 6 1 2 は、使用中のプロファイルによってサポートされる任意のクロマフォーマットおよびビット深度、たとえば、8 ~ 1 0 ビットのサンプル精度で、VVC 規格の「Main 10」プロファイルに対して 4 : 0 : 0、4 : 2 : 0 であり得る。ブロックパーティショナ 7 1 0 は最初に、フレームデータ 6 1 2 を、形状が概ね正方形であり、CTU のための特定のサイズが使用されるように構成された CTU に分割する。CTU の最大有効サイズはたとえば、「シーケンスパラメータセット」中に存在する「sps_log2_ctu_size_minus 5」シンタックス要素によって構成される、3 2 × 3 2、6 4 × 6 4、または 1 2 8 × 1 2 8 ルマサンプルであり得る。CTU サイズはまた、さらなる分割を伴わない CTU が 1 つの CU を含むので、最大 CU サイズを提供する。ブロックパーティショナ 7 1 0 はさらに、ルマコーディングツリーおよびクロマコーディングツリーに従って、各 CTU を 1 つまたは複数の CB に分割する。ルマチャネルは、一次 (primary) カラーチャネルと呼ばれることもある。各クロマチャネルは、二次 (secondary) カラーチャネルとも呼ばれ得る。CB は様々なサイズを有し、二乗アスペクト比と非二乗アスペクト比の両方を含み得る。しかしながら、VVC 規格では、CB、CU、PU、および TU は常に 2 のべき乗である辺長を有する。したがって、7 1 2 として表される現在の CB は、CTU のルマコーディングツリーおよびクロマコーディングツリーに従って、CTU の 1 つまたは複数のブロックにわたる反復に従って進行する、ブロックパーティショナ 7 1 0 から出力される。

20

30

40

【0093】

フレームデータ 6 1 4 の最初の分割から得られる CTU は、ラスタ走査順序で走査され得、1 つまたは複数の「スライス」にグループ化され得る。スライスは、「イントラ」(または「I」) スライスであり得る。イントラスライス (イスライス) は、スライス内のすべての CU がイントラ予測されることを示す。一般に、符号化レイヤビデオシーケンス (CLVS) 中の第 1 ピクチャはイスライスのみを含み、「イントラピクチャ」と呼ばれる。CLVS は「ランダムアクセスポイント」(すなわち、復号が開始され得るビデオシ

50

ーケンス中の中間フレーム)を形成する、周期的なイントラピクチャを含み得る。代替として、スライスは、それぞれ、スライス中の単予測および双予測の追加の利用可能性を示す、単予測または双予測(それぞれ、「P」または「B」スライス)であり得る。

【0094】

ビデオエンコーダ614は、ピクチャ構造に従ってピクチャのシーケンスを符号化する。1つのピクチャ構造は「低遅延」であり、その場合、インター予測を使用するピクチャは、シーケンス中で以前に発生したピクチャのみを参照し得る。低遅延は、後続のピクチャによる可能な参照のために記憶されることに加えて、各ピクチャが復号されるとすぐに出力されることを可能にする。別のピクチャ構造は「ランダムアクセス」であり、ピクチャの符号化順序は、表示順序とは異なる。ランダムアクセスはインター予測されたピクチャが、復号されたが、まだ出力されていない他のピクチャを参照することを可能にする。表示順序に関して将来の参照ピクチャが復号ピクチャバッファ内に存在するように、ピクチャバッファリングの度合いが必要とされ、その結果、複数フレームの遅延が生じる。

10

【0095】

4:0:0以外のクロマフォーマットが使用されているとき、Iスライスでは、各CTUのコーディングツリーが64×64レベルより下で、1つはルマ用であり、別の1つはクロマ用である2つの別個のコーディングツリーに発散し得る。別個のツリーの使用は、CTUのルマ64×64エリア内のルマとクロマとの間に異なるブロック構造が存在することを可能にする。たとえば、大きいクロマCBは、多数のより小さいルマCBとコロケートされ得、逆もまた同様である。PまたはBスライスにおいて、CTUの単一のコーディングツリーは、ルマおよびクロマに共通のブロック構造を定義する。単一ツリーの結果として生じるブロックは、イントラ予測またはインター予測され得る。

20

【0096】

ピクチャのスライスへの分割に加えて、ピクチャはまた、「タイル」に分割され得る。タイルは、ピクチャの矩形領域をカバーするCTUのシーケンスである。CTU走査は、各タイル内でラスタ走査方式で行われ、1つのタイルから次のタイルに進む。スライスは、整数個のタイル、または所与のタイル内のCTUの整数個の連続する行のいずれかであり得る。

【0097】

各CTUについて、ビデオエンコーダ614は、2つのステージで動作する。第1ステージ(「サーチ」ステージと呼ばれる)では、ブロックパーティショナ710がコーディングツリーの様々な潜在的な構成をテストする。コーディングツリーの各潜在的構成は、関連する「候補」CBを有する。第1ステージは比較的低い歪みで比較的高い圧縮効率を提供するCBを選択するために、様々な候補CBをテストすることを伴う。テストは一般に、ラグランジュ最適化を伴い、それによって、レート(すなわち、符号化コスト)および歪み(すなわち、入力フレームデータ612に対する誤差)の重み付けされた組合せに基づいて候補CBが評価される。「最良」候補CB(すなわち、評価されたレート/歪みが最も低いCB)が、ビットストリーム616への後続の符号化のために選択される。候補CBの評価に含まれるのは、所与のエリアに対してCBを使用するか、または様々な分割オプションに従ってエリアをさらに分割し、より小さい結果のエリアの各々をさらなるCBで符号化するか、またはエリアをさらに分割するオプションである。結果として、コーディングツリーおよびCB自体の両方が、サーチステージにおいて選択される。

30

40

【0098】

ビデオエンコーダ614はCB、たとえば、CB712ごとに、矢印720によって示される予測ブロック(PB)を生成する。PB720は、関連するCB712のコンテンツの予測である。減算器モジュール722は、PB720とCB712との間に、724として示される差分(または、空間領域における差分を参照する「残差」)を生成する。差分724は、PB720およびCB712における対応するサンプル間のブロックサイズ差分である。差分724は、変換され、量子化され、矢印736によって示される変換ブロック(TB)として表される。PB720および関連するTB736は典型的にはた

50

例えば、評価されたコストまたは歪みに基づいて、多くの可能な候補 C B のうちの 1 つから選択される。

【0099】

候補コーディングブロック (C B) は、関連付けられた P B および結果として生じる残差のためにビデオエンコーダ 614 に利用可能な予測モードのうちの 1 つから生じる C B である。T B 736 は、ビデオエンコーダ 614 中の予測された P B と組み合わせられると、ビットストリーム中の追加のシグナリングを犠牲にして、復号された C B とオリジナルの C B 712 との間の差を低減する。

【0100】

したがって、各候補コーディングブロック (C B)、すなわち変換ブロック (T B) と組み合わせた予測ブロック (P B) は、関連する符号化コスト (または「レート」と、関連する差分 (または「歪み」と) を有する。C B の歪みは、典型的には絶対差の和 (S A D)、二乗差の和 (S S D)、または差に適用されるアダマール変換など、サンプル値の差として推定される。各候補 P B から結果として生じる推定は予測モード 787 を決定するために、差分 724 を使用してモードセクタ 786 によって決定され得る。予測モード 787 は、現在の C B のための特定の予測モード、たとえば、イントラフレーム予測またはインターフレーム予測を使用する決定を示す。各候補予測モードと対応する残差符号化とに関連する符号化コストの推定は、残差のエントロピー符号化よりも著しく低いコストで実行され得る。したがって、いくつかの候補モードが、リアルタイムビデオエンコーダにおいてさえ、レート歪み検知において最適モードを決定するために評価され得る。

【0101】

レート歪みに関して最適モードを決定することは、典型的にはラグランジュ最適化のバリエーションを使用して達成される。

【0102】

ラグランジアンまたは同様の最適化処理は、(ブロックパーティショナ 710 による) C B への C T U の最適なパーティショニングの選択、ならびに複数の可能性からの最良の予測モードの選択の両方に使用され得る。モード選択モジュール 786 における候補モードのラグランジュ最適化プロセスの適用を通して、最も低いコスト測定を有するイントラ予測モードが「最良」モードとして選択される。最低コストモードは選択された二次変換インデックス 788 を含み、これはまた、エントロピーエンコーダ 738 によってビットストリーム 614 において符号化される。

【0103】

ビデオエンコーダ 614 の動作の第 2 ステージ (「符号化」ステージと呼ばれる) では、各 C T U の決定されたコーディングツリーに対する反復がビデオエンコーダ 614 において実行される。別個のツリーを使用する C T U の場合、C T U の 64×64 ルマ領域ごとに、ルマコーディングツリーが最初に符号化され、次いでクロマコーディングツリーが符号化される。ルマコーディングツリー内ではルマ C B のみが符号化され、クロマコーディングツリー内ではクロマ C B のみが符号化される。共有ツリーを使用する C T U の場合、単一のツリーは共有ツリーの共通ブロック構造に従って C U (すなわち、ルマ C B およびクロマ C B) を記述する。

【0104】

エントロピーエンコーダ 738 は、可変長および固定長コードワードを使用するシンタックス要素のビットワイズ符号化と、シンタックス要素のための算術符号化モードとをサポートする。「パラメータセット」などのビットストリームの一部、例えば、シーケンスパラメータセット (S P S) およびピクチャパラメータセット (P P S) は、固定長コードワードと可変長コードワードとの組合せを使用する。連続部分とも呼ばれるスライスは、可変長符号化を使用するスライスヘッダと、それに続く算術符号化を使用するスライスデータを有する。スライスヘッダは、スライスレベルの量子化パラメータオフセットなど、現在のスライスに固有のパラメータを定義する。スライスデータは、スライス中の各 C T U のシンタックス要素を含む。可変長符号化および算術符号化の使用は、ビットストリ

10

20

30

40

50

ームの各部分内での逐次的な構文解析を必要とする。部分は、「ネットワーク抽象化レイヤユニット」または「NALユニット」を形成するための開始コードで区切られ得る。算術符号化は、コンテキスト適応型バイナリ算術符号化プロセスを使用してサポートされる。

【0105】

算術的に符号化されたシンタックス要素は、1つ以上の「ピン」のシーケンスからなる。ピンはビットと同様に、「0」または「1」の値を有する。しかしながら、ピンは、ディスクリットビットとしてビットストリーム614内に符号化されない。ピンは、関連する予測（または「可能性が高い」または「最も可能性が高い」）値と、「コンテキスト」として知られる関連する確率とを有する。符号化されるべき実際のピンが予測値と一致するとき、「最確シンボル」（MPS）が符号化される。最も可能性の高いシンボルを符号化することは、ビットストリーム614内の消費ビットに関して比較的安価であり、1つ未満のディスクリットビットに相当するコストを含む。符号化される実際のピンが可能性のある値と一致しない場合、「最低確率シンボル」（LPS）が符号化される。最も可能性の低いシンボルを符号化することは、消費されるビットに関して比較的高いコストを有する。ピン符号化技法は、「0」対「1」の確率が歪んでいるピンの効率的な符号化を可能にする。2つの可能な値（すなわち、「フラグ」）を有するシンタックス要素については、単一のピンが適切である。多くの可能な値を有するシンタックス要素の場合、ピンのシーケンスが必要とされる。

【0106】

シーケンス中の後のピンの存在は、シーケンス中の前のピンの値に基づいて決定され得る。さらに、各ピンは、2つ以上のコンテキストに関連付けられ得る。特定のコンテキストの選択は、シンタックス要素中の以前のピン、隣接するシンタックス要素のピン値（すなわち、隣接するブロックからのもの）などに依存し得る。文脈符号化されたピンが符号化されるたびに、そのピン（もしあれば）のために選択されたコンテキストは、新しいピン値を反映する方法で更新される。このように、バイナリ算術符号化方式は、適応的であると言われる。

【0107】

また、エントロピーエンコーダ738によってサポートされるのは、「バイパスピン」と呼ばれる、コンテキストを欠くピンである。バイパスピンは、「0」と「1」との間の等確率分布を仮定して符号化される。したがって、各ピンは、ビットストリーム614中の1ビットの符号化コストを有する。コンテキストがないことはメモリを節約し、複雑さを低減し、したがって、バイパスピンは特定のピンの値の分布が歪まない場合に使用される。コンテキストおよび適応を使用するエントロピーエンコーダの一例はCABAC（コンテキスト適応型バイナリ算術エンコーダ）として当技術分野で知られており、このエンコーダの多くの変形例がビデオ符号化において使用されている。

【0108】

QPコントローラ790は、量子化器734および逆量子化器740による使用のための量子化ステップサイズを確立するために使用される量子化パラメータ792を決定する。より大きい量子化ステップサイズは一次変換係数728がより小さい値に量子化されることをもたらし、逆変換係数746の忠実度の低減を犠牲にしてビットストリーム121のビットレートを低減する。

【0109】

エントロピーエンコーダ738は文脈符号化されたピンとバイパス符号化されたピンとの組合せを使用して、量子化パラメータ792を符号化し、現在のCBのために使用されている場合、LFNSTインデックス788を符号化する。量子化パラメータ792は各スライスの最初に符号化され、スライス内の量子化パラメータ792の変化は「デルタQP」シンタックス要素を使用して符号化される。デルタQPシンタックス要素は、「量子化グループ」として知られる各エリアにおいて最大で1回シグナリングされる。量子化パラメータ792は、ルマCBの残差係数に適用される。調整された量子化パラメータが、

10

20

30

40

50

コロケートされたクロマ C B の残差係数に適用される。調整された量子化パラメータは、オフセットのリストから選択されたマッピングテーブルおよび C U レベルオフセットに従ってルマ量子化パラメータ 7 9 2 からマッピングすることを含み得る。二次変換インデックス 7 8 8 は、変換ブロックに関連する残差が二次変換の適用によって一次係数に変換される係数位置にのみ有意な残差係数を含むときにシグナリングされる。

【 0 1 1 0 】

C B に関連する各 T B の残差係数は、残差シンタックスを使用して符号化される。残差シンタックスは、係数の有意性を示すために主に算術的に符号化されたピンを使用し、より低い値の大きさとともに、より高い大きさの残差係数のためにバイパスピンを予約することによって、低い大きさに係数を効率的に符号化するように設計される。したがって、非常に低い大きさの値を含む残差ブロックと、有意な係数の疎な配置とが、効率的に圧縮される。さらに、2つの残差符号化方式が存在する。変換が適用されるときに見られるように、正規の残差符号化方式は、T B の左上隅に主に位置する有意係数を有する T B に対して最適化される。変換スキップ残差符号化方式は、変換が実行されない T B に利用可能であり、T B 全体にわたるそれらの分布にかかわらず、残差係数を効率的に符号化することができる。

【 0 1 1 1 】

マルチプレクサモジュール 7 8 4 は、各候補 C B のテストされた予測モードから選択された、決定された最良のイントラ予測モードに従って、イントラフレーム予測モジュール 7 6 4 から P B 7 2 0 を出力する。候補予測モードは、ビデオエンコーダ 6 1 4 によってサポートされるすべての考えられる予測モードを含む必要はない。イントラ予測は3つのタイプに分けられ、第1に、P B に、近くの再構成されたサンプルの平均を表す単一の値をポピュレートすることを伴う「D C イントラ予測」、第2に、P B に、平面に従ってサンプルをポピュレートすることを伴い、D C オフセットおよび垂直および水平勾配は、近くの再構成された隣接するサンプルから導出される「平面イントラ予測」。近隣の再構成されたサンプルは、典型的にはある程度まで P B の右に延在する現在の P B の上の再構成されたサンプルの行と、ある程度まで P B を越えて下方に延在する、現在の P B の左に再構成されたサンプルの列と、を含み、第3に、特定の方向（または「角度」）に P B にわたってフィルタリングされ伝播された再構成された近隣のサンプルを P B にポピュレートすることを含む、「角度イントラ予測」を含む。V V C では、6 5 個の角度がサポートされ、矩形ブロックは合計 8 7 個の角度を生成するために、正方形ブロックには利用できない追加の角度を利用することができる。

【 0 1 1 2 】

クロマ P B には第4のタイプのイントラ予測が利用可能であり、それにより P B は「クロスコンポーネント線形モデル」(C C L M) モードに従ってコロケートされたルマ再構成サンプルから生成される。3つの異なる C C L M モードが利用可能であり、各モードは、隣接するルマサンプルおよびクロマサンプルから導出された異なるモデルを使用する。導出されたモデルは、コロケートされたルマサンプルからクロマ P B のためのサンプルのブロックを生成するために使用される。ルマブロックは、行列の事前定義されたセットから選択された1つの行列を使用して、参照サンプルの行列乗算を使用してイントラ予測され得る。この行列イントラ予測 (M I P) はビデオデータの大きなセットについてトレーニングされた行列を使用することによって利得を達成し、行列は、角度、平面、または D C イントラ予測モードでは容易に捕捉されない予測ブロックと参照サンプルとの間の関係を表す。

【 0 1 1 3 】

モジュール 7 6 4 はまた、「イントラブロックコピー」(I B C) 方法を使用して現在フレームの近くからブロックをコピーすることによって予測ユニットを生成し得る。参照ブロックの位置は V P D U として知られる 6 4 × 6 4 の領域に分割された1つの C T U に相当する領域に制約され、各行または C T U 内および各スライスまたはタイル内の前の C T U の現在の C T U および V P D U の処理された V P D U をカバーする領域はビットスト

リームの構成されたCTUサイズにかかわらず、1つの 128×128 ルマサンプルに対応する領域制限までである。この領域は「IBC仮想バッファ」として知られており、IBC参照領域を制限し、したがって、必要とされる記憶を制限する。IBCバッファは再構成されたサンプル754でポピュレートされ（すなわち、ループフィルタリングの前に）、したがって、フレームバッファ772とは別個のバッファが必要とされる。CTUサイズが 128×128 であるとき、仮想バッファは現在のCTUに隣接し、その左にあるCTUからのサンプルのみを含む。CTUサイズが 32×32 または 64×64 であるとき、仮想バッファは、現在のCTUの左に4または16個のCTUまでのCTUを含む。CTUサイズにかかわらず、IBC参照ブロックのためのサンプルを取得するための隣接CTUへのアクセスは、ピクチャ、スライス、またはタイルのエッジなどの境界によって制約される。特に、より小さい寸法を有するFPNレイヤの特徴マップの場合、 32×32 または 64×64 などのCTUサイズの使用は、以前の特徴マップのセットをカバーするようにより整列された参照領域をもたらす。特徴マップ配置がSAD、SSE、または他の差分メトリックに基づいて順序付けられる場合、IBC予測のための同様の機能マップへのアクセスは、符号化効率の利点を提供する。

10

20

30

40

50

【0114】

特徴マップデータを符号化するときの予測ブロックの残差は、自然ビデオについて見られる残差とは異なる。自然ビデオは通常、オペレーティングシステムのユーザインターフェースなどに一般的に見られるように、画像センサまたはスクリーンコンテンツによってキャプチャされる。特徴マップ残差は、多くの詳細を含む傾向がある。特徴マップ残差における詳細のレベルは、主に様々な変換の低周波数係数よりも多くのスキップ符号化の変換に適している。DCT-2、MTS（水平方向および垂直方向のDST-7、DCT-8の組合せ）、およびLFNST（様々なトレーニングされた分離不可能な変換）を使用して残差符号化の利点を測定するための本発明者らによる実験は、特徴マップ残差が変換符号化から利点を得るのに十分なローカル類似性を有することを示す。しかしながら、特徴マップ残差係数の分布は、変換ブロックのDC（左上）係数に向かってクラスタ化されない。言い換えれば、特徴マップデータを符号化するとき、変換が利得を示すのに十分な相関が存在する。イントラブロックコピーが特徴マップデータの予測ブロックを生成するために使用されるときにも、十分な相関が存在する。したがって、アダマールコスト推定は、SADまたはSSDコスト推定のみ依存するのではなく、特徴マップデータを符号化するとき、イントラブロックコピーの候補ブロックベクトルから生じる残差を評価するときに使用され得る。SADまたはSSDコスト推定は、符号化スキップを変換することにより適した残差を有するブロックベクトルを選択する傾向があり、変換を使用してコンパクトに符号化される残差を有するブロックベクトルを見逃す可能性がある。VVC規格の多重変換選択（MTS）ツールはDCT-2変換に加えて、DST-7およびDCT-8変換の組合せが残差符号化のために水平および垂直に利用可能であるように、特徴マップデータを符号化するとき使用され得る。

【0115】

イントラ予測されたルマコーディングブロックは、垂直または水平のいずれかで、等しいサイズの予測ブロックのセットに区分され得、各ブロックは16個のルマサンプルの最小エリアを有する。このイントラサブパーティション（ISP）アプローチは、別個の変換ブロックがルマコーディングブロック中の1つのサブパーティションから次のサブパーティションへの予測ブロック生成に寄与することを可能にし、圧縮効率を改善する。

【0116】

以前に再構成された隣接するサンプルが利用できない場合、例えば、フレームのエッジにおいて、サンプルの範囲の半分のデフォルトハーフトーン値が使用される。例えば、10ビットビデオの場合、512の値が使用される。フレームの左上位置に配置されたCBには以前のサンプルが利用可能ではないので、角度および平面イントラ予測モードはDC予測モード（すなわち、大きさとしてハーフトーン値を有するサンプルの平面）と同じ出力を生成する。

【0117】

インターフレーム予測の場合、予測ブロック782は、動き補償モジュール780によってビットストリーム内の符号化順序フレームにおいて現在のフレームに先行する1つまたは2つのフレームからのサンプルを使用して生成され、マルチプレクサモジュール784によってPB720として出力される。さらに、インターフレーム予測の場合、単一のコーディングツリーが典型的には、ルマチャネルとクロマチャネルの両方のために使用される。ビットストリーム中のフレームを符号化する順序は、キャプチャまたは表示されたときのフレームの順序とは異なり得る。1つのフレームが予測のために使用されるとき、ブロックは「単予測」であると言われ、1つの関連する動きベクトルを有する。2つのフレームが予測のために使用されるとき、ブロックは「双予測」であると言われ、2つの関連する動きベクトルを有する。Pスライスの場合、各CUは、イントラ予測または単予測され得る。Bスライスの場合、各CUは、イントラ予測、単予測、または双予測され得る。

10

【0118】

フレームは、典型的には「ピクチャのグループ」構造を使用して符号化され、フレームの時間的階層を可能にする。フレームは複数のスライスに分割され得、スライスの各々はフレームの一部分を符号化する。フレームの時間的階層は、フレームを表示する順序で、フレームが前後のピクチャを参照することを可能にする。画像は、各フレームを復号するための依存性が満たされることを保証するために必要な順序で符号化される。予測ユニットのための参照サンプルブロックを選択およびフィルタリングするために1つまたは2つの動きベクトルを使用する代わりに、予測ユニットは複数のより小さいブロックに分割され、動きフィールドが生成され、したがって、各々のより小さいブロックは別個の動きベクトルを有するアフィンインター予測モードが利用可能である。動きフィールドは、予測ユニットの近くの点の動きベクトルを「制御点」として使用する。アフィン予測は、深く分割されたコーディングツリーを使用する必要性がより少ない、変換とは異なる動きの符号化を可能にする。VV Cに利用可能な双予測モードは、シグナリングされたブロックの中心からのオフセットおよび角度を有する、選択された軸に沿った2つの参照ブロックの幾何学的ブレンドを実行する。この幾何学的分割モード（「GPM」）は、コーディングユニットのために符号化された境界のジオメトリを角度および中心オフセットとして、2つのオブジェクト間の境界に沿ってより大きいコーディングユニットを使用することを可能にする。デカルト（x、y）オフセットを使用する代わりに、動きベクトル差は、サポートされる2の累乗距離のセットを用いて、方向（上／下／左／右）および距離として符号化され得る。動きベクトル予測子は、オフセットが適用されないかのように、隣接ブロック（「マージモード」）から取得される。現在のブロックは、選択された隣接ブロックと同じ動きベクトルを共有する。

20

30

【0119】

サンプルは、動きベクトル778および参照ピクチャインデックスに従って選択される。動きベクトル778および参照ピクチャインデックスは、すべてのカラーチャネルに適用され、したがって、インター予測はPBではなくPuでの動作に関して主に説明される。各CTUの1つまたは複数のインター予測ブロックへの分解は、単一のコーディングツリーを用いて説明される。インター予測方法は、動きパラメータの数およびそれらの精度において変化し得る。動きパラメータは典型的には参照フレームのリストからのどの参照フレームが使用されるべきかを示す参照フレームインデックスと、参照フレームの各々のための空間変換とを含むが、より多くのフレーム、専用のフレーム、またはスケーリングおよび回転などの複雑なアフィンパラメータを含み得る。加えて、参照されたサンプルブロックに基づいて密な動き推定を生成するために、所定の動き精緻化処理が適用され得る。

40

【0120】

PB720を決定して選択し、減算器722でオリジナルのサンプルブロックからPB720を減算すると、724として表される最も低い符号化コストを有する残差が得られ

50

、非可逆圧縮を受ける。非可逆圧縮プロセスは、変換、量子化、およびエントロピー符号化のステップを含む。順方向一次変換モジュール726は、差分724に順方向変換を適用し、差分724を空間領域から周波数領域に変換し、矢印728によって表される一次変換係数を生成する。一次元における最大の一次変換サイズは、シーケンスパラメータセット中の「sps_max_luma_transform_size_64_flag」によって構成される32ポイントDCT-2または64ポイントDCT-2変換のいずれかである。符号化されているCBがブロックサイズ（たとえば、64×64または32×32）として表される最大のサポートされる一次変換サイズよりも大きい場合、一次変換726は、差分724のすべてのサンプルを変換するためにタイル方式で適用される。非正方形CBが使用される場合、タイリングはまた、CBの各寸法において最大の利用可能な変換サイズを使用して実行される。たとえば、32の最大変換サイズが使用されるとき、64×16CBは、タイル方式で構成された2つの32×16一次変換を使用する。CBが最大サポート変換サイズよりもサイズが大きいとき、CBはタイル状にTBで埋められる。例えば、64-pt変換最大サイズを有する128×128CBは、2×2配置で4つの64×64TBで埋められる。32-pt変換最大サイズを有する64×128CBは、2×4配置で8つの32×32TBで埋められる。

10

20

30

40

50

【0121】

変換726の適用は、CBのための複数のTBをもたらす。変換の各アプリケーションが32×32よりも大きい差分724のTB、例えば64×64、に対して動作する場合、TBの左上の32×32エリアの外側のすべての結果として生じる一次変換係数728はゼロに設定される（すなわち、破棄される）。残りの一次変換係数728は、量子化器モジュール734に渡される。一次変換係数728は、CBに関連する量子化パラメータ792に従って量子化されて、一次変換係数732を生成する。量子化パラメータ792に加えて、量子化器モジュール734はまた、TB内のそれらの空間位置に従って残差係数をさらにスケールリングすることによって、TB内の不均一量子化を可能にするために「スケールリングリスト」を適用し得る。量子化パラメータ792は、ルマCB対各クロマCBについて異なり得る。一次変換係数732は、順方向二次変換モジュール730に渡され、非分離可能二次変換（NST）動作を実行するか、または二次変換をバイパスすることによって、矢印736によって表される変換係数を生成する。順方向一次変換726は典型的には分離可能であり、行のセット、次いで、各TBの列のセットを変換する。順方向一次変換モジュール726は、水平方向および垂直方向におけるタイプII離散コサイン変換（DCT-2）、または水平および垂直の変換のバイパス、または幅および高さが16サンプルを超えないルマTBについて水平方向または垂直方向におけるタイプVII離散サイン変換（DST-7）とタイプVIIII離散コサイン変換（DCT-8）との組合せのいずれかを使用する。DST-7とDCT-8との組み合わせの使用は、VVC規格において「多重変換選択セット」（MTS）と呼ばれる。

【0122】

モジュール730の順方向二次変換は概して、非分離可能変換であり、それは、イントラ予測されたCUの残差に対してのみ適用され、それにもかかわらず、バイパスされ得る。順方向二次変換は、16個のサンプル（一次変換係数728の左上の4×4サブブロックとして構成される）または48個のサンプル（一次変換係数728の左上の8×8係数において3つの4×4サブブロックとして構成される）のいずれかに対して動作して、二次変換係数のセットを生成する。二次変換係数のセットは、それらが導出される一次変換係数のセットよりも数が少なくてもよい。互いに隣接し、DC係数を含む係数のセットのみに二次変換を適用することにより、二次変換は、「低周波数非分離可能二次変換」（LFNST）と呼ばれる。そのような二次変換は、トレーニングプロセスを通して、およびそれらの分離不可能な性質およびトレーニングされたオリジンによって取得され得、DCTおよびDSTの変種などの分離可能な変換によって捕捉することができない残差信号における追加の冗長性を活用することができる。さらに、LFNSTが適用されるとき、TB中のすべての残りの係数は、一次変換領域および二次変換領域の両方において、ゼロで

ある。

【0123】

量子化パラメータ792は、所与のTBについて一定であり、したがって、TBについての一次変換領域における残差係数の生成のための均一なスケールリングをもたらす。量子化パラメータ792は、シグナリングされた「デルタ量子化パラメータ」とともに周期的に変化し得る。デルタ量子化パラメータ（デルタQP）は、「量子化グループ」と呼ばれる、所与のエリア内に含まれるCUについて1回シグナリングされる。CUが量子化グループサイズよりも大きい場合、デルタQPは、CUのTBのうちの1つを用いて1回シグナリングされる。すなわち、デルタQPは、CUの第1量子化グループについて1回エントロピーエンコーダ738によってシグナリングされ、CUの任意の後続の量子化グループについてはシグナリングされない。「量子化行列」を適用することによって、不均一なスケールリングも可能であり、それによって、各残差係数に適用されるスケールリング係数は、量子化パラメータ792と、スケールリング行列における対応するエントリとの組合せから導出される。スケールリング行列はTBのサイズよりも小さいサイズを有することができ、TBに適用されるとき、TBサイズよりも小さいサイズのスケールリング行列からの各残差係数のためのスケールリング値を提供するために、最近傍アプローチが使用される。残差係数736は、ビットストリーム614における符号化のためにエントロピーエンコーダ738に供給される。典型的には、TUの少なくとも1つの有意な残差係数を有する各TBの残差係数が走査パターンに従って、値の順序付けられたリストを生成するために走査される。走査パターンは一般に、 4×4 「サブブロック」のシーケンスとしてTBを走査し、 4×4 セットの残差係数の粒度で規則的な走査動作を提供し、サブブロックの配置は、TBのサイズに依存する。各サブブロック内の走査および1つのサブブロックから次のサブブロックへの進行は、典型的には後方斜め走査パターンに従う。さらに、量子化パラメータ792は、デルタQPシンタックス要素を使用してビットストリーム614に符号化され、所与のスライスまたはサブピクチャおよび二次変換インデックス788中の初期値のスライスQPがビットストリーム614に符号化される。

【0124】

上述したように、ビデオエンコーダ614は、ビデオデコーダに見られる復号されたフレーム表現に対応するフレーム表現にアクセスする必要がある。したがって、残差係数736は逆二次変換モジュール744を通過し、二次変換インデックス788に従って動作して、矢印742によって表される中間逆変換係数を生成する。中間逆変換係数742は、量子化パラメータ792に従って逆量子化器モジュール640によって逆量子化され、矢印746によって表される逆変換係数を生成する。逆量子化器モジュール740はまた、量子化器モジュール734において実行される順方向スケールリングに対応する、スケールリングリストを使用して残差係数の逆不均一スケールリングを実行し得る。逆変換係数746は、逆一次変換モジュール748に渡されて、TUの、矢印750によって表される残差サンプルを生成する。逆一次変換モジュール748は、順方向一次変換モジュール726を参照して説明したように、最大の利用可能な変換サイズによって制約され、DC-2変換を水平および垂直に適用する。逆二次変換モジュール744によって実行される逆変換のタイプは、順方向二次変換モジュール730によって実行される順方向変換のタイプに対応する。逆一次変換モジュール748によって実行される逆変換のタイプは、一次変換モジュール726によって実行される一次変換のタイプに対応する。加算モジュール752は、残差サンプル750およびPU720を加算して、CUの再構成されたサンプル（矢印754によって示される）を生成する。

【0125】

再構成されたサンプル754は、参照サンプルキャッシュ756およびループ内フィルタモジュール768に渡される。典型的には、コストのかかるオフチップメモリアクセスを回避するためにASIC上のスタティックRAMを使用して実装される参照サンプルキャッシュ756は、フレーム内の後続のCUのためのイントラフレームPBを生成するための依存性を満たすために必要とされる最小限のサンプル記憶を提供する。最小依存性は

10

20

30

40

50

、典型的にはC T Uの次の行による使用のためのC T Uの行の底部に沿ったサンプルの「ラインバッファ」と、C T Uの高さによってその範囲が設定される列バッファリングとを含む。参照サンプルキャッシュ756は、参照サンプル（矢印758によって表される）を参照サンプルフィルタ760に供給する。サンプルフィルタ760は、平滑化演算を適用して、フィルタリングされた参照サンプルを生成する（矢印762によって示される）。フィルタリングされた参照サンプル762は、イントラフレーム予測モジュール764によって使用されて、矢印766によって表される、サンプルのイントラ予測されたブロックを生成する。各候補イントラ予測モードについて、イントラフレーム予測モジュール764は、サンプルのブロック、すなわち766を生成する。サンプルのブロック766は、DC、平面、または角度イントラ予測などの技法を使用して、モジュール764によって生成される。サンプルのブロック766はまた、ビデオエンコーダ614によって行列のセットから選択された行列と、入力として隣接する参照サンプルを用いる行列乗算アプローチを使用して生成され得、選択された行列は、インデックスを使用してビットストリーム614にシグナリングされ、行列のセットのどの行列がビデオデコーダによって使用されるべきかを識別する。

10

20

30

40

50

【0126】

ループ内フィルタモジュール768は、再構成されたサンプル754にいくつかのフィルタリングステージを適用する。フィルタリングステージは、不連続性に起因するアーチファクトを低減するためにCU境界に整列された平滑化を適用する「デブロッキングフィルタ」（DBF）を含む。ループ内フィルタモジュール768に存在する別のフィルタリングステージは「適応ループフィルタ」（ALF）であり、これは、ウィナーベースの適応フィルタを適用して、歪みをさらに低減する。ループ内フィルタモジュール768においてさらに利用可能なフィルタリングステージは、「サンプル適応オフセット」（SAO）フィルタである。SAOフィルタは最初に、再構成されたサンプルを1つまたは複数のカテゴリに分類し、割り当てられたカテゴリに従って、サンプルレベルでオフセットを適用することによって動作する。

【0127】

矢印770によって表されるフィルタリングされたサンプルは、ループ内フィルタモジュール768から出力される。フィルタリングされたサンプル770は、フレームバッファ772に格納される。フレームバッファ772は典型的にはいくつかの（たとえば、最大16個の）ピクチャを記憶する容量を有し、したがって、メモリ206に記憶される。フレームバッファ772は通常、必要とされる大きなメモリ消費のために、オンチップメモリを使用して記憶されない。したがって、フレームバッファ772へのアクセスは、メモリ帯域幅の点でコストがかかる。フレームバッファ772は、参照フレーム（矢印774によって表される）を動き推定モジュール776および動き補償モジュール780に提供する。参照フレーム774は、対応するサブピクチャエンコーダモジュール600（514、532、556、572）の再構成フレーム618として出力され、アンパッカモジュール620に提供される。図7の例では、再構成されたフレームがモジュール726～748および752～754の動作に起因する、非可逆VVC符号化の動作の結果である。

【0128】

動き推定モジュール776は（778として示される）いくつかの「動きベクトル」を推定し、各々は現在のCBの位置からのデカルト空間オフセットであり、フレームバッファ772中の参照フレームのうちの1つの中のブロックを参照する。参照サンプルのフィルタリングされたブロック（782として表される）が、動きベクトルごとに生成される。フィルタリングされた参照サンプル782は、モードセクタ786による潜在的な選択のために利用可能なさらなる候補モードを形成する。さらに、所与のCUについて、PU720は、1つの参照ブロック（「単予測された」）を使用して形成され得るか、または2つの参照ブロック（「双予測された」）を使用して形成され得る。選択された動きベクトルについて、動き補償モジュール780は、動きベクトルにおけるサブピクセル精度

をサポートするフィルタリング処理に従ってP B 7 2 0を生成する。したがって、（多くの候補動きベクトルに対して動作する）動き推定モジュール7 7 6は、（選択された候補のみに対して動作する）動き補償モジュール7 8 0のものと比較して簡略化されたフィルタリング処理を実行して、低減された計算の複雑さを達成することができる。ビデオエンコーダ6 1 4がC Uのためのインター予測を選択するとき、動きベクトル7 7 8はビットストリーム6 1 6に符号化される。

【0 1 2 9】

図7のビデオエンコーダ6 1 4は、汎用ビデオコーディング（V V C）を参照して説明されるが、他のビデオコーディング規格または実装形態はまた、モジュール7 1 0～7 9 0の処理ステージを採用し得る。フレームデータ6 1 2（およびビットストリーム6 1 6）はまた、メモリ2 0 6、ハードディスクドライブ2 1 0、C D-R O M、ブルーレイディスクTM、または他のコンピュータ可読記憶媒体から読み出される（またはこれに対して書き込まれる）ことができる。さらに、フレームデータ6 1 2（およびビットストリーム6 1 6）は、通信ネットワーク2 2 0または無線周波数受信機に接続されたサーバなどの外部ソースから受信（または外部ソースに送信）され得る。通信ネットワーク2 2 0は制限された帯域幅を提供することができ、フレームデータ6 1 2を圧縮することが困難なときにネットワークを飽和させることを回避するために、ビデオエンコーダ1 2 0におけるレート制御の使用を必要とする。

【0 1 3 0】

ビットストリーム6 1 6は、ビデオエンコーダ6 1 4の1つまたは複数のインスタンスによって生成され、プロセッサ2 0 5の制御下で協調的に動作する、フレームデータ6 1 2の空間セクション（C T Uのコレクション）を表す1つまたは複数のスライスから構築され得る。ビットストリーム6 1 6はまた、1つのピクチャを形成するサブピクチャの集合として出力されるべき1つのサブピクチャに対応する1つのスライスを含み得、各サブピクチャはピクチャ中の他のスライスまたはサブピクチャのいずれかに関して独立して符号化可能であり、独立して復号可能である。ピクチャ内の任意のサブピクチャを独立して符号化および復号する能力は、任意の所与のサブピクチャに含まれるパックされた特徴マップまたは係数に対する非可逆圧縮の影響を、テンソル圧縮の後のステージにおける特徴マップまたは係数の非可逆バージョンを使用することによって、P C Aエンコーダ1 6 0において考慮に入れることを可能にする。

【0 1 3 1】

図8は、分散マシンタスクシステム1 0 0の一部としてのチャンネル間非相関ベーステンソルデコーダ1 7 0の実装を示す概略ブロック図8 0 0である。ビデオビットストリーム1 4 3は、V V Cビデオデコーダを実装し、P C Aエンコーダ1 6 0によって生成されたビットストリームを復号するピクチャデコーダ8 0 4に渡される。ピクチャデコーダ8 0 4は、ビデオビットストリーム1 4 3中に存在するサブピクチャを復号し、各サブピクチャは、テンソル1 4 9を再構成するために必要とされる異なるデータタイプに対応する。各復号されたサブピクチャは、復号されたテンソルのための情報のユニットを提供し、ユニットは、平均テンソル値5 1 1、係数5 2 6、基底ベクトル5 5 0、および係数5 6 6のうちの1つに対応する。平均特徴サブピクチャ8 4 0は、ピクチャデコーダ8 0 4によって出力され、アンパッカ8 4 2に渡される。アンパッカ8 4 2は、テンソル1 4 9の幅および高さに基づいて、整数平均特徴マップ8 4 4を抽出する。整数平均特徴マップ8 4 4は、逆量子化器8 4 6に渡され、ここで、サンプル領域から浮動小数点領域への変換が例えばビットストリーム1 4 3から得られる適切な量子化範囲を使用して実行され、復号された平均特徴マップ8 4 8をもたらす。

【0 1 3 2】

平均係数サブピクチャ8 3 0もピクチャデコーダ8 0 4によって出力され、アンパッカ8 3 2に渡される。アンパッカ8 3 2は、テンソル1 4 9のチャンネルカウントに基づいて整数平均係数8 3 4を抽出する。整数平均係数8 3 4は、逆量子化器8 3 6に渡される。逆量子化器8 3 6は、ビットストリーム1 4 3から得られた量子化範囲を使用して、平均

係数 8 3 4 から、整数領域から浮動ポイント領域に変換し、浮動点平均係数 8 3 8 を出力する。ドット積モジュール 8 5 0 は、平均係数 8 3 8 のうちのそれぞれの係数を有する平均特徴 8 4 8 のドット積を実行することによって、テンソル 1 4 9 の各チャンネルの平均特徴マップ 8 5 2 を生成する。逆量子化器 8 4 6、8 3 6、8 2 4、および 8 1 6 は、ビットストリームを符号化するために実装 5 0 0 において使用される対応する量子化機能（5 1 2、5 2 8、5 5 2、および 5 6 8）の逆である。

【0 1 3 3】

量子化および非可逆圧縮の前に、分解モジュール 5 4 8 によって生成された（たとえば、PCA を実行する）パックされた基底ベクトルを含むサブピクチャ 8 2 0 も、ピクチャデコーダ 8 0 4 によって出力され、アンパッカ 8 2 2 に渡される。アンパッカ 8 2 2 は、整数基底ベクトル 8 2 3 を、サブピクチャ 8 2 0 に重複しないように配置された一連の特徴マップとして抽出する。整数基底ベクトル 8 2 3 は、逆量子化器 8 2 4 に渡される。逆量子化器 8 2 4 は、浮動小数点基底ベクトル 8 2 6 を構築し、ビットストリーム 1 4 3 から得られた量子化範囲を適用する。

【0 1 3 4】

テンソル 1 4 3 の特徴マップごとに基底ベクトルごとに 1 つの係数を有する係数群を含むサブピクチャ 8 1 0 も、ピクチャデコーダ 8 0 4 によって出力され、アンパッカ 8 1 2 に渡される。アンパッカ 8 1 2 は、テンソル 1 4 9 の次元数に基づいて、サブピクチャ 8 1 0 から各係数を抽出する。1 つの係数は特徴マップまたはチャンネルごとの基底ベクトルごとに必要とされ、所与の特徴マップまたはチャンネルを再構成する際の各基底ベクトルの寄与を表し、アンパッカ 8 1 2 から出力される整数係数 8 1 4 を集合的に形成する。逆量子化器 8 1 6 は、ビットストリーム 1 4 3 から得られた量子化範囲に従って、整数係数 8 1 4 を整数領域から浮動小数点領域に変換し、浮動小数点係数 8 1 8 を出力する。ゼロ中心テンソル 8 5 4 は、係数 8 1 8 と基底ベクトル 8 2 6 とに対してドット積を実行するドット積モジュール 8 2 8 によって生成される。加算モジュール 8 5 6 は、ゼロ中心テンソル 8 5 4 を平均特徴マップ 8 4 8 と加算して、PCA デコーダ 1 7 0 からの出力としてテンソル 1 4 9 を生成する。PCA デコーダ 1 7 0 における動作の流れは、圧縮されたテンソルの次元数を元の次元数に戻し、テンソル 1 4 9 が所与のマシントスクの実行のために CNN ヘッド 1 5 0 に渡されることを可能にする。

【0 1 3 5】

図 1 2 A および 1 2 B に関連して以下に説明するように、各復号ピクチャ 8 4 0、8 3 0、8 2 0 および 8 1 0 内の情報は、複数のサンプルの 2 次元アレイに配列される。復号されたピクチャ 8 4 0、8 3 0、8 2 0、および 8 1 0 の各々は、上記の図 8 に関連して説明された例では独立して復号可能である。他の実装形態では、量子化器 5 1 2、5 2 8、5 5 2、および 5 6 8 の出力がどのようにパックされ、ビットストリームに符号化されるかに応じて、復号ピクチャ 8 4 0、8 3 0、8 2 0、および 8 1 0 のうちの少なくとも 1 つは他に対して独立して復号可能である。たとえば、復号されたピクチャ 8 4 0、8 3 0、8 2 0、および 8 1 0 の独立および／または同時復号は、単一のデコーダが復号されたピクチャ 8 4 0、8 3 0、8 2 0、および 8 1 0 を順次デコードすることによって達成可能なものよりも高いフレームレートでの動作を容易にするために実行され得る。例えば、宛先デバイス 1 4 0 がクラウド内のサーバに実装されるとき、復号パフォーマンスは特に、ハードウェアモジュールがより計算上要求の厳しい動作をオフロードするために利用可能でないとき、プロセッサ 2 0 5 内の複数のコアにわたって分散されることから利益を得ることができる。

【0 1 3 6】

様々な基底ベクトル（8 2 0）および関連する係数（8 3 0 および 8 1 0）の形態のテンソル 1 4 9 の縮小次元表現は、ベクトルおよび係数をチャンネルごとにピクチャまたはサブピクチャにパックするとき、縮小されたエリアを可能にする。パックされたデータの削減されたエリアは、テンソル 1 1 5 のパックされた特徴マップまたはチャンネルを含むフレームを直接圧縮しようと試みることと比較して、より効率的な圧縮表現を

10

20

30

40

50

もたらず。

【0137】

図9に、ビデオデコーダとも呼ばれるピクチャデコーダ804の例示的な実装900を示す。図9のビデオデコーダ804は汎用ビデオコーディング（VVC）ビデオ復号パイプラインの一例であるが、本明細書で説明する処理ステージ、たとえば、HEVCなどを実行するために、他のビデオコーデックも使用され得る。図9に示されるように、ビットストリーム910（ビットストリーム143に対応する）は、ビデオデコーダ804に入力される。ビットストリーム143は、メモリ206、ハードディスクドライブ210、CD-ROM、ブルーレイディスクTM、または他の非一時的コンピュータ可読記憶媒体から読み出され、ビットストリーム910として実施態様900に提供され得る。あるいは、ビットストリーム910が通信ネットワーク220または無線周波数受信機に接続されたサーバなどの外部ソースから受信されてもよい。ビットストリーム143は、復号されるべきキャプチャされたフレームデータを表す符号化されたシンタックス要素を含む。サブピクチャが独立して復号される場合、各サブピクチャに対応するビットストリーム143の部分は、実装形態900の別個のインスタンスに供給され得る。各サブピクチャのための実装910の別個のインスタンスは、スループットを改善するためのサブピクチャの並列復号を可能にする。

10

【0138】

ビットストリーム910は、エントロピーデコーダモジュール920に入力される。エントロピーデコーダモジュール920は「ピン」のシーケンスを復号することによってビットストリーム910からシンタックス要素を抽出し、シンタックス要素の値をビデオデコーダ804内の他のモジュールに渡す。エントロピーデコーダモジュール920は、可変長および固定長復号を使用して、SPS、PPS、またはスライスヘッダ、算術復号エンジン、を復号し、スライスデータのシンタックス要素を1つまたは複数のピンのシーケンスとして復号する。各ピンは1つまたは複数の「コンテキスト」を使用することができ、コンテキストは、ピンの「1」値および「0」値を符号化するために使用される確率レベルを記述する。所与のピンについて複数のコンテキストが利用可能である場合、「コンテキストモデリング」または「コンテキスト選択」ステップが実行されて、ピンを復号するために利用可能なコンテキストのうちの1つが選択される。ピンを復号するプロセスは順次フィードバックループを形成し、したがって、各スライスは、所与のエントロピーデコーダ920のインスタンスによってスライス全体において復号され得る。単一の（または少数の）高性能エントロピーデコーダ920インスタンスは、ビットストリーム910からのフレームまたはピクチャのすべてのスライスまたはサブピクチャを復号することができ、複数の低性能エントロピーデコーダ920インスタンスは、ビットストリーム910からのフレームのスライスを同時に復号することができる。

20

30

【0139】

エントロピーデコーダモジュール920は、算術符号化アルゴリズム、例えば「コンテキスト適応型バイナリ算術符号化」（CABAC）を適用して、ビットストリーム910からシンタックス要素を復号する。復号されたシンタックス要素は、ビデオデコーダ804内のパラメータを再構成するために使用される。パラメータは、残差係数（矢印924によって表される）、量子化パラメータ974、二次変換インデックス970、およびイントラ予測モード（矢印958によって表される）などのモード選択情報を含む。モード選択情報はまた、動きベクトル、および各CTUの1つまたは複数のCBへのパーティショニングなどの情報を含む。パラメータはPBを生成するために使用され、典型的には以前に復号されたCBからのサンプルデータと組み合わせられる。

40

【0140】

残差係数924は、逆二次変換モジュール936に渡され、二次変換インデックスに従って、二次変換が適用されるか、または動作が実行されない（バイパス）。逆二次変換モジュール936は、二次変換領域係数から、再構成された変換係数932、すなわち一次変換領域係数を生成する。再構成された変換係数932は、逆量子化器モジュール928

50

に入力される。逆量子化器モジュール 928 は、量子化パラメータ 974 に従って、矢印 940 によって表される、再構成された中間変換係数を作成するために、残差係数 932 に対して、すなわち、一次変換係数領域において、逆量子化（または「スケーリング」）を実行する。逆量子化器モジュール 928 はまた、スケーリング行列を適用して、逆量子化器モジュール 740 の動作に対応する、TB 内の不均一な逆量子化を提供し得る。不均一逆量子化行列を使用することがビットストリーム 143 に示される場合、ビデオデコーダ 804 はビットストリーム 143 から量子化行列をスケーリングファクタのシーケンスとして読み出し、スケーリングファクタを行列に配置する。逆スケーリングは、量子化パラメータと組み合わせて量子化行列を使用して、再構成された中間変換係数 940 を作成する。

10

【0141】

再構成された変換係数 940 は、逆一次変換モジュール 944 に渡される。モジュール 944 は、係数 940 を周波数領域から空間領域に戻すように変換する。逆一次変換モジュール 944 は、順方向一次変換モジュール 726 を参照して説明したように、最大利用可能変換サイズによって制約された、逆 DCT-2 変換を水平および垂直に適用する。モジュール 944 の動作の結果は、矢印 948 によって表される残差サンプルのブロックである。残差サンプル 948 のブロックのサイズは、対応する CB に等しい。残差サンプル 948 は、加算モジュール 950 に供給される。

【0142】

加算モジュール 950 において、残差サンプル 948 は、復号された PB（952 として表される）に加算されて、矢印 956 によって表される、再構成されたサンプルのブロックを生成する。再構成されたサンプル 956 は、再構成されたサンプルキャッシュ 960 およびループ内フィルタリングモジュール 988 に供給される。ループ内フィルタリングモジュール 988 は、992 として表される、フレームサンプルの再構成されたブロックを生成する。フレームサンプル 992 は、フレームバッファ 996 に書き込まれる。フレームバッファ 996 は、図 1 のテンソル 149 に対応する画像またはビデオフレーム 914 を出力する。

20

【0143】

再構成されたサンプルキャッシュ 960 は、ビデオエンコーダ 614 の再構成されたサンプルキャッシュ 756 と同様に動作する。再構成されたサンプルキャッシュ 960 は、メモリ 206 なしで後続の CB をイントラ予測するのに必要な再構成されたサンプルのための記憶を提供する（例えば、典型的にはオンチップメモリであるデータ 232 を代わりに使用することによって）。矢印 964 によって表される参照サンプルは、再構成されたサンプルキャッシュ 960 から取得され、参照サンプルフィルタ 968 に供給されて、矢印 972 によって示されるフィルタリングされた参照サンプルを生成する。フィルタリングされた参照サンプル 972 は、イントラフレーム予測モジュール 976 に供給される。モジュール 976 は、ビットストリーム 910 においてシグナリングされ、エントロピーデコーダ 920 によって復号されたイントラ予測モードパラメータ 958 に従って、矢印 980 によって表されるイントラ予測サンプルのブロックを生成する。イントラ予測モジュール 976 は、IBC および MIP を含むエンコーダサイドのモジュール 764 のモードをサポートする。サンプル 980 のブロックは、DC、平面、または角度イントラ予測などのモードを使用して生成される。

30

40

【0144】

CB の予測モードがビットストリーム 143 においてイントラ予測を使用するように示されるとき、イントラ予測サンプル 980 は、マルチプレクサモジュール 984 を介して復号 PB 952 を形成する。イントラ予測はサンプルの予測ブロック（PB）を生成し、これは、同じ色成分における「隣接サンプル」を使用して導出される、1つの色成分におけるブロックである。隣接サンプルは現在のブロックに隣接するサンプルであり、ブロック復号順序において先行しているため、すでに再構成されている。ルマブロックとクロマブロックとがコロケートされる場合、ルマブロックとクロマブロックとは異なるイントラ

50

予測モードを使用し得る。しかしながら、2つのクロマCBは、同じイントラ予測モードを共有する。

【0145】

CBの予測モードがビットストリーム910内のインター予測であると示されるとき、動き補償モジュール934は、938として表されるインター予測サンプルのブロックを生成する。インター予測サンプル938のブロックは、エントロピーデコーダ920によってビットストリーム143から復号された動きベクトルと、フレームバッファ996からサンプルのブロック998を選択およびフィルタリングするための参照フレームインデックスとを使用して生成される。サンプルのブロック998は、フレームバッファ996に記憶された以前に復号されたフレームから取得される。双予測の場合、サンプルの2つのブロックが生成され、一緒にブレンドされて、復号されたPB952のためのサンプルが生成される。フレームバッファ996は、ループ内フィルタリングモジュール988からのフィルタリングされたブロックデータ992でポピュレートされる。ビデオエンコーダ614のループ内フィルタリングモジュール768と同様に、ループ内フィルタリングモジュール988は、DBF、ALF、およびSAOフィルタリング動作のいずれかを適用する。一般に、動きベクトルは、ルマチャネルとクロマチャネルの両方に適用されるが、ルマチャネルとクロマチャネルにおけるサブサンプル補間のためのフィルタリング処理は異なる。

10

【0146】

図7および図9には示されていないが、各クロマチャネル内のサンプル値の範囲のより均一な使用が達成されるようにサンプル値をシフトするために、符号化前にビデオを前処理し、復号後にビデオを後処理するためのモジュールである。マルチセグメント線形モデルは、ビデオエンコーダ614において導出され、サンプルシフティングを取り消すためにビデオデコーダ804によって使用するためにビットストリームにおいてシグナリングされる。この線形モデルクロマスケーリング(LMCS)ツールは、量子化のアプリケーションからより高い品質損失をもたらし得るサンプル空間の利用において、ある程度の不均一性、特に限られた範囲の利用を有する特定の色空間およびコンテンツに対して圧縮利点を提供する。

20

【0147】

図10Aは、「YOLOv3」ネットワークの「DarkNet-53」バックボーン部分を除く部分に対応する、オブジェクト検出のためのCNNのヘッド部分150の例示的な実装1000を示す概略ブロック図である。実装1000は、例えば図3AのようにCNNバックボーンが実装される場合に使用することができる。宛先デバイス140において実行されるべきタスクに応じて、CNNヘッド150の代わりに異なるネットワークを使用することができる。入力テンソル149は、各レイヤのテンソル(すなわち、テンソル1010、1020、および1034)に分離される。テンソル1010は、CBLモジュール1012に渡されてテンソル1014を生成する。テンソル1014は、検出モジュール1016およびアップスケーラモジュール1022に渡される。検出モジュールは、検出テンソルの形成でバウンディングボックス1018を出力する。バウンディングボックス1018は、非最大抑制(NMS)モジュール1048に渡される。

30

40

【0148】

オリジナルのビデオデータ113内の座標を示すバウンディングボックスを生成するために、ネットワーク114のバックボーン部分のサイズ変更の前に、オリジナルのビデオ幅および高さによるスケーリングが、アップスケーラモジュール1022において実行される。アップスケーラモジュール1022は、テンソル1014およびテンソル1020を受信し、CBLモジュール1026に渡されるアップスケールされたテンソル1024を生成する。CBLモジュール1026は、出力としてテンソル1028を生成する。テンソル1028は、検出モジュール1030およびアップスケーラモジュール1036に渡される。検出モジュール1030は、NMSモジュール1048に供給される検出テンソル1032を生成する。アップスケーラモジュール1036は、モジュール1020の

50

もう1つの例である。アップスケーラモジュール1036は、テンソル1028およびテンソル1034を受信し、アップスケールされたテンソル1038を出力する。アップスケールされたテンソル1038は、CBLモジュール1040に渡され、CBLモジュール1040はテンソル1042を検出モジュール1044に出力する。CBLモジュール1012、1026、および1040はそれぞれ、5つのCBLモジュール、例えば、図3Dに示されるCBLモデル360の連結を含む。アップスケーラモジュール1022及び1036は、図10Bに示すように、アップスケーラモジュール1060の各インスタンスである。モジュール1048は、テンソル1018、1032、および1036を受け取り、タスク結果151を出力する。

【0149】

図10Bに示すように、アップスケーラモジュール1060は、テンソル1062（例えば、図10Aのテンソル1014）を入力として受け付ける。テンソル1062は、CBLモジュール1066に渡されてテンソル1068を生成する。テンソル1068は、アップサンプリングされたテンソル1072を生成するために、アップサンプラ1070に渡される。連結モジュール1074は、アップサンプリングされたテンソル1072を第2入力テンソル1064（例えば、図10Aのアップスケーラ1022へ入力されるテンソル1020）と連結することによって、テンソル1076を生成する。

【0150】

検出モジュール1016、1030、および1044は、図10Cに示されるような検出モジュール1080のインスタンスである。検出モジュール1080は、テンソル1082を受信する。テンソル1082は、CBLモジュール1084に入力される。CBLモジュール1084は、テンソル1086を生成する。テンソル1086は、検出カーネルを実装する畳み込みモジュール1088に渡される。いくつかの構成では、検出カーネルが 1×1 カーネルを適用して、テンソルの3つのレイヤの各々における特徴マップ上に出力を生成する。検出カーネルは $1 \times 1 \times (B \times (5 + C))$ であり、ここで、Bは特定のセルが予測することができるバウンディングボックスの数であり、典型的には3であり、Cは80であり得るクラスの数であり、255の検出属性（すなわち、テンソル1090）のカーネルサイズをもたらす。定数「5」は、4つのバウンディングボックス属性（ボックス中心x、yおよびサイズスケールx、y）および1つのオブジェクト信頼レベル（「オブジェクトネス」）を表す。検出カーネルの結果は、入力特徴マップと同じ空間次元を有するが、出力の深さは検出属性に対応する。検出カーネルは、各レイヤ、典型的には3つのレイヤに適用され、多数の候補バウンディングボックスをもたらす。非最大抑制のプロセスは、NMSモジュール1048によって、結果として得られるバウンディングボックスに適用され、類似のスケールでの重複予測などの冗長ボックスを破棄し、結果として、オブジェクト検出のための出力としてバウンディングボックスの最終セットをもたらす。

【0151】

図11は、CNNの代替ヘッド部分1100を示す概略ブロック図である。ヘッド部分1100は、CNNヘッド150として実装することができ、CNNバックボーン114は、例えばバックボーン400として実装される。ヘッド部分1100は「高速RCNN」として知られる全体的なネットワークの一部を形成し、特徴ネットワーク（すなわち、バックボーン部分400）、リージョンプロポーザルネットワーク（a region proposal network）、および検出ネットワークを含む。ヘッド部分1100への入力、P2-P6レイヤテンソル1110、1112、1114、1116、および1118を含むテンソル149である。P2-P6レイヤテンソル1110、1112、1114、1116、および1118は、図4のP2~P6出力477、475、473、471、および429に対応する。P2-P6テンソル1110、1112、1114、1116、および1118は、リージョンプロポーザルネットワーク（RPN）ヘッドモジュール1120に入力される。RPNヘッドモジュール1120は、入力テンソルに対して畳み込みを実行し、中間テンソルを生成する。中間テンソルは、2つの後続の兄弟レイヤ、（i）

10

20

30

40

50

分類のための1つと、(ii) バウンディングボックス、または「注目領域」(ROI) 回帰のための1つと、に供給される。結果として得られる出力は、分類およびバウンディングボックス1122である。分類およびバウンディングボックス1122は、NMSモジュール1124に渡される。NMSモジュール1124は、より低いスコアを有する重複ボックスを除去することによって、冗長なバウンディングボックスを枝刈りして、枝刈りされたバウンディングボックス1126を生成する。バウンディングボックス1126は、注目領域(ROI) プーラ1128に入力される。ROIプーラ1128は、テンソル149のレイヤテンソル(以下でさらに説明する)およびバウンディングボックス1126のいくつかを使用して、最大プーリング動作を使用して様々な入力サイズマップから固定サイズ特徴マップを生成する。最大プーリング動作では、サブサンプリングが入力値の各グループ内の最大値を取得して、出力テンソル内に1つの出力値を生成する。

10

【0152】

CNNバックボーン400およびCNNヘッド1100の構成では、「P6」レイヤテンソル429が(テンソル149として受信される)出力テンソル115から省略される。P6テンソルが省略される構成では、CNNヘッド1100がP5テンソル1116に対して2に等しいストライドを有する「Maxpool」動作を実行することによって、P6入力テンソル1118を生成する。P6レイヤは、P5レイヤから再構成することができるので、PCAエンコーダ160、PCAデコーダ170において、明示的なFPNレイヤとしてP6レイヤを個別に符号化および復号する必要がない。

【0153】

ROIプーラ1128への入力、P2-P5特徴マップ1110、1112、1114、および1116、ならびに注目領域提案1126である。1126からの各提案(ROI)は、固定サイズのマップを生成するために、特徴マップ(1110~1116)の一部に関連付けられる。固定サイズのマップは、特徴マップ1110~1116の基礎となる部分とは無関係のサイズである。特徴マップ1110~1116のうちの1つは、結果として得られるクロップされたマップが例えば、以下のルールに従って十分な詳細を有するように選択される: $\text{floor}(4 + \log_2(\sqrt{\text{box_area}}) / 224)$ 、ここで224は基準のボックスサイズである。ROIプーラ1128は、テンソル1130を生成する提案1126に従って、入力特徴マップをクロップするように動作する。

20

30

【0154】

テンソル1130は、全結合(FC)ニューラルネットワークヘッド1132に供給される。FCヘッド1132は、2つの全結合レイヤを実行して、クラススコアおよびバウンディングボックス予測デルタテンソル1134を生成する。クラススコアは一般に、80要素テンソルであり、各要素は、対応するオブジェクトカテゴリの予測スコアに対応する。バウンディングボックス予測デルタテンソルは $80 \times 4 = 320$ 要素テンソルであり、対応するオブジェクトカテゴリのバウンディングボックスを含む。最終処理は出力レイヤモジュール1136によって実行され、テンソル1134を受信し、フィルタリング演算を実行して、フィルタリングされたテンソル1138を生成する。低スコアリング(低分類)オブジェクトは、さらなる考慮から除外される。非最大抑制モジュール1140は、フィルタリングされたテンソル1138を受信し、より低い分類スコアを有する重複ボックスを除去することによって重複するバウンディングボックスを除去し、テンソル151に対応する推論出力テンソル1140をもたらす。

40

【0155】

図12Aおよび図12Bはたとえば、サブピクチャビットストリームコンバイナ580によって実装される、ピクチャ1200をサブピクチャ1210、1212、1214、および1216に分割することを示す概略ブロック図である。各サブピクチャは、エンコーダ514、532、556、および572のうちの対応する1つのパッカ610によってパックされる。各サブピクチャは、サンプルの2次元アレイに配列された情報を含む。図12Bを参照すると、サブピクチャ1210は、FPNのテンソルにおけるテンソルの

50

平均特徴マップ 1 2 2 0 などの平均特徴マップを保持する。平均特徴マップ 1 2 2 0 は、図 5 の構成におけるサブピクチャエンコーダ 5 1 4 のビットストリーム 5 1 5 に対応する。サブピクチャ 1 2 1 2 は、図 5 の 5 3 3 に対応する平均係数 1 2 2 2 を保持する。平均係数 1 2 2 2 は、符号化レイヤまたは F P N レイヤごとに 1 つの行と、対応するテンソルのチャンネルごとに 1 つの値とを有する値の行に配列される。符号化レイヤの数および次元は、図 1 8 A、1 8 B、1 9、および 2 0 を参照して説明したように、テンソル再サンプリングおよび／または連結に基づいて F P N レイヤの数および次元とは異なり得る。サブピクチャ 1 2 1 6 は、図 5 の 5 5 7 に対応する基底ベクトルを保持する。図 1 2 B の例では、基底ベクトルがとりわけ、基底ベクトル 1 2 2 6 を含み、基底ベクトルは重複しない方法でサブピクチャ 1 2 1 6 のエリアにパックされる。基底ベクトルは隣接してパックされてもよく、または、例えば、基底ベクトルの各セット間の 1 つまたは 2 つのサンプルの間隔を含んでもよい。各符号化レイヤまたは F P N レイヤの基底ベクトルは、サブピクチャ 1 2 1 6 にパックされる。ピクチャ 1 2 0 0 内の、1 2 0 8 などのデータを記憶するために使用されないエリアは、サンプル値から浮動小数点領域に変換するための逆量子化の適用後に、値「0」に対応するサンプル値によって占有され得る。同様に、サブピクチャ 1 2 1 0、1 2 1 2、1 2 1 4、および 1 2 1 6 内の、データを記憶するために使用されないエリアは、サンプル値から浮動小数点領域に変換するための逆量子化の適用後に、値「0」に対応するサンプル値によって占有され得る。サブピクチャ 1 2 1 4 は、図 5 の 5 7 2 に対応する係数を保持し、F P N のテンソルにおけるテンソル内の各チャンネルの各基底ベクトルに対して 1 つの係数を有する。例えば、テンソル内の 2 5 6 個のチャンネルでは 2 5 個の基底ベクトル、1 つのテンソルに対する係数は 2 5 6 × 2 5 個のサンプルアレイとして配置される。

10

20

【0 1 5 6】

複数のテンソルがピクチャ 1 2 0 0 において符号化される場合、対応するサンプルアレイは、サブピクチャ 1 2 1 4 において積み重ねられる。量子化された特徴マップはサンプルの 2 次元アレイであり、係数もまた、サンプルの 1 次元または 2 次元アレイにマッピングされるので、ピクチャ 1 2 0 0 のためのモノクロフォーマットの使用は十分である。C N N バックボーン 1 1 4 がフレーム 1 1 3 のためのテンソル 1 1 5 を生成したとき、P C A エンコーダ 1 6 0 は、テンソル 1 1 5 の次元に基づいてサブピクチャ 1 2 1 0、1 2 1 2、1 2 1 4、および 1 2 1 6 のサイズを決定することができる。サブピクチャ 1 2 1 0、1 2 1 2、および 1 2 1 6 は C T U の整数として、すなわち、1 2 8 ルマサンプルの倍数として、幅および高さにおいて指定され得る。サブピクチャ 1 2 1 6 の幅は、水平方向に配置されたサブピクチャ 1 2 1 0、1 2 1 2、および 1 2 1 4 の幅の合計に等しい。サブピクチャ 1 2 1 0、1 2 1 2、および 1 2 1 6 の下に配置されたサブピクチャ 1 2 1 6 の高さは、パックされた基底ベクトル、たとえば基底ベクトル 1 2 2 6 に十分なエリアを提供するように設定される。

30

【0 1 5 7】

図 1 2 A および 1 2 B の例では、量子化係数 5 1 3、5 3 0、および 5 6 8 にそれぞれ対応するサブピクチャ 1 2 1 0、1 2 1 2、1 2 1 4 が水平に配置され、ピクチャ 1 2 0 0 の幅を定義する。量子化係数 5 5 4 に対応するサブピクチャ 1 2 1 6 は、サブピクチャ 1 2 1 0、1 2 1 2、1 2 1 4 の幅にわたり、ピクチャ 1 2 0 0 の底部まで延在する、第 1、第 2、および第 3 ユニットの下に位置する。別の実装形態では、量子化係数がサブピクチャ以外のフォーマットで構成され得る。たとえば、V V C または H E V C の場合、量子化係数は、スライスまたはタイルとして配置またはパックされ得る。さらに他の実装形態では、量子化係数が符号化タイプ（たとえば、V V C または H E V C）および予想されるデータ特性などの要因に応じて、異なるように構成され得る。たとえば、量子化係数は、変換が使用されないときに量子化係数の順序付けられたセットに残差符号化が適用されるように、変換スキップが適用された小さいブロックの走査順序で配列され得る。たとえば、量子化係数が H E V C を使用して圧縮されるとき、変換スキップが 4 × 4 ブロックに適用され得、量子化係数は 4 × 4 ブロックへの変換の適用なしに残差符号化を活用するた

40

50

めに、 4×4 ブロック中の対角走査パターン中に構成され得る。基底ベクトルの数が16以下である場合、各特徴マップの量子化係数は、別個の 4×4 ブロックに記憶され得る。各 4×4 ブロック内で、量子化係数の圧縮効率は、 4×4 ブロック内の適応ライスパラメータ符号化の適用から恩恵を受ける。別個のサブピクチャの代わりに、量子化係数の各セットに対応する別個のスライスを用いて、単一のピクチャを他の実装形態で 사용할ことができ、または各サブピクチャをピクチャ1200とは異なるように配置することができる。

【0158】

ピクチャ1200の構成では、係数1222のサブピクチャ1212へのパッキングがNがチャンネルの数である $N \times 1$ アレイから $W \times H$ アレイにアレイを修正し、ここで、 $W \times H$ はNに等しく、サブピクチャのアスペクト比を、可能な限り正方形（1：1）または2：1比などの長方形に近くなるように調整するが、 $N \times 1$ の極端に細長い比を回避するように選択される。正方形に近いアスペクト比は、パックされた係数サンプルのエッジに隣接するサンプルが少なくなり、アンパッカによって使用されないが、依然として比較的高い忠実度で符号化され、したがって、テンソル再構成に何ら利益を及ぼすことなく残差符号化コストに寄与する。

【0159】

ピクチャ1200の別の構成では、サブピクチャ1212または1214への係数1222または1224のパッキングが、各係数が2つ以上のサンプルにパッキングされるように実行される。たとえば、係数は、パッカ610によって 2×2 のサンプルのセットにパッキングされ得る。アンパッカ620または832または812は、 2×2 セットのサンプルを読み取り、フィルタリング動作を実行して係数を復元する。2つ以上のサンプルへのパッキングは、ビデオエンコーダ614における非可逆符号化の使用に起因して係数値に導入される誤差を補償するための機構を提供する。 2×4 、 4×2 、 4×4 を含む、サンプルのセットの他の構成も可能である。

【0160】

図13は、符号化されたパッキングされた特徴マップおよび関連するメタデータを保持するビットストリーム1300を示す概略ブロック図である。ビットストリーム1300は、PCAエンコーダ160によって生成されたビットストリーム121またはPCAデコーダ170によって復号されたビットストリーム143に対応する。ビットストリーム1300は、「ネットワーク抽象化レイヤ」ユニットヘッダによって前置されたシンタックスのグループを含む。例えば、NALユニットヘッダ1308は、シーケンスパラメータセット（SPS）1310に先行する。SPS1310はプロファイル（符号化ツールのセット）およびレベルおよびティアを指定する（たとえば、最大サンプルレートおよび圧縮ビットレートに関して動作点を共同で指定する）シンタックスの「プロファイルレベルティア」（PLT）ユニットを含み得る。シンタックスのPLTユニットは、選択されたプロファイルと比較して利用可能なツールのセットをさらに制約するシンタックスの「一般制約情報」（GCI）ユニットを含み得る。SPS1310は、サブピクチャ1210、1212、1214、および1216の位置およびサイズを含む、ピクチャ1200のレイアウトを指定する。GCIはフラグのセットを含み、各フラグは、ビットストリーム1300において使用されない特定の符号化ツールを制約する。PLTは、ツールの特定のセットがビットストリーム1300において使用可能であることをシグナリングし、ツールの特定のセットは「プロファイル」として知られている。プロファイルの一例は「Main 10」であり、4：0：0または4：2：0クロマフォーマットのいずれかを有する8～10ビットビデオを提供し、広範な展開をターゲットとする。GCIは、「サブプロファイル」として知られる、ツールのサブセットへのプロファイルのツールのセットに対するさらなる制約を示し得る。PCAエンコーダ160は、フレームにパッキングされた特徴マップを符号化する（すなわち、図5に従って動作する）際に、VVC規格の特定のツールは、圧縮の利点を提供しない。パッキングされた特徴マップに圧縮利点を提供しないツールは、ビデオエンコーダ614によって試される必要はなく、ビットストリ

10

20

30

40

50

ーム1300において使用されないものとしてGCIにおいてシグナリングされ得る。SPS1310はまた、ビットストリーム1300によって表されるフレームデータの解像度、ビット深度、クロマフォーマット、を示す。

【0161】

PCAエンコーダ160によって使用されるパッキングフォーマットは、すべての利用可能な特徴パッキングフォーマットの列挙から1つの特徴パッキングフォーマットを選択するためのインデックスを使用して、SEIメッセージ1313において符号化され得る。特徴マップを生成するために使用された特定のCNNバックボーンは、SEIメッセージ1313において、インデックスを使用して、1組の所定のCNNバックボーンの列挙から1つのCNNバックボーンを選択することも示され得、そのうちのいくつかまたはすべてがソースデバイス110に利用可能である。CNNバックボーンタイプインデックスから、各レイヤにおけるチャンネルの数およびレイヤの数、ならびに各レイヤにおける各特徴マップの解像度が決定され得る。基底ベクトル550のサイズ（すなわち、幅および高さ）、基底ベクトル550の最大個数（すなわち、FPNの各レイヤの最大個数、またはテンソルコンバイナ162から得られるFPNの縮小形成）、および使用された基底ベクトル550の数（すなわち、方法2100から得られる）は、付録Aを参照して説明される、SEIメッセージ1313内の基底ベクトルパッキング情報1391として符号化される。テンソル抽出器172内で、テンソル115aと同じ数および次元数を有するテンソル149を復元するために（テンソルコンバイナ162によって決定される）実行される動作は、レイヤマッピング1390として説明される。レイヤマッピング1390は、SEIメッセージ1313内に存在する。レイヤマッピング1390は、付録Aを参照して説明した、`layer__update__mapping__info`、`layer__extraction__flag`、`layer__upsample__flag`、`layer__downsampling__flag`、`resampling__filter__idx`、`resample__vertical__only__flag`、`resample__horizontal__only__flag`、`src__layer__idx`、`src__channel__offset`、`dst__layer__idx`を含む。

【0162】

ピクチャ1314は、ビットストリーム1300において符号化される。各ピクチャは、符号化サブピクチャ1320、符号化サブピクチャ1210などの1つまたは複数のサブピクチャを含む。ビットストリームの第1ピクチャについて、一般に「ランダムアクセスポイント」アクセスユニットについて、イントラスライスは、ビットストリーム1300中の他のアクセスユニットに対するいかなる予測依存性も回避するために使用される。サブピクチャ1212を符号化する符号化サブピクチャ1322は、スライスヘッダ1330と、それに続くスライスデータ1340とを含む。スライスデータ1340は、フレームデータの符号化表現を与えるCTUのシーケンスを含む。CTUは正方形であり、典型的には128×128のサイズであり、これは典型的な特徴マップサイズに十分に整合されない。特徴マップの4×4グリッドなどの最小ブロックサイズへのアライメントは、この不整合を部分的に改善する。整数またはサンプル領域から浮動小数点領域への変換を可能にするために、量子化範囲1392がビットストリーム1300に存在し、浮動小数点の最小値および最大値を定義し、そこから、例えば、520、536、560、846、836、824、816において実行されるように、逆量子化プロセスにおいてサンプルがマッピングされる。量子化範囲は、量子化モジュール、すなわち、512、528、552、および568において決定され、ビットストリーム121において、SEIメッセージ1313における量子化範囲1392として符号化される。符号化サブピクチャ1324は、サブピクチャ1214を符号化し、符号化サブピクチャ1326は、サブピクチャ1216を符号化する。符号化サブピクチャ1220、1222、1224、および1226は、それぞれ、ビットストリーム部分515、533、557、および573に対応する。別個のサブピクチャの代わりに、単一のピクチャを、サブピクチャ1320、1322、1324、および1326に対応する別個のスライスとともに使用することも

できる。

【0163】

図14は、CNNの第1部分を実行し、ビデオデータのフレームのために結果として得られる特徴マップを符号化するための方法1400を示す。特徴マップを符号化する際に、テンソルがビットストリームに符号化される。方法1400は、構成されたFPGA、ASIC、またはASSPなどの装置を使用して実装され得る。代替的に、以下で説明するように、方法1400は、プロセッサ205の実行下で、アプリケーションプログラム233の1つまたは複数のソフトウェアコードモジュールとして、ソースデバイス110によって実装され得る。方法1400を実施するアプリケーションプログラム233のソフトウェアコードモジュールは例えば、ハードディスクドライブ210および／またはメモリ206に常駐することができる。方法1400は、ビデオソース112によって生成されたビデオデータの各フレームに対して繰り返される。方法1400は、コンピュータ可読記憶媒体および／またはメモリ206に記憶され得る。

10

【0164】

方法1400は、CNN第1部分実行ステップ1410で開始する。ステップ1410において、CNNバックボーン114は、プロセッサ205の実行下で、特定のCNNのレイヤのサブセットを実行して、入力フレーム113を中間テンソル115aに変換する。中間テンソル115aは例えば、メモリ206および／またはハードディスクドライブ210に記憶され得る。CNNの例は図4に示されるように、「より速いR-CNN」または「マスクR-CNN」であり、レイヤのサブセットは、「Pレイヤ」分割点までのすべてのレイヤに対応する。例えば、FPNの使用のために、CNNバックボーン114から複数のテンソルが抽出されるとき、テンソル115aは、複数のテンソルを含み得る。方法1400は、ビデオソース112からのビデオデータの1つのフレームに対応するテンソルを符号化するように動作する。プロセッサ205における制御は、ステップ1410からテンソル低減実行ステップ1415に進む。

20

【0165】

ステップ1415において、テンソルコンバイナ162は、プロセッサ205の実行下で動作して、テンソルの数を再サンプリングする。図19は、ステップ1415で実施される方法1900を示す。ステップ1415の動作の結果として、FPNからのテンソル115aは、(i)ダウンサンプリングされ(数が削減され)、(ii)アップサンプリングされ(数が増加され)、または(iii)直接通過されて、テンソル115を生成することができる。アップサンプリングまたはダウンサンプリングは、2つのレイヤからのテンソルが互換性のある幅および高さで連結されるように実行される。テンソル115aの所与のテンソルをアップサンプリングまたはダウンサンプリングするか、またはすべてのテンソル115aを変更せずに残すかの決定は、ビットストリーム121において符号化される。プロセッサ205における制御はステップ1415から、平均特徴マップ決定ステップ1420に進む。

30

【0166】

平均特徴マップ決定ステップ1420において、モジュール510は、プロセッサ205の実行下で、チャンネル寸法にわたってテンソル115を平均化して、平均特徴511を生成する。ステップ1420において、平均特徴511は、プロセッサ205の実行下で量子化器512によって量子化され、整数(量子化)平均特徴マップ513を生成する。プロセッサ205における制御は、ステップ1420から平均特徴符号化ステップ1430に進む。

40

【0167】

平均特徴符号化ステップ1430において、サブピクチャエンコーダ514は、プロセッサ205の実行下で、整数平均特徴マップ513をサブピクチャ1210にパッキングし、サブピクチャ1210を符号化してビットストリーム部分515を生成する。プロセッサ205における制御は、ステップ1430から再構成平均特徴リカバーステップ1440に進む。

50

【0168】

再構成平均特徴リカバーステップ1440において、サブピクチャエンコーダ514は、プロセッサ205の実行下で、ビデオ圧縮のためのサブピクチャ入力の非可逆バージョン、たとえば612に対応する、エンコーダ514の実装形態内の再構成されたピクチャ、たとえば618を出力する。プロセッサ205における制御は、ステップ1440から平均係数決定ステップ1450に進む。

【0169】

平均係数決定ステップ1450において、逆量子化器520は、プロセッサ205の実行下で、整数平均特徴マップ516を逆量子化し、マップ516を特徴マップ522として浮動小数点領域に変換して戻すように動作する。ドット積524は、プロセッサ205

10

【0170】

平均係数符号化ステップ1460において、量子化器モジュール528は、プロセッサ205の実行下で平均係数526mを量子化して、整数平均係数530を生成（generate）または生成（produce）する。サブピクチャエンコーダ532は、プロセッサ205の実行下で、整数平均係数530をサブピクチャ1212にパッキングし、サブピクチャ1212を符号化してビットストリーム部分533を生成する。プロセッサ

20

【0171】

再構成平均係数リカバーステップ1470において、サブピクチャエンコーダ532は、プロセッサ205の実行下で、サブピクチャ1212の再構成されたバージョンを出力し、そこから再構成された平均係数534が生成される。再構成された平均係数534は、モジュール536によって逆量子化されて、リカバーされた平均係数538を生成する。ステップ1470において、ピクチャの再構成されたバージョンは、エンコーダ532の実装内の618に対応する。プロセッサ205における制御は、ステップ1470から基底ベクトル決定ステップ1480に進む。

30

【0172】

基底ベクトル決定ステップ1480において、逆量子化器536、ドット積モジュール540、減算器544、および分解モジュール548は、プロセッサ205の実行下で動作して、テンソル115のための基底ベクトルのセットを生成する。逆量子化器536は、浮動小数点平均係数538を出力する。ドット積モジュール540は、リカバーされた平均特徴マップ522およびhte係数538を用いてドット積演算を実行し、オフセットテンソル542を生成する。テンソル542はテンソル115と同じ次元を有し、平均（平均）特徴マップ情報のみを使用して予測される、テンソル115内の各チャンネルの特徴マップの予測を含む。したがって、テンソル542は、このチャンネルごとの平均特徴マップデータを減算することによって、テンソル115のゼロ平均バージョンの生成を可能にする。オフセット特徴マップ542は、モジュール544によってテンソル115から減算されて、ゼロ中心テンソル546を生成する。分解モジュール548は、テンソル546を入力として受け取り、基底ベクトルのセット550を生成する。セット550中の各基底ベクトルは、「成分」と呼ばれることがある。図21を参照して、基底ベクトル550を生成する際の分解モジュール548の動作を説明する。ベクトル550は、テンソル115内のチャンネルよりも少ない基底ベクトルを含み、テンソル115の次元数の減少に対応する。550内の基底ベクトルは、基底ベクトル550内の成分の数に対するテンソル115内の最大分散量を説明する部分空間内のテンソル115を表す。言い換えれば、基底ベクトル550は、所与の数の成分についての品質の最小限の劣化でテンソル115の表現を可能にする。プロセッサ205における制御は、ステップ1480から基底ベ

40

50

クトル符号化ステップ 1 4 9 0 に進む。

【 0 1 7 3 】

基底ベクトル符号化ステップ 1 4 9 0 において、量子化器モジュール 5 5 2 は、プロセッサ 2 0 5 の実行下で、基底ベクトル 5 5 0 を整数領域に量子化するように動作する。得られた整数基底ベクトル 5 5 4 は、サブピクチャにパッキングされ、符号化されて、プロセッサ 2 0 5 の実行下でサブピクチャエンコーダ 5 5 6 によってビットストリーム部分 5 5 7 を生成する。プロセッサ 2 0 5 における制御は、ステップ 1 4 9 0 から再構成基底ベクトルリカバーステップ 1 4 1 0 0 に進む。

【 0 1 7 4 】

ステップ 1 4 1 0 0 において、サブピクチャエンコーダ 5 5 6 は、プロセッサ 2 0 5 の実行下で、再構成された整数テンソル 5 5 8 を生成し、サブピクチャ 1 3 1 6 の再構成されたバージョン、たとえば、6 1 8 を取得するように動作する。基底ベクトル 5 5 8 は、再構成されたサブピクチャからアンパックされ、プロセッサ 2 0 5 の実行下で逆量子化器 5 6 0 によって（再構成された基底ベクトル 5 6 2 として）浮動小数点領域に逆量子化される。プロセッサ 2 0 5 における制御は、ステップ 1 4 1 0 0 から係数決定ステップ 1 4 1 1 0 に進む。

【 0 1 7 5 】

ステップ 1 4 4 0、1 4 7 0、および 1 4 1 0 0 の各々は、再構成された係数をリカバーする。各ステップで生成された再構成された係数は、ステップ 1 4 3 0、1 4 0、および 1 4 9 0 で量子化および符号化された特徴またはベクトルの再構成を提供する。各再構成は、デコーダ側で見られる特徴またはベクトルのバージョンがビットストリーム 1 2 1 を符号化する際に使用されるものを反映するように、モデル化されるサブピクチャエンコーダによって実行される符号化動作による損失をモデル化して考慮することを可能にする。実装形態 5 0 0 において符号化ビットストリームを生成するために使用される量子化機能（たとえば、5 1 2、5 2 8、および 5 5 2 において）の動作によって被る損失も考慮される。

【 0 1 7 6 】

ステップ 1 4 1 1 0 において、ドット積モジュール 5 6 4 は、プロセッサ 2 0 5 の実行下で、再構成された基底ベクトル 5 6 2 内の各ベクトルに対してテンソル 5 4 6 内の各チャンネルのドット積を実行して、係数のセット、すなわち係数 5 6 6 を生成する。係数 5 6 6 は、各特徴マップのコンテンツを再生する際の各基底ベクトルの寄与を表す。プロセッサ 2 0 5 における制御は、ステップ 1 4 1 1 0 から係数符号化ステップ 1 4 1 2 0 に進む。

【 0 1 7 7 】

ステップ 1 4 1 2 0 において、量子化器モジュール 5 6 8 は、プロセッサ 2 0 5 の実行下で、係数 5 6 6 を量子化して、整数係数 5 7 0 を生成する。整数係数 5 7 0 は、サブピクチャ 1 2 1 4 にパッキングするためにサブピクチャエンコーダ 5 7 2 に渡される。サブピクチャ 1 2 1 4 は、ビットストリーム部分 5 7 3 を生成するためにモジュール 5 7 2 によって符号化される。プロセッサ 2 0 5 における制御は、ステップ 1 4 1 2 0 からサブピクチャ合成ステップ 1 4 1 3 0 に進む。テンソル 1 1 5 が 2 つ以上のテンソルを含む場合、ステップ 1 4 2 0 ~ 1 4 1 2 0 はテンソル 1 1 5 内の各テンソルについて繰り返され、最終圧縮表現はステップ 1 4 1 3 0 において単一のピクチャに構成される。

【 0 1 7 8 】

上述したように、モジュール 5 5 2 ~ 5 7 2 は、モジュール 5 1 2 ~ 5 3 2 と同様に動作して、ビットストリームを生成する。ステップ 1 4 1 3 0 において、サブピクチャビットストリームコンバイナ 5 8 0 は、プロセッサ 2 0 5 の実行下で、サブピクチャ 1 2 1 0、1 2 1 2、1 2 1 6、および 1 2 1 4 にそれぞれ対応するビットストリーム部分 5 1 5、5 3 3、5 5 7、および 5 7 3 を単一のビットストリーム、すなわち 1 2 1 に結合する。

【 0 1 7 9 】

10

20

30

40

50

方法 1400 は完了し、プロセッサ 205 における処理は、次のフレームのために方法 1400 を呼び出すように進む。

【0180】

図 15 は、分割 CNN の第 2 部分を実行するために PCA デコーダ 800 によって実装されるような復号テンソル 149 を生成するためにビットストリーム 143 を復号するための方法 1500 を示す。方法 1500 は、構成された FPG A、ASIC、または ASSP などの装置を使用して実装され得る。代替的に、以下で説明するように、方法 1500 は、プロセッサ 205 の実行下で、アプリケーションプログラム 233 の 1 つまたは複数のソフトウェアコードモジュールとして、ソースデバイス 110 によって実装され得る。方法 1500 を実施するアプリケーションプログラム 233 のソフトウェアコードモジュールは例えば、ハードディスクドライブ 210 および／またはメモリ 206 に常駐することができる。方法 1500 は、コンピュータ可読記憶媒体および／またはメモリ 206 に記憶され得る。方法 1500 は、平均特徴復号ステップ 1510 で開始する。

10

【0181】

ステップ 1510 において、ピクチャデコーダ 804 は、プロセッサ 205 の実行下で、ビットストリーム 143 から（サブピクチャ 840 に対応する）サブピクチャ 1210 を復号する。プロセッサ 205 における制御は、ステップ 1510 から平均係数復号ステップ 1520 に進む。

【0182】

ステップ 1520 において、ピクチャデコーダ 804 は、プロセッサ 205 の実行下で、ビットストリーム 143 から（サブピクチャ 830 に対応する）サブピクチャ 1212 を復号する。プロセッサ 205 における制御は、ステップ 1520 から DC 成分決定ステップ 1530 に進む。

20

【0183】

ステップ 1530 において、アンパッカ 842 および 832、逆量子化器 846 および 836、ならびにドット積モジュール 850 は、プロセッサ 205 の実行下で、テンソル 852 を生成する。アンパッカ 842 および 832 は、サブピクチャをアンパックし、各サブピクチャは平均特徴 511 に関する量子化された特徴マップを含み、実装 500 の量子化機能を使用して量子化される。逆量子化器 846 は、アンパッカ 842 によってサブピクチャ 840 からアンパックされた平均特徴マップを逆量子化して、平均特徴マップ 848 を導出する。逆量子化器 836 はアンパッカ 832 によってサブピクチャ 830 からアンパックされた平均係数を、ビットストリーム 143 においてシグナリングされるそれぞれの量子化範囲を使用して整数領域から浮動小数点領域に逆量子化し、平均係数 838 を生成する。ドット積モジュール 850 は、平均特徴 848 および平均係数 838 に対してドット積を実行してテンソル 852 を生成する。テンソル 852 は、ゼロ中心テンソルからの非ゼロ中心テンソルの再構成のためのオフセットを表す。プロセッサ 205 における制御は、ステップ 1530 から基底ベクトル復号ステップ 1540 に進む。

30

【0184】

ステップ 1540 において、ピクチャデコーダ 804 は、プロセッサ 205 の実行下で、ビットストリーム 143 から（サブピクチャ 820 に対応する）サブピクチャ 1216 を復号する。サブピクチャ 1216 は、復号サブピクチャ 820 としてアンパッカ 822 に渡される。アンパッカ 822 は、ステップ 1560 において、逆量子化のために、復号された基底ベクトル、例えば 1226 を抽出する。図 22 を参照して、ビットストリーム 143 から基底ベクトルを復号する際の宛先デバイス 140 の動作について説明する。プロセッサ 205 における制御は、ステップ 1540 から係数復号ステップ 1550 に進む。

40

【0185】

ステップ 1550 において、ピクチャデコーダ 804 は、プロセッサ 205 の実行下で、ビットストリーム 143 から（サブピクチャ 810 に対応する）サブピクチャ 1214 を復号する。サブピクチャ 1214 は、復号サブピクチャ 810 としてアンパッカ 812

50

に渡される。アンパッカ 8 1 2 は、ステップ 1 5 6 0 において、逆量子化のために、復号された係数、例えば、1 2 2 4 を抽出する。プロセッサ 2 0 5 における制御は、ステップ 1 5 5 0 から A C 成分決定ステップ 1 5 6 0 に進む。

【0 1 8 6】

ステップ 1 5 6 0 において、アンパッカ 8 2 2 および 8 1 2、逆量子化器 8 2 4 および 8 1 6、ならびにドット積モジュール 8 2 8 は、プロセッサ 2 0 5 の実行下で、ゼロ中心テンソル 8 5 4 を生成するように動作する。アンパッカ 8 2 2 は、サブピクチャ 8 2 0 から基底ベクトルを抽出し、基底ベクトルを整数特徴マップ 8 2 2 として逆量子化器 8 2 4 に渡す。逆量子化器は、ビットストリーム 1 4 3 から得られた量子化範囲に従って、基底ベクトルを整数領域から浮動小数点領域に変換し、基底ベクトル 8 2 6 を生成する。アンパッカ 8 1 2 は、サブピクチャ 8 1 0 から係数を抽出し、係数を整数係数として逆量子化器 8 1 6 に渡す。逆量子化器 8 1 6 は、浮動小数点領域係数 8 1 8 を生成する。ドット積モジュール 8 2 8 は、係数 8 1 8 の各係数とベクトル 8 2 6 内のそれぞれの基底ベクトルとのドット積を実行して、ゼロ中心再構成テンソル、すなわちテンソル 8 5 4 を生成する。プロセッサ 2 0 5 における制御は、ステップ 1 5 6 0 からテンソル生成ステップ 1 5 7 0 に進む。

10

【0 1 8 7】

ステップ 1 5 7 0 において、加算モジュール 8 5 6 は、プロセッサ 2 0 5 の実行下で、ゼロ中心再構成テンソル 8 5 4 をテンソル 8 5 2 と合計して、再構成テンソル 1 4 9 を生成する。P C A プロセスの最適な、またはほぼ最適な次元削減のために、テンソル 1 4 9 はテンソル 1 1 5 と実質的に同様であり、したがって、所与のマシントスクを実行するための C N N ヘッド 1 5 0 による使用に適している。F P N が使用されている場合、ステップ 1 5 1 0 ~ 1 5 7 0 は、方法 1 9 0 0 から得られた F P N レイヤの表現の各テンソルに対して実行される。プロセッサ 2 0 5 における制御は、ステップ 1 5 7 0 からテンソル抽出ステップ 1 5 7 5 に進む。

20

【0 1 8 8】

ステップ 1 5 7 5 において、プロセッサ 2 0 5 は、テンソル抽出器 1 7 2 の動作によって、任意の複合テンソルから別個の F P N レイヤテンソルを抽出するために実行する。テンソル抽出器 1 7 2 は、図 2 0 に示される方法 2 0 0 0 を実行して、テンソル 1 4 9 a の任意の複合テンソルから別個の F P N レイヤテンソルを抽出し、その結果をテンソル 1 4 9 に沿って渡す。プロセッサ 2 0 5 における制御は、ステップ 1 5 7 5 からニューラルネットワーク第 2 部分実行ステップ 1 5 8 0 に進む。

30

【0 1 8 9】

ステップ 1 5 8 0 において、C N N ヘッド 1 5 0 は、プロセッサ 2 0 5 の実行下で、テンソル 1 4 9 を入力として使用してニューラルネットワークの後段を実行する。例えば、図 1 0 A 又は図 1 1 を参照して説明したネットワークを用いた分類は、テンソル 1 4 9 の次元数（及び、その中に含まれるテンソルの数、F P N を用いる場合）に応じて行うことができる。さらに、いくつかのネットワークがバックボーン部分にマッチングアーキテクチャおよび重みを有する場合、そのようなバックボーン部分から符号化されたテンソルを使用して、C N N ヘッド 1 5 0 の異なるインスタンスを実行することによって、これらの異なるネットワークの完了を実行することができる。ステップ 1 5 8 0 は、タスク結果 1 5 1 を出力するために実行される。

40

【0 1 9 0】

図 1 6 A および図 1 6 B は、固有ベクトル「不安定性」の概念を示すために、2 つの連続するデータセットおよび関連する固有ベクトルを示し、これは、類似のデータセットに対して分解が繰り返し実行されるときに生じる。非相関テンソルがビデオから生成されるとき、連続するフレームは強く相関され、したがって、結果として得られる基底ベクトルも極めて類似している。図 1 6 A の第 1 データセット 1 6 0 0 は、2 次元データセットを示し、図 1 6 B の第 2 データセット 1 6 4 0 は、2 次元データセットの後のバージョンを示す。データセット 1 6 0 0 および 1 6 4 0 は、類似しているが同一ではなく、ビデオデ

50

バイス112からのフレームキャプチャなどのサンプリングプロセスに対してCNNバックボーン114動作を実行することによって導出され得る。特徴マップ内の各サンプルは、システム100における分解の目的のために別個の次元を形成するが、説明の目的のために、2次元データセットが図16Aおよび16Bに示される。単一の固有ベクトルが生成され、2次元から1次元への削減をもたらす。データセット1600については、固有ベクトル1610が生成され、データセット1640については、固有ベクトル1650が生成される。データセット1600および1640が類似しているため、固有ベクトル1610および1650は類似の絶対勾配を有し、「勾配」は、主成分の切片が原点を通ることに留意して、2次元ベクトルの場合におけるベクトルの方向を反映する。しかしながら、データセット1600および1640におけるわずかな差異のために、分解を実行するためのプロセスは、固有ベクトルの方向を「反転 (flip)」または反転 (invert) させ、逆方向を有するベクトル1610および1650をもたらすことがある。言い換えれば、1650の反転は、ベクトル1610のそれぞれの要素の負である各要素（すなわち、xおよびy成分）を表す。基底ベクトルの方向の反転は、以前に生成された基底ベクトルのそれぞれの要素と同じ（または同様の）大きさを有するが符号が反対であるベクトルの各要素に対応する。基底ベクトルの場合、図16Aおよび16Bの例に示されるよりも多くの次元が存在する。例えば、136×72の特徴マップサイズは、9792個の要素または9792個の次元を有する基底ベクトルを示す。したがって、ベクトル1650に負の1を乗算すると、ベクトル1610と実質的に同じ値を有するベクトルが得られる。

10

20

【0191】

テンソル115の特徴マップにわたる分解が繰り返し実行されるとき、結果として生じる基底ベクトルの要素ごとの反転の同じ問題が生じる。個々の基底ベクトルは、たとえば基底ベクトルの内容が実質的に類似していても、時々その方向を反転させることができる。この極性反転は任意の基底ベクトルに対して発生する可能性があり、所与の基底ベクトルに対する発生は、他の基底ベクトルに対する発生とは無関係である。パッキングされた基底ベクトルを含む連続するフレームを圧縮するとき、反転された基底ベクトルに対応する反転された領域がビデオエンコーダ614による効率的なツール選択のために容易に認識されないため、反転は、高い符号化効率を達成するためのインター予測の能力を低減する。

30

【0192】

図17は、CNNの第1部分からの所与のレイヤから連続する特徴マップを符号化するための方法1700を示し、連続する特徴マップは同じ次元数を有する。方法1700は、低減された次元空間における表現がPCAの呼び出し間の指向性に関して安定化されるように動作する。方法1700は、ビデオデータを処理するのに特に適しており、低減された次元表現は、複数回、おそらくフレームごとに1回の頻度で再生成される必要があり、連続するビデオフレームに相関があり、連続するビデオフレームからのテンソルに対してPCAを実行することによって生成される基底ベクトルに相関が生じる。そのような場合、結果として生じる基底ベクトルの方向は、基底ベクトル内の全体的な基底およびサンプルごとの大きさが比較的不变であるにもかかわらず、基底ベクトルのすべてのサンプルにわたって1つの呼び出しから次の呼び出しに反転し得る。そのような「極性反転」は、基底ベクトル生成プロセスにおける不安定性、または入力テンソルにおける小さな変化に対する感度に起因し得る。方法1700は、構成されたFPGA、ASIC、またはASSPなどの装置によって実装され得る。方法1700は、プロセッサ205の実行下で、アプリケーションプログラム233の1つまたは複数のソフトウェアコードモジュールとして、たとえばPCAエンコーダ160の機能として、宛先デバイス140によって実装され得る。方法1700は、ビットストリーム143において符号化されたビデオデータの各フレームに対して繰り返される。方法1700を実施するアプリケーションプログラム233のソフトウェアコードモジュールは例えば、ハードディスクドライブ210および/またはメモリ206に記憶され得る。方法1700は、第1PCA実行ステップ17

40

50

10で開始する。

【0193】

ステップ1710において、PCAエンコーダ160は、プロセッサ205の実行下で、第1テンソルの基底ベクトルの第1セットを生成し、第1テンソルは、ビデオソース112からの所与のフレームを処理するCNNバックボーン114から生成される。例えば、分解モジュール548は、テンソル546を受信することができる。プロセッサ205における制御は、ステップ1710から基底ベクトル格納ステップ1720に進むステップから進む。

【0194】

ステップ1720において、PCAエンコーダ160は、プロセッサ205の実行下で、分解モジュール548の後続の呼び出しにおいて使用するために、基底ベクトルの第1セットをメモリ206に格納する。プロセッサ205における制御は、ステップ1720から第2PCA実行ステップ1730に進む。

【0195】

ステップ1730において、PCAエンコーダ160は、プロセッサ205の実行下で、第2テンソルの基底ベクトルの第2セットを生成し、第2テンソルは、ビデオソース112からのステップ1710の所与のフレームに後続するフレームを処理するCNNバックボーン114から生成される。ステップ1730は、ステップ1720と同様に動作し、すなわち、分解モジュール548を使用する。後続のフレームは、ビデオソース112によって提供される次のフレームであってもよく、または後のフレームであってもよい。ステップ1730で生成された基底ベクトルは、現在のテンソルの成分を提供し、一方、ステップ1720で記憶された基底ベクトルは、先行するテンソルに関連する。基底ベクトルの生成がすべてのフレームに対してより少ない頻度で実行される場合、生成は周期的（Nフレームごと）であり得るか、またはフレームコンテンツが有意な新しい情報を含むとみなされるときなど、フレームのコンテンツに関するいくつかの基準によって駆動され得る。例えば、基底ベクトルのさらなる生成は、ビデオソース112によって提供される以前のフレームと比較して大きな動きまたは他の有意な変化の検出のために実行され得る。ビデオソース112からのビデオフレームにおける相関のために、基底ベクトルの第1および第2セットは、ある程度の相関を有し、図12Aおよび12Bのパッキングされた表現においてインター予測ツールを使用する圧縮に適した基底ベクトルを作成する。しかしながら、第1フレームと第2フレームとの間の雑音または他の小さな変動は、基底ベクトルの第1セットのそれぞれの基底ベクトルと比較して、反対の「方向」に生成される（または反転される）基底ベクトルの第2セットの様々な基底ベクトルをもたらし得る。プロセッサ205における制御は、ステップ1730から基底ベクトル反転チェックステップ1740に進む。

【0196】

ステップ1730で生成された基底ベクトルは、現在のテンソルの成分を提供し、一方、ステップ1710で生成された基底ベクトルは、ビデオデータ113内の以前のフレームを表す先行テンソルの成分を提供する。ステップ1740では、現在の基底ベクトルが極性反転をチェックするためにテストされる。極性反転をチェックすることは、FPNの所与の符号化されたレイヤまたはレイヤについて、現在（ステップ1730）および前（earlier）または前（preceding）（ステップ1710）に生成された基底ベクトルをテストして、基底ベクトル内の各要素の符号をスワップまたは反転することを除いて、ベクトルが実質的に類似したままであるかどうかを決定することを伴う。所与の基底ベクトル（ステップ1710の第1または先行ベクトルで始まる）について、合計が決定される。合計は、基底ベクトルの第1および第2セットからのそれぞれのベクトルの要素ごとの加算を最初に行った結果である。要素ごとの加算から得られたsimsの絶対値の合計が、類似性スコアを生成するために基底ベクトルごとに決定され、類似性スコアは、基底ベクトルごとに単一のスカラー値である。

【0197】

所与の基底ベクトルに対する類似性スコアの低い値は、基底ベクトルの第1セットにおける対応する基底ベクトルと実質的に類似しているが極性が反対であるセットの第2基底ベクトルを示す。したがって、類似性スコアの低い値は、分解の連続するパフォーマンス間の基底ベクトルにおける極性反転の指標である。合計が閾値を下回る場合、極性反転が検出され（ステップ1740において「YES」）、プロセッサ205における制御は、ステップ1740から基底ベクトル反転ステップ1750に進む。合計が閾値を下回らない場合、極性反転は検出されず（ステップ1740で「NO」）、プロセッサ205における制御は、ステップ1740から最後の基底ベクトルテストステップ1760に進む。類似性閾値は、典型的には予め決定される。いくつかの実施形態では、類似性閾値が現在のテンソルの幅および高さのうちの少なくとも1つ、例えば、幅および／または高さの0.8または0.9に線形的に関連する。他の構成では、類似性閾値がテンソルの1つのチャンネルのエリアの $\log 2$ に等しくてもよい。類似性閾値は、非反転ケースを表すフィルタリングされた類似性スコアなどの要因に基づいて決定され得る。中央値フィルタは、外れ値（例えば、極性反転を示す低スコア）を除外するために、類似性スコアに適用され得る。次いで、メジアンフィルタリングされた類似性スコアは、スケーリングされ、極性反転を検出するために所与のフレームの類似性スコアと比較され得る。フィルタリングは、基底ベクトルとレイヤとの間の異なる統計を考慮するために、基底ベクトルごとおよびレイヤごとの粒度に実行され得る。期待されるコンテンツの変化およびコンテンツの変動の結果としての類似性スコアの変動を許容することができる。閾値を設定する際の許容は、使用される分解方法などに基づいてもよい。

10

20

【0198】

基底ベクトル反転ステップ1750において、PCAエンコーダ160は、プロセッサ205の実行下で、現在処理されている基底ベクトルの極性を、例えば、ベクトルに-1（負の1）を乗算することによって反転し、更新された基底ベクトルを生成する。事実上、現在の基底ベクトルからの成分の更新されたセットが生成され、その結果、成分の更新されたセット内の成分は、ステップ1730で生成された基底ベクトル内のそれぞれの成分の逆である。この反転の結果として、現在の基底ベクトルは、基底ベクトルの第1セットからの対応する基底ベクトルに実質的に類似するようになる。いったん反転が行われると、次の分解動作でさらなる極性反転が行われない場合、現在の分解動作の反転された基底ベクトルとの整合極性を維持するために、基底ベクトルを後続の分解動作から反転し続ける必要がある。基底ベクトルごとに1つのフラグを有するフラグのセットは例えば、後続のフレームのための基底ベクトル反転ステップの持続性を提供するために、将来の基底ベクトル生成動作のための各基底ベクトルの反転状態を示すために使用される。現在の基底ベクトルのためのフラグは、極性反転が起こったことを示すためにフラグのセットにおいてトグルまたは変更され、極性反転が最初に検出されたときに反転を開始し、第2極性反転が検出されたときに反転を停止する場合の処理を可能にする。フラグのセットは、任意の後続テンソルの分解機能を使用して生成された成分が反転されるべきかどうかの基底を提供する。プロセッサ205における制御は、ステップ1750から最後の基底ベクトルテストステップ1760に進む。

30

【0199】

ステップ1760において、プロセッサ205は、処理されている現在のベクトルが基底ベクトルのセット内の最後のベクトルであるかどうかをテストする。そうでない場合（ステップ1760で「NO」）、処理は基底ベクトルのセット内の次の基底ベクトルに進み、プロセッサ205内の制御は、ステップ1760からステップ1740に進む。その場合（ステップ1760で「YES」）、処理はステップ1760から基底ベクトル符号化ステップ1770に進む。

40

【0200】

ステップ1770において、エンコーダ160は、プロセッサ205の実行下で、ステップ1750において反転が実行されたベクトルのための更新された基底ベクトルと、反転が実行されなかったベクトルのための基底ベクトルの第2セットからの基底ベクトルと

50

を含む最終基底ベクトルに対して動作する。基底ベクトルのこの最終セットは、方法1700の後続の呼び出しのための基準点として、すなわち「基底ベクトルの第1セット」として記憶される。基底ベクトルの最終セットは、基底ベクトル550として量子化器モジュール552に渡され、サブピクチャエンコーダ556によって符号化される。さらに、反転が1回実行される場合、各コンポーネントのフラグは方法1700の後続の呼び出しにおいて、極性が方法1700の前の呼び出しの極性と一致することを保証するために、それぞれのコンポーネントに対して反転が実行される必要があることを示す。反転が方法1700の後続の呼び出しで2回目に検出される場合、フラグはトグルされ、方法1700のさらなる呼び出しの際に、それぞれの成分の反転の停止をもたらす。次いで、方法1700は現在のフレームについて終了し、次のフレーム、または後のフレームについて再び呼び出され得、ここで、PCAエンコーダ160は、すべてのフレームよりも少ない頻度で実行される。

10

【0201】

図18Aは、アップサンプリング動作またはダウンサンプリング動作のいずれかを使用して異なるFPNレイヤからのテンソルを結合するためのモジュール162の例示的な実装形態1800を示す概略ブロック図である。図18Aの例は、CNNバックボーン部分400からのP2-P5レイヤを使用する。レイヤP6 429は、PCAデコーダ170が復号されたP5レイヤからP6レイヤを再構成することができるので、考慮から省略される。図18Aに示される例は、P2またはP3のうちの1つのリサンプリングに関するものであり、以下の図19を参照して説明される。

20

【0202】

図18Bは、アップサンプリング動作またはダウンサンプリング動作のいずれかを使用して、合成表現からテンソルを抽出して、FPNレイヤに対応する別個のテンソルに戻すためのモジュール172の実装1890を示す概略ブロック図である。図18BのTej例は、CNNヘッド1100のための例示的なP2-P5レイヤを提供する。レイヤP6 1118は、レイヤP5 1116に対して最大プーリング演算を実行することによって導出され、したがって、圧縮または解凍される必要はない。図18Bに示される例は、P2とP3とが組み合わされた図18Aに示される場合に関するものであり、以下の図20を参照して説明される。

【0203】

図19は、CNNのマルチレイヤ第1部分からの別のテンソルとの非相関のためにCNNのマルチレイヤ第1部分からテンソルを再サンプリングするための方法1900を示す。方法1900は、構成されたFPGA、ASIC、またはASSPなどの装置によって実施され得る。代替的に、以下で説明するように、方法1900は、プロセッサ205の実行下で、アプリケーションプログラム233の1つまたは複数のソフトウェアコードモジュールとして、ソースデバイス110によって実施され得る。方法1900は、ビットストリーム143において符号化されたビデオデータの各フレームについて繰り返される。方法1900を実施するアプリケーションプログラム233のソフトウェアコードモジュールは例えば、ハードディスクドライブ210および／またはメモリ206に記憶され得る。方法1900は、テンソルマッピング判断ステップ1910で開始する。

30

40

【0204】

ステップ1910において、テンソルコンバイナ162は、プロセッサ205の実行下で、FPNの2つの異なるレイヤからの（テンソル115aの）テンソルが結合されるかどうかを判断する。組合せは、典型的には、テンソル115aが、第1サイズを有する特徴マップの第1セットと、第1サイズとは異なる第2サイズを有する特徴マップの第2セットと、を少なくとも有する場合に生じる。結合されない場合、テンソル115aは、変更されずに通過される。テンソルが結合される場合、テンソルコンバイナ162は、より大きいテンソルのダウンサンプリング、またはより小さいテンソルのアップサンプリングを実行することによってテンソルが結合されるか、または再サンプリングなしにテンソルが結合されるか、を判断する。アップサンプリングおよびダウンサンプリングの選択肢は

50

テンソル 1 1 5 a が少なくとも 2 つのテンソルを含むときに利用可能であり、2 つのテンソルの幅および高さは 2 倍の関係にあり、すなわち、含まれる 2 つの動作に等しいストライドを有する層状ネットワークから生成される。再サンプリングなしで連結するオプションはテンソル 1 1 5 a が少なくとも 2 つのテンソルを含み、2 つのテンソルが同じ幅および高さを有する場合に利用可能である。テンソル 1 1 5 a の次元 (dimensionality) はネットワークアーキテクチャおよび分割点によって設定されるので、テンソルコンバイナ 1 6 2 に利用可能なオプションはネットワークアーキテクチャおよび分割点の選択によって制約される。テンソル 1 1 5 a が 1 つのテンソルのみを含む場合、テンソルを結合するための範囲はなく、テンソルコンバイナ 1 6 2 の唯一の選択肢は、テンソル 1 1 5 a をテンソル 1 1 5 として変更せずに通過させることである。同じ幅であるが高さの半分または 2 倍のテンソルへのダウンサンプリングまたはアップサンプリングが評価される場合、`resample__vertical__only__flag` シンタックス要素は、垂直方向のみにおけるリサンプリングの適用を示す。同じ高さであるが幅の半分または 2 倍のテンソルへのダウンサンプリングまたはアップサンプリングが評価される場合、`resample__horizontal__only__flag` シンタックス要素は、水平方向のみにおけるリサンプリングの適用を示す。垂直にのみ、または水平にのみ再サンプリングするためのオプションはいくつかの CNN アーキテクチャで遭遇し得るように、2 に等しいストライドが垂直または水平にのみ適用される、異なるレイヤにおけるテンソル結合を可能にする。

10

20

30

40

50

【0205】

より大きなテンソルがダウンサンプリングされる場合、より大きなテンソルにおける詳細は失われる。したがって、ダウンサンプリングは、主に大きなオブジェクトを含むフレームデータ 1 1 3 などの、詳細を欠くデータにより適している。フレームデータ 1 1 3 がより小さい詳細を含むと判定された場合、プロセッサ 2 0 5 は、より大きいテンソルをダウンサンプリングすることを選択しない。ダウンサンプリングかアップサンプリングにかかわらず、クロスチャネル分解に基づいて圧縮を適用する前に 2 つの異なるレイヤからテンソルを組み合わせた結果は、レイヤにわたって冗長性を発見する能力である。各レイヤが PCA エンコーダ 1 6 0 によって独立して符号化される場合、各テンソルのチャンネルにわたる冗長性は発見されるが、レイヤにわたる冗長性は発見することができない。レイヤにわたる冗長性が識別される場合、2 つの異なる FPN レイヤを結合するテンソルを符号化するために必要とされる基底ベクトルの総数は各 FPN レイヤのテンソルに割り振られる基底ベクトルの数よりも低くなり得、ここで、テンソルは単一のレイヤに結合されない。より小さいテンソルがアップサンプリングされる場合、レイヤ間冗長性を利用するという点での利点が見られ、その結果、結合テンソルの基底ベクトルの数を増加させる必要がほとんどまたは全くない。2 つのレイヤ間で幅および高さが同じである (チャンネルカウントが異なり得る) 階層化ネットワークの場合、テンソルコンバイナ 1 6 2 は、同じ幅および高さを有する 2 つのレイヤからテンソルを再サンプリングすることなく、クロスレイヤ冗長性を発見することを試みることができる。プロセッサ 2 0 5 は再サンプリング、アップサンプリング、またはダウンサンプリングをせずに結合し、テンソルを修正せずに残すオプションを試み、結果として得られるテンソル 1 1 5 の符号化コストが最も低くなるオプションを選択し得る。他の構成では、ステップ 1 9 1 0 が再サンプリングが実施されるかどうかの所定のデフォルト動作を設定することができる。ステップ 1 9 1 0 で決定されたマッピングは所定のマッピングであってもよいし、または、フレーム毎など、より定期的に決定されてもよい。マッピングが予め決められている場合、マッピングは、ビットストリームの開始時にのみシグナリングされる必要がある。SEI メッセージ 1 3 1 3 中の `layer__update__mapping__info` シンタックス要素は、マッピングが宛先デバイス 1 4 0 に通信されるべき SEI メッセージ 1 3 1 3 のシグナリングインスタンスを許可する。所定のマッピングの場合、CLVS の第 1 ピクチャ 1 2 0 0 のマッピングの 1 回のシグナリングインスタンスで十分である。プロセッサ 2 0 5 における制御は、ステップ 1 9 2 0 からテンソルマッピング符号化ステップ 1 9 2 0 に進む。

【0206】

ステップ1920において、テンソルコンバイナ162は、プロセッサ205の実行下で、ステップ1920の動作から得られた判定をビットストリーム121に符号化する。符号化された判定は、テンソルコンバイナ162において実行される動作を逆転させるためにテンソルセパレータ172において実行される必要がある動作を表す。F P Nレイヤをアップサンプリングする、ダウンサンプリングする、または変更されないままにする決定は、フラグまたは可変長コードワードのいずれかを使用してS E Iメッセージ1313中で符号化される。ステップ1910で行われた判定をビットストリーム121に符号化するためのシンタックスは付録Aを参照して説明される。シンタックスがテンソル149がテンソル115aと同じ数、次元、およびチャンネルカウントを有するように、すなわち、テンソルコンバイナ162の動作の前に、テンソルセパレータ172において実行されるべき動作を符号化する。テンソルコンバイナ162によってテンソルがアップサンプリングされ、別のテンソルと結合された場合、シグナリングは、テンソルセパレータ172が復号された結合テンソルから2つのテンソルを抽出することを示す。抽出されたテンソルのうちの1つがテンソル149のうちの1つとして出力され、抽出されたテンソルがダウンサンプリングされて、テンソル149に含まれる追加のテンソルが生成される。テンソルコンバイナ162によってテンソルがダウンサンプリングされ、別のテンソルと結合された場合、シグナリングは、テンソルセパレータ172が復号された結合テンソルから2つのテンソルを抽出することを示す。抽出されたテンソルのうちの一方はテンソル149のうちの一方として出力され、他方はアップサンプリングされて、テンソル149に含まれる追加のテンソルを生成する。テンソルコンバイナ162によってテンソルが再サンプリングされなかったが、別のテンソルと結合された場合、シグナリングはテンソルセパレータ172が復号された結合テンソルから2つのテンソルを抽出することを示し、その両方がテンソル149のうちのテンソルとして出力される。結合テンソルを取得するためのインデックス、すなわちsrc_layer_idx、および再サンプリングされたテンソルを挿入するためのインデックス、すなわちdst_layer_idx。ソースおよび宛先レイヤインデックスのシグナリングはたとえば、連結されたレイヤがテンソル115a間で隣接していない、または別のレイヤの半分の空間解像度を持つレイヤがテンソル115a間で先に出現する、またはテンソル115a間で後に出現するケースをサポートするために、柔軟である。アップサンプリングが実装される符号化は、符号化時にアップサンプリングされた特徴マップが復号時にダウンサンプリングされるべきであり、逆もまた同様であることを指定する。プロセッサ205における制御は、ステップ1920からテンソルエンコーダマッピングテストステップ1930に進む。

【0207】

ステップ1930において、ステップ1910で行われた判定が、テンソル115aの異なるF P Nレイヤからのテンソルを結合しないことである場合、方法1900は終了する。プロセッサ205における制御は方法1400の平均特徴決定ステップ1420に戻り、ここで、すべてのテンソル115aが、残りのステップのためのテンソル115として渡される。ステップ1910で行われる判定がテンソル115aのF P Nレイヤからのテンソルのテンソルコンバイナ162におけるアップサンプリングを実行することである場合、プロセッサ205における制御は、ステップ1930から選択テンソルアップサンプルステップ1940に進む。ステップ1910で行われる判定がテンソル115aのF P Nレイヤからのテンソルのテンソルコンバイナ162におけるダウンサンプリングを実行することである場合、プロセッサ205における制御は、ステップ1930から選択テンソルダウンサンプルステップ1950に進む。ステップ1910で行われた判定がリサンプリングフィルタを適用せずにテンソル115aのうちのテンソルのテンソルコンバイナ162において連結を実行することである場合、プロセッサ205における制御は、ステップ1958から進む。

【0208】

図18Aを参照すると、実装1800は、入力P3～P5（それぞれ1810～181

10

20

30

40

50

3)を受信する。実装1800は、テンソルアップサンプラ1822と、テンソルダウンサンプラ1831と、連結モジュール1818および1836と、マルチプレクサ1844を含む。

【0209】

アップサンプリングは、特徴マップ（またはテンソル）の第2セットの第2サイズが第1のサイズより大きい、またはその逆である場合に、第1テンソルに対して行われる。ステップ1940において、テンソルアップサンプラ1822はプロセッサ205の実行下で、テンソル、例えば、P3テンソル1811をテンソル115aからアップサンプリングし、アップサンプリングされたP3テンソル1824を生成する。アップサンプリングされたテンソル1824は、アップサンプリングされた特徴マップのセット1824と呼ばれることもある。P3テンソル1811は、幅wおよび高さhを有する。アップサンプリングされたP3テンソル1824は、幅2dおよび高さ2hを有する。アップサンプリングは、アップサンプリングされたP3テンソル1824に必要な中間サンプルを生成するために、補間フィルタ、最近傍フィルタ、または他のフィルタを使用して実行され得る。プロセッサ205における制御は、ステップ1940からテンソル連結ステップ1945に進む。

10

【0210】

ステップ1945において、連結モジュール1818は、プロセッサ205の実行下で、アップサンプリングされたP3テンソル1824とP2テンソル1810とをチャンネル次元に沿って連結して、連結テンソル1842を生成する。連結テンソル1842は、幅2wおよび高さ2hと、テンソル1824および1810のチャンネルカウンットの合計に等しいチャンネルカウンットとを有する。マルチプレクサ1844は、テンソル1842を連結出力テンソル1846として選択する。テンソル1846、1812および1813は、残りの符号化ステップのためのテンソル115として出力される。次いで、方法1900は終了し、プロセッサ205における制御は、方法1400のステップ1420において再開する。

20

【0211】

ステップ1950において、テンソルダウンサンプラ1831は、プロセッサ205の実行下で、テンソル、例えば、P2テンソル1810をテンソル115aからダウンサンプリングして、ダウンサンプリングされたP2テンソル1832を生成する。P2テンソル1810は、幅2wおよび高さ2hを有する。ダウンサンプリングされたP2テンソル1832は、P2テンソル1810の幅の半分および高さの半분을有するように空間的にダウンサンプリングされ、寸法w、hをもたらす。ダウンサンプリングは、サブサンプリングされたバイリニア、バイキュービック、または他のフィルタリング動作を使用して実行され得る。ダウンサンプリングまたは他のフィルタリングは、P2テンソル1810のチャンネル次元に適用されないため、ダウンサンプリングされたP2テンソル1832のチャンネルカウンットは変更されない。プロセッサ205における制御は、ステップ1950からテンソル連結ステップ1955に進む。

30

【0212】

ステップ1955において、連結モジュール1836は、プロセッサ205の実行下で、チャンネル次元に沿ってP3テンソル1811とダウンサンプリングされたP2テンソル1832とを連結して、連結テンソル1839を生成する。連結テンソル1839は、幅wおよび高さhと、テンソル1811および1832のチャンネルカウンットの合計に等しいチャンネルカウンットとを有する。マルチプレクサ1844は、テンソル1839を、連結出力テンソル1846として選択する。テンソル1846、1812および1813は、残りの符号化ステップのためのテンソル115として出力される。再サンプリングが実施された場合、出力テンソル1846はP2およびP3および連結器1818および1836の動作に関して図18Aに記載されているように、特徴マップの再サンプリングされたセットと、再サンプリングされていない特徴マップのセットと、を含む。次いで、方法1900は終了し、プロセッサ205における制御は、方法1400のステップ1420にお

40

50

いて再開する。

【0213】

ステップ1958において、連結モジュール1823は、プロセッサ205の実行下で、テンソル1811とテンソル1832（この場合、両方とも同じ幅および高さを有する）とを連結して、連結テンソル1838を生成する。マルチプレクサ1844は、連結出力テンソル1846としてテンソル1838を通す。テンソル1846、1812、および1813は、残りの符号化ステップのためのテンソル115として渡される。次いで、方法1900は終了し、プロセッサ205における制御は、方法1400のステップ1420において再開する。

【0214】

方法1400は、ステップ1420～14120を実行し続け、例えば、ステップ1480においてステップ2070で出力されたテンソルを使用して基底ベクトルを決定し、ステップ14110において基底ベクトルを使用して連結テンソル内の両方の特徴マップに関連する係数を導出し、ステップ1490および14120において係数および基底ベクトルを符号化する。

【0215】

方法1900の動作の結果として、連結テンソルは、FPNの2つの別個のレイヤを合成するテンソル149のうちの1つとして渡される。連結テンソルは、PCAなどの分解法を用いて圧縮される。分解方法はまた、アップサンプリング／ダウンサンプリングが発生したFPNの2つのレイヤに沿って非相関化することができ、非相関化がFPNの各レイヤに限定されるのではない。さらに、詳細が保存される必要がある場合、ソースデバイス110は、基底ベクトル導出および圧縮の目的のために、より小さいFPNレイヤ、たとえば、P3からテンソルをアップサンプリングするオプションを有する。アップサンプリングされたP3レイヤとP2レイヤとの連結のために生成された基底ベクトルの数は、P2レイヤのために純粋に生成された基底ベクトルの数と比較して増加される必要はない。したがって、P3レイヤのためのピクチャ1200中の基底ベクトルの追加のエリアは必要とされない。

【0216】

図20は、CNNヘッド150に提供するための出力テンソル149を生成するために、復号された結合表現、すなわちテンソル149aからテンソルを再サンプリングするための方法2000を示す。方法2000は、符号化データからテンソルを復号する際に使用される。方法2000は、構成されたFPGA、ASIC、またはASSPなどの装置によって実施され得る。代替的に、以下で説明するように、方法2000は、プロセッサ205の実行下で、アプリケーションプログラム233の1つまたは複数のソフトウェアコードモジュールとして、宛先デバイス130によって実施され得る。方法2000は、ビットストリーム143において符号化されたビデオデータの各フレームについて繰り返される。方法2000を実施するアプリケーションプログラム233のソフトウェアコードモジュールは例えば、ハードディスクドライブ210および／またはメモリ206に記憶され得る。方法2000は、テンソルマッピング復号ステップ2010で開始する。

【0217】

ステップ2010において、エントロピーデコーダ920は、プロセッサ205の実行下で、テンソル149aが結合されたりサンプリングされたテンソルを含むかどうかを示すビットストリーム143に含まれるSEIメッセージ1313からシンタックス要素を復号する。デコーダはリサンプリングを実行するか否かを、例えば、符号化時にリサンプリングが発生したか否かを特定する復号情報に基づいて、又は、デフォルト設定に基づいて、決定する。すなわち、エントロピー復号器は図19の符号化で実装されるように、テンソルのうちの1つが、別のFPNレイヤからのテンソルの再サンプリング（アップサンプリングまたはダウンサンプリング）されたバージョンを有する1つのFPNレイヤからのテンソルの連結であるかどうかを復号する。方法2000は、ステップ1540および1550における基底ベクトルおよび係数の復号がすでに行われ、ステップ1570にお

10

20

30

40

50

いてテンソルが導出されたときに実行される。結合テンソルは、同じ最終幅および高さを有するテンソルの連結であってもよく、またはテンソル 1 4 9 a に存在する連結テンソルがなくてもよい。

【0218】

SEIメッセージ 1 3 1 3 におけるシグナリングは、テンソル 1 4 9 を復元するために、すなわち、CNNヘッド 1 5 0 による使用に適したテンソル 1 1 5 a と同じ数および次元を有するテンソルのセットを生成するために、テンソルセパレータ 1 7 2 によって実行される必要がある動作を示す。テンソルコンバイナ 1 6 2 が、連結されたテンソルを生成するためにテンソルをダウンサンプリングしている場合、ビットストリーム 1 4 3 におけるシグナリングはテンソルを復元するために、連結されたテンソルの一部をアップサンプリングするようにテンソルセパレータ 1 7 2 に指示する。テンソルコンバイナ 1 6 2 が、連結されたテンソルを生成するためにテンソルをアップサンプリングしている場合、ビットストリーム 1 4 3 におけるシグナリングはテンソルを復元するために、連結されたテンソルの一部をダウンサンプリングするようにテンソルセパレータ 1 7 2 に指示する。テンソルコンバイナ 1 6 2 が連結されたテンソルを生成するためにテンソルを再サンプリングしていない場合、ビットストリーム 1 4 3 におけるシグナリングは、テンソルセパレータ 1 7 2 に、再サンプリングフィルタを適用せずに、連結されたテンソルの一部を抽出してテンソルを回復するように指示する。連結されたテンソルからのテンソルの抽出は、復号された `layer__extraction__flag` シンタックス要素から決定される。抽出されたテンソルへのダウンサンプリングフィルタの適用は、`layer__downsample__flag` シンタックス要素を介してシグナリングされる。抽出されたテンソルへのアップサンプリングフィルタの適用は、`layer__upsample__flag` を介してシグナリングされる。テンソル 1 4 9 a のうちのどのテンソルが連結テンソルであるかは、`src__layer__idx` シンタックス要素を介してシグナリングされる。テンソル 1 4 9 を生成するために抽出されたテンソルが挿入されるオフセットは、`dst__layer__idx` シンタックス要素を介してシグナリングされる。2つのレイヤ間の連結テンソル内の境界を描写するチャンネルインデックスは、`src__channel__offset` シンタックス要素を介してシグナリングされる。プロセッサ 2 0 5 における制御は、ステップ 2 0 1 0 からテンソルデコーダマッピングテストステップ 2 0 2 0 に進む。

【0219】

ステップ 2 0 2 0 において、ステップ 2 0 1 0 の結果が再マッピングが適用されないことを示す場合、プロセッサ 2 0 5 における制御は、ステップ 2 0 1 0 からテンソル出力ステップ 2 0 7 0 に進む。ステップ 2 0 1 0 の結果がテンソルダウンサンプリングが適用されるべきであることを示す場合、プロセッサ 2 0 5 における制御は、ステップ 2 0 2 0 からテンソル抽出ステップ 2 0 3 0 に進む。ステップ 2 0 1 0 の結果がテンソルアップサンプリングが適用されるべきであることを示す場合、プロセッサ 2 0 5 における制御は、ステップ 2 0 2 0 からテンソル抽出ステップ 2 0 5 0 に進む。ステップ 2 0 1 0 の結果が再サンプリングなしのテンソル抽出が適用されることを示す場合、プロセッサ 2 0 5 における制御は、ステップ 2 0 2 0 からテンソル抽出ステップ 2 0 3 5 に進む。

【0220】

図 1 8 B の実装 1 8 9 0 は、入力 P 2 / 3、P 4 および P 5（それぞれ 1 8 5 0、1 8 5 2 および 1 8 5 4）を受信する。入力 P 2 / 3 は例えば、方法 1 9 0 0 の動作によって、符号化側でのアップサンプリングまたはダウンサンプリングによって生成される結合テンソルである。実装 1 8 9 0 は、テンソルスライサ 1 8 5 6、ダウンサンプラ 1 8 6 2、アップサンプラ 1 8 6 4 及びセクタ 1 8 7 0 および 1 8 7 2 を含む。

【0221】

ステップ 2 0 3 0 において、テンソルスライサ 1 8 5 6 は、プロセッサ 2 0 5 の実行下で、(`src__channel__offset` に従うインデックスにおいて) チャンネル次元に沿って、結合された P 2 / P 3 テンソル 1 8 5 0 を 2 つのテンソルにスライスし、各

々はP2レイヤおよびP3レイヤのチャンネルカウントに対応するチャンネルカウント、すなわち、256個のチャンネルを有する。テンソルスライサ1856は、第1のスライスされたテンソル1858および第2のスライスされたテンソル1860を出力する。プロセッサ205における制御は、ステップ2030から抽出テンソルダウンサンプルステップ2040に進む。

【0222】

ステップ2035において、テンソルスライサ1856は、プロセッサ205の実行下で、チャンネル次元に沿って(src_channel_offsetに従うインデックスで結合P2/P3テンソル1850を2つのテンソルにスライスする。テンソルスライサ1856は、第1のスライスされたテンソル1858をセクタ1870に出力し、第2のスライスされたテンソル1860をセクタ1872に出力する。プロセッサ205における制御は、ステップ2035からテンソル出力ステップ2070に進む。

10

【0223】

ステップ2040において、ダウンサンプラ1862は、プロセッサ205の実行下で、テンソル1858をダウンサンプリングして、再スケールされたテンソル1866を導出する。再スケーリングされたテンソルは、結合テンソルを復号するために使用される特徴マップのセットを提供し、ダウンサンプラ1862は、デシメーション、またはサブサンプリングされたバイリニアもしくはバイキュービックもしくは他のフィルタを使用して、再スケーリングされたテンソル1866を生成し得る。プロセッサ205における制御は、ステップ2040からテンソル出力ステップ2070に進む。

20

【0224】

ステップ2050において、テンソルスライサ1856は、プロセッサ205の実行下で、チャンネル次元に沿って結合されたP2/P3テンソル1850を2つの半分にスライスし、各々はP2レイヤおよびP3レイヤのチャンネルカウントに対応するチャンネルカウント、すなわち、256個のチャンネルを有する。テンソルスライサ1856は、第1のスライスされたテンソル1858及び第2のスライスされたテンソル1860を出力する。プロセッサ205における制御は、ステップ2050から抽出テンソルアップサンプルステップ2060に進む。

【0225】

ステップ2060において、アップサンプラ1864は、プロセッサ205の実行下で、テンソル1860をアップサンプリングして、再スケールされたテンソル1868を生成する。アップサンプリングは再スケールされたテンソル1868をポピュレートするために必要とされる補間されたサンプルを導出するために、バイリニア、バイキュービック、最近傍、または他のフィルタ方法を使用し得る。プロセッサ205における制御は、ステップ2060からテンソル出力ステップ2070に進む。

30

【0226】

ステップ2040および2060は、ステップ2030または2050で導出されたテンソルの第1部分を再サンプリング(それぞれダウンサンプリングおよびアップサンプリング)して、テンソル149aを復号することによって、特徴マップの第1セットを導出するように動作する。

40

【0227】

ステップ2070において、出力テンソル149が、CNNヘッド150による使用のために提供される。P5テンソル1854およびP4テンソル1852は、テンソル149aからテンソル149に渡される。再マッピングまたは再サンプリング動作が行われなかった場合、P2レイヤおよびP3レイヤに対応するテンソルは、テンソル149aからテンソル149に渡される。再マッピング動作が行われた場合、P3テンソル1882およびP2テンソル1880が、テンソル149の対応するFPNレイヤとして出力される。ダウンサンプリングが行われた場合、セクタ1870はダウンサンプリングされたテンソル1866をテンソル1882として出力し、セクタ1872は、テンソル1858をテンソル1880として出力する。アップサンプリングが行われた場合、セクタ1

50

870はテンソル1860をテンソル1882として出力し、セクタ1872は、テンソル1868をテンソル1880として出力する。リサンプリングが実施された場合、出力テンソル1880および1882のうちの1つは、P2およびP3およびセクタ1870および1872の動作に関して図18Bに記載されているように、特徴マップのリサンプリングされたセットと、リサンプリングされていない特徴マップのセットと、を含む。再サンプリングが行われなかった場合、セクタ1870はテンソル1860をテンソル1882として出力し、セクタ1872は、テンソル1858をテンソル1880として出力する。方法2000は終了し、プロセッサ205における制御は方法1500に戻る。

【0228】

方法1900および2000の構成では、ステップ1910はまた、P2～P5のうちのどのレイヤがアップサンプリングまたはダウンサンプリングされるべきかを決定し、隣接するレイヤの対、すなわち、P2およびP3、P3およびP4、またはP4およびP5を選択する。ステップ1920において、隣接レイヤの選択されたペアがビットストリーム121に符号化される。ステップ2010において、隣接レイヤの選択されたペアがビットストリーム143から復号され、示されたテンソルが連結テンソルから復元される。

【0229】

図21は、CNNバックボーン114からテンソル115を符号化するために分解モジュール548によって基底ベクトルを決定するための方法2100を示す。方法2100は、構成されたFPGA、ASIC、またはASSPなどの装置によって実施され得る。方法2100は、付録Aに存在する基底ベクトルパッキングフォーマット情報SEIメッセージシンタックスを参照して説明される。あるいは以下で説明されるように、方法2100は、プロセッサ205の実行下で、アプリケーションプログラム233の1つまたは複数のソフトウェアコードモジュールとして、ソースデバイス110によって実施され得る。方法2100は、ビットストリーム121において符号化されたビデオデータの各フレームに対して繰り返される。方法2100を実施するアプリケーションプログラム233のソフトウェアコードモジュールは例えば、ハードディスクドライブ210および／またはメモリ206に記憶され得る。方法2100は、最大基底ベクトルカウント符号化ステップ2110で開始する。

【0230】

ステップ2110において、エンロピーエンコーダ738は、プロセッサ205の実行下で、値**bv__max__cnt**をSEIメッセージ1313に符号化する。値**bv__max__cnt**は、分解モジュール548によってビットストリーム121に生成されるべき基底ベクトルの数に対応する基底ベクトルパッキング情報1391の一部としてSEIメッセージ1313に符号化される。値**bv__max__cnt**はテンソル115のうち、テンソルの分解に用いることができる基底ベクトルの最大数を示す。テンソルコンバイナ162においてレイヤの減少が生じなかった場合、**bv__max__cnt**の1つのインスタンスが、テンソル115の各FPNレイヤに対して符号化される。テンソルコンバイナ162の結果としてレイヤの数が低減された場合、1つの**bv__max__cnt**値が低減されたフォーマットで得られたレイヤの各々について、すなわち、符号化されたレイヤごとに、符号化される。ステップ2110は、初期化ステップとして実行される必要があるだけであり、したがって、1つまたは複数の**bv__max__cnt**値は、ビットストリーム121内の初期化またはランダムアクセスポイントにおいて符号化される必要があるだけである。さらに、ステップ2110において、各レイヤの基底ベクトルの幅および高さが、SEIメッセージ1313において**bv__width**および**bv__height**として符号化される。フレームサイズは、ステップ2110で復号された基底ベクトルの最大数を使用して、符号化データについて効果的に決定され、フレームサイズは符号化データの少なくとも1つのフレームに適用可能である。プロセッサ205における制御は、ステップ2110から基底ベクトルパッキングフォーマット決定ステップ2120に進む。

【0231】

ステップ 2 1 2 0 において、サブピクチャエンコーダ 5 5 6 のためのパッカ 6 1 0 は、サブピクチャ 1 2 1 6 のためのパッキングフォーマットと、サブピクチャ 1 2 1 6 のためのサイズとを決定する。パッキングフォーマットおよびサイズはたとえば、図 1 2 B を参照して示されるように、テンソル 1 1 5 内の各テンソルについて指定されたサイズおよびカウント bv_max_cnt の基底ベクトルがサブピクチャ 1 2 1 6 にパッキングされることが可能であるように決定される。パッキングされた基底ベクトルを含む後続のピクチャ（すなわち、1 2 0 0）は、所与のピクチャに対して使用される基底ベクトルがより少なくとも、方法 2 1 0 0 の最初の呼び出しで決定されたものと同じサイズを有する。したがって、サブピクチャエンコーダ（すなわち、5 1 4、5 3 2、5 5 6、および 5 7 2）は同じサイズのフレームを使用して圧縮テンソル 1 1 5 の連続セットを符号化し、ピクチャ解像度の変化をシグナリングするために S P S 1 3 1 0 の新しいインスタンスを送る必要はない。ピクチャ解像度を変更することは、フレームバッファ 7 7 2 のリセットを必要とし、それは使用されるべきインター予測方法のために必要に応じて、後続のピクチャがフレームバッファ 7 7 2 内の以前のピクチャを参照することを防ぐ。最大基底ベクトルカウントに基づいてピクチャ 1 2 0 0 のサイズおよびパッキングフォーマットを決定することは、フレームバッファ 7 7 2 が互換性のない解像度のピクチャを収容するためにリセットされる必要がないので、利用される基底ベクトルの数が変更するときにインター予測が実行されることを可能にする。したがって、パッキングされる基底ベクトルの数が変化する場合であっても、ピクチャ 1 2 0 0 の固定サイズを維持することは、結果として得られるパッキングされたピクチャにわたってインター予測を使用することができるので、圧縮効率を改善する。プロセッサ 2 0 5 における制御は、ステップ 2 1 2 0 から基底ベクトルセット決定ステップ 2 1 3 0 に進む。

【0 2 3 2】

ステップ 2 1 3 0 において、分解モジュール 5 4 8 は、プロセッサ 2 0 5 の実行下で、テンソル 1 1 5 の各テンソルの M 個の基底ベクトルのセットを生成し、ここで、M は、テンソル 1 1 5 のそれぞれのテンソルの値 bv_max_cnt （' M '）に対応する。値 M は、P 3 および P 4 などの 2 5 6 個のチャンネルを含むレイヤに対して固定値 2 5 に設定され得る。P 2 および P 3 などの 2 つのレイヤがコンバイナ 1 6 2 において連結された場合、値 M はテンソル 1 1 5 a のレイヤごとに 2 5 個の基底ベクトルを維持することに対応して、すなわち、テンソルコンバイナ 1 6 2 からの任意のレイヤ低減の前に、5 0 に設定され得る。テンソルの M 個の基底ベクトルは主成分の順序付けられたセットであり、セットの各々は、M 個の基底ベクトルの先行する主成分によってまだ捕捉されていないテンソルの最大分散量を捕捉する。分散は、テンソルのチャンネル次元にわたって「捕捉」または説明される。したがって、M 個の基底ベクトルは、テンソルの各チャンネルがテンソルの所与のチャンネルにおけるサンプルの数（すなわち、幅 x 高さ）よりも少ない値で表現され得る空間を定義する。各基底ベクトルによって説明される分散の量は固有値である。M 個の基底ベクトルは M 個の固有値の対応するセットを有し、固有値は、厳密に減少する数の配列を形成する。プロセッサ 2 0 5 における制御は、ステップ 2 1 3 0 から固有値累積ステップ 2 1 4 0 に進む。

【0 2 3 3】

ステップ 2 1 4 0 において、分解モジュール 5 4 8 は、プロセッサ 2 0 5 の実行下で、各値 n について累積された「説明された分散」値を生成し、ここで、n は 1 から M までの範囲である。累積された説明された分散値は、第 1 固有値から第 n 固有値までの固有値の合計に等しい。累積された説明された分散値は、ステップ 2 1 3 0 で生成された固有ベクトルまたは主成分の総セットのサブセットによってどの程度の分散が説明されるかを示す。プロセッサ 2 0 5 における制御は、ステップ 2 1 4 0 から基底ベクトルサブセット決定ステップ 2 1 5 0 に進む。

【0 2 3 4】

ステップ 2 1 5 0 において、分解モジュール 5 4 8 は、プロセッサ 2 0 5 の実行下で、「説明された分散閾値」に対応する基底ベクトル M のサブセットを選択する。説明された

分散閾値は、使用される基底ベクトルの数を決定し、精度、エリア、および符号化効率の間の所望のトレードオフに従ってテンソルを適切に分解するように選択され得る。説明された分散閾値は、基底ベクトル自体におけるデータに基づいて設定されるか、または所望のエリアもしくは符号化効率、または最小の所望の精度に基づいて設定されることができ。基底ベクトルのサブセットの選択は、M個の基底ベクトルの最初のN個の基底ベクトルを選択すること、すなわち、利用可能な（最大の）基底ベクトルの中から基底ベクトルを選択することをもたす。説明された分散閾値は、フレームごとに、すなわち、符号化されるデータのフレームごとに決定することができる。サブセットの選択は、累積された説明された分散に対するそれらのより小さい寄与によって、テンソルの再構成にあまり寄与しない基底ベクトルを省略する。サブセットの選択は、サブピクチャ1216がより少ない占有エリアを有するので、圧縮効率を改善する。サブピクチャ1216の非占有エリアは、一定のサンプル値によって占有され得る。一定のサンプル値は、VVCの柔軟なブロック構造のために圧縮性が高い。プロセッサ205における制御は、ステップ2150から基底ベクトルサブセット符号化ステップ2160に進む。

10

20

30

40

50

【0235】

ステップ2160において、エントロピーエンコーダ738は、プロセッサ205の実行下で、テンソル115の各テンソルの値b v __ c n tをビットストリーム121に符号化する。値b v __ c n tは、ステップ2150の実行から得られる選択されたサブセットを示す。値b v __ c n tは、基底ベクトル0～N、すなわち、使用される基底ベクトルの数が、所与のテンソルに対して1からMまで変化し得ることを示し得る。ステップ2160は、テンソルを符号化するために使用される基底ベクトルの数を符号化する。使用されるべき基底ベクトルの数を指定する情報は、ステップ2110で決定されるように使用されるべき基底ベクトルの最大個数と、ステップ2150で決定されるように使用されるべき基底ベクトルの実際の数とを含むと考えることができる。方法2100は終了し、プロセッサ205における制御は、ステップ2160からステップ1490に進み、ここで、ステップ1490で基底ベクトルのサブセットがビットストリーム内で符号化される。

【0236】

図22は、CNNの第1部分からテンソルを符号化するために使用される基底ベクトルを復号するための方法2200を示す。方法2200は、PCSエンコーダ160によって符号化されたデータからテンソルを復号する際に使用される。方法2200は、構成されたFPGA、ASIC、またはASSPなどの装置によって実施され得る。方法2100は、付録Aに存在する基底ベクトルパッキングフォーマット情報SEIメッセージシンタックスを参照して説明される。あるいは以下で説明されるように、方法2200は、プロセッサ205の実行下で、アプリケーションプログラム233の1つまたは複数のソフトウェアコードモジュールとして、宛先デバイス140によって実施され得る。方法2200は、ビットストリーム143において符号化されたビデオデータの各フレームに対して繰り返される。方法2200を実施するアプリケーションプログラム233のソフトウェアコードモジュールは例えば、ハードディスクドライブ210および／またはメモリ206に記憶され得る。方法2200は、最大基底ベクトルカウント復号ステップ2210で開始する。

【0237】

ステップ2210において、エントロピーデコーダ920は、プロセッサ205の実行下で、SEIメッセージ1313からb v __ m a x __ c n tシンタックス要素を復号する。b v __ m a x __ c n tシンタックス要素は、ピクチャ1200をパッキングする際に使用されることが可能な基底ベクトルの最大数を指定または示す情報を提供する。別個の最大値は、テンソルコンバイナ162から得られる結合表現内の各テンソルについて復号され得る。ステップ2210において、各結合表現の基底ベクトルの幅および高さもSEIメッセージ1313から復号される。b v __ m a x __ c n tシンタックス要素は、b v __ m a x __ c n tの別のインスタンスがビットストリーム143から復号されることが示されるまで、ピクチャのシーケンスの第1ピクチャおよび後続のピクチャによる使用のため

に保持される値のために復号される。b v _ m a x _ c n t の別のインスタンスは、たとえば b a s i s _ v e c t o r _ d i m e n s i o n s _ u p d a t e フラグを介して示され得る。プロセッサ 205 における制御は、ステップ 2210 から基底ベクトルパッキングフォーマット決定ステップ 2220 に進む。

【0238】

ステップ 2220 において、アンパッカ 822 は、プロセッサ 205 の実行下で、ステップ 2210 において決定された基底ベクトル幅、高さ、および最大個数に従って、サブピクチャ 1216 においてパッキングされたときの各基底ベクトルの空間位置を決定する。基底ベクトルは一般に、まず左から右の方法でパッキングされ、次に、上から下の方法でパッキングされ、基底ベクトルの順序付けられたリストを通して進む。基底ベクトルは一般に、隣接して、重複しない方法でパッキングされる。複数のレイヤが存在する場合、たとえば、FPN の使用のため、それぞれのレイヤの基底ベクトルの空間位置がサブピクチャ 1216 について決定される。ステップ 2220 は、b v _ m a x _ c n t がビットストリーム 143 から復号されるとき、すなわち、b a s i s _ v e c t o r _ d i m e n s i o n s _ u p d a t e フラグによって示されるときはいつでも実行される必要がある。b v _ m a x _ c n t がビットストリーム 143 から復号されないとき、方法 2200 のより早い呼び出しにおいてステップ 2220 を実行することから決定されたパッキング構成が再使用され得る。プロセッサ 205 における制御は、ステップ 2220 から基底ベクトルサブセット復号ステップ 2230 に進む。

【0239】

ステップ 2230 において、エントローピーデコーダ 920 は、プロセッサ 205 の実行下で、S E I メッセージ 1313 から b v _ c n t シンタックス要素を復号する。b v _ c n t シンタックス要素は、使用された基底ベクトルカウン트의更新を示す S E I メッセージ 1313 から復号された b a s i s _ v e c t o r _ c n t _ u p d a t e が宛先デバイス 140 に伝達される必要があるときに復号される。ステップ 2230 は、復号に使用される基底ベクトルの数を示す情報を復号するように動作する。使用されるべき基底ベクトルの数を指定する情報は、2210 において導出されるようなベクトルの最大個数と、ステップ 2230 において復号される使用されるべき実際の量との両方を含むと考えることができる。テンソル 115 中の各レイヤは、別個の基底ベクトルカウンートを有し得る。シグナリングされた基底ベクトルカウンートは値 N を示し、ここで、N は、定義された閾値に従ってテンソルを適切に再構成するために選択された M 個の基底ベクトルのうちの最初の N 個の基底ベクトルであるサブセットを定義する。使用される基底ベクトルの数を確立するための閾値の使用は、テンソル 115 の統計への適応を可能にする。b v _ c n t シンタックス要素は、フレームごとに、すなわち符号化データのフレームごとに、決定または復号することができる。

【0240】

指定された品質レベル、すなわち、所与の説明された分散しきい値に到達するために必要とされる基底ベクトルがより少ない場合、追加の基底ベクトルが省略され得、圧縮効率を改善する。基底ベクトルの大部分を保持するために、98% または 99% などの閾値を使用することができる。ほとんどの場合、すべての M 個の基底ベクトルが保持され、すなわち、N は M に等しいが、値 N はデータ依存方法で低減され得る。より低い品質が許容可能である場合、より良好なビットレートまたは性能トレードオフは量子化ステップサイズをさらに増加させるよりも優先的に閾値を減少させることによって、すなわち、量子化パラメータを増加させることによって達成される。例えば、85% の閾値はビットレートを低減し、一方で、QP をより高いレベル（例えば、45 より大きい QP）に増加させるよりも、タスク性能を相対的に著しく低下させることは、非常にビット圧縮されたレートが望まれる場合に好適であり得る。95% の閾値は、達成可能なタスク性能の中程度の低減でビットレートを低減し、無損失に近い圧縮と非常に低いビットレート動作点の圧縮と間の性能を必要とするユースケースに適している。閾値および QP は、「パレートフロント (Pareto front)」、すなわち、最適な閾値および QP 値からなるビットレートまたは

タスク性能フロンティア曲線を生成するように、一緒に最適化され得る。このパレートフロントを導出するために様々な閾値およびQP値を徹底的にテストすることは、一般に、ソースデバイス110にとって法外に高価であり、適切な値を示すための発見的規則が望ましい。実験は、画素当たりの下位ビット(BPP)について、QPを増加させる代わりに閾値を減少させることが好ましいことを示している。さらに、しきい値は異なる符号化レイヤ、たとえば、FPNの、またはテンソルコンバイナ162から得られるFPNの結合表現に対して変更され得る。実験は、より多くの分解されたレイヤがわずかに高い閾値を与えられ得、ビットレートの比較的制限された増加においてタスク性能の改善を達成することを示す。たとえば、テンソル115のP2/P3連結レイヤ、P4、およびP5レイヤの90%、94%、および98%の閾値は、すべてのレイヤにわたって適用される95%の90%などの固定閾値よりも良好な性能を提供する。より分解されたレイヤ(例えば、P4およびP5)はより分解されていないレイヤ(例えば、P2またはP3)と比較して、より小さいエリアを有する。したがって、サブピクチャ1216中のパッキングされたエリアの増分増加は、より分解されていないレイヤについての同様の追加の基底ベクトルカウントよりも小さい。最大基底ベクトルカウントは同様に、レイヤごとに変更され得る。プロセッサ205における制御は、ステップ2230から基底ベクトルサブピクチャ復号ステップ2240に進む。

10

【0241】

ステップ2240において、ピクチャデコーダ804は、プロセッサ205の実行下で、パッキングされた基底ベクトルを含むサブピクチャ1216を復号する。ステップ2240は、少なくともステップ2230で復号された情報に従って、符号化されたデータから基底ベクトルを効果的に導出する。プロセッサ205における制御は、ステップ2240から基底ベクトルアンパックステップ2250に進む。

20

【0242】

ステップ2250において、アンパッカ822は、プロセッサ205の実行の下で、ステップ2230においてピクチャ1200において使用されるようにシグナリングされた基底ベクトルの数に従って、サブピクチャ1216から数bv_cnt基底ベクトルをアンパックする。ピクチャ1200において使用されるようにシグナリングされる基底ベクトルの数は、ステップ2220において決定されたパッキング構成によって与えられる基底ベクトルの最大個数未満であり得る。ステップ2250は、少なくとも使用されるべき基底ベクトルの数を指定する情報に従って、符号化データから基底ベクトルを導出する。複数のレイヤがサブピクチャ1216にパッキングされる場合、アンパッキングは、SEIメッセージ1313から取得されたそれぞれの基底ベクトルカウントbv_cnt[layer_idx]値に従って各レイヤについて動作する。方法2200は終了し、プロセッサ205における制御は方法1500に戻り、ステップ1550において処理を再開する。

30

【0243】

図23は、ネットワーク内の分割点における異なるレイヤのテンソルと、テンソルコンバイナ162などのテンソルコンバイナから符号化テンソルから再生するための結果として得られるオプションと、の間の関係を示す概略ブロック図2300である。結合テンソル2310は、src_layer_idxによって識別されるテンソル149aのテンソルである。結合テンソル2310から、1つまたは2つの出力テンソル、すなわちテンソル2320および2330のいずれかが、テンソル149の一部として出力される。テンソル2320は、テンソル149内の位置src_layer_idxに留まり、テンソル2330は、存在する場合、テンソル149内の位置dst_layer_idxに挿入される。図23は、仮に存在した場合、SEIメッセージから復号され、付録Aを参照して説明されたフラグ2315(すなわち、layer_extraction_flag, layer_upsample_flag, layer_downsample_flag, resample_vertical_only_flag, resample_horizontal_only_flag)に基づくテンソル2320の幅および高さと比較した、テンソル2330の空間的関係を示す。フラグ2315は、ステップ1910で決定され、ステ

40

50

アップ1920でレイヤマッピング1390として符号化される。フラグ2315は、ステップ2010において、テンソル2320および存在する場合にはテンソル2330を生成するために復号される。結合テンソル2310は、 $k_1 + k_2$ チャンネルを有し、ここで、 k_1 はテンソル2320のチャンネルに対応し、 k_2 はテンソル2330のチャンネルに対応し、境界は`src_channel_offset`を使用してシグナリングされる。フラグ2315は、抽出されたテンソルのための様々な再サンプリングオプションを可能にするためのシグナリングを提供する。テンソル149はテンソル115aと同じ次元を有する必要があるので、使用され得るフラグ2315の利用可能な組合せは、ネットワークおよび分割点に依存するテンソル115aの次元に依存する。

【0244】

10

ソースデバイス110および宛先デバイス140の構成では、パックまたはアンパックされるべき基底ベクトルの数、すなわち`bv_cnt`は、パッキング構成によって与えられる基底ベクトルの最大個数に対して相対的に、すなわち`bv_max_cnt`に対して相対的に表される。一般に、最大利用可能基底ベクトル（すなわち、`bv_max_cnt`）の大部分が使用され得、したがって、`bv_max_cnt`と`bv_cnt`との間の差は比較的小さくなり得る。`bv_max_cnt`と`bv_cnt`との間の差が小さいとき、指数Golombシンタックス要素を使用して`bv_max_cnt`と`bv_cnt`との間の差を符号化することによって、圧縮効率が改善され得る

【0245】

20

PCAエンコーダ160の構成では、それぞれ基底ベクトルに適用可能な平均係数および係数のためのサブピクチャエンコーダ532および572が、QPコントローラ790によって、それぞれサブピクチャエンコーダ514および556によって平均特徴または基底ベクトルを符号化するために適用されるよりも低い量子化パラメータを適用するように構成される。サブピクチャエンコーダ532および572内のQPコントローラモジュール790のそれぞれのインスタンスは、ベースQPにオフセットを適用することができ、ベースQPは、基底ベクトルを符号化するCUなどの他の場所に適用される。QPオフセットは、サブピクチャ1212および1214などにおいて、基底ベクトルに適用可能な平均係数または係数の残差を符号化するとき使用される量子化ステップサイズを低減するために、マイナス6またはマイナス12、マイナス18、またはマイナス24などの値、または他の値を有し得る。サブピクチャ符号化係数の量子化ステップサイズを低減するためにQPオフセットを適用することは、PCAエンコーダ160中の順変換（たとえば、524および564）によって生成された係数と比較して、PCAデコーダ170中で生成された復号された係数間の不一致を低減する。係数は、再構成された特徴マップ上のそれぞれの平均特徴または基底ベクトルの寄与を変調する際の係数の役割によって、特徴マップの再構成に対して乗法的な効果を有する。したがって、係数に対する非可逆符号化の影響は、再構成された特徴マップにおける忠実度の実質的な損失を引き起こし、CNNヘッド150におけるタスク性能に悪影響を及ぼす。したがって、係数は、基底ベクトルよりも高い品質レベルで符号化される必要がある。係数を表すことを担うサブピクチャまたはスライスに、固定の比較的低いQPを適用することも可能である。係数がピクチャ1200全体の比較的小さいエリアを占めるので、低いQPの使用による使用されるビットレートの増加は比較的小さく、いずれの場合も、再構成された特徴マップの忠実度を改善する。QPは「量子化グループ」（QG）として知られるエリアを指定することによって、スライスレベルよりも細かい粒度で変更することができ、これは、「デルタQP」シンタックス要素を使用してQPを変更することができる各CTU内のエリアの最大粒度に対応する。したがって、係数および基底ベクトルが同じスライス中で符号化されるべきである場合、デルタQPは、基底ベクトルのために必要とされるよりも高い精度で係数を符号化するための別の機構を与える。

30

40

【0246】

ピクチャ1200の構成では、サブピクチャではなくタイルへの分割が使用される。タイルを用いた構成では、平均特徴マップ1220、平均係数1222、基底ベクトル12

50

26、および基底ベクトル係数1224が別々のタイルに含まれる。タイルは、PCAエンコーダ160において独立して符号化され、PCAデコーダ170において復号される。

【0247】

PCAエンコーダ160およびPCAデコーダ170の別の構成では、平均特徴がテンソル115のすべての特徴マップに均一に適用される。言い換えれば、すべての平均係数は、ドット積モジュール524によって特徴マップごとに導出されるのではなく、「1」の値が割り当てられる。平均係数が「1」に割り当てられる場合、PCAエンコーダ160では、再構成された平均特徴マップ522が平均係数によってスケーリングされる必要がなく、したがって、モジュール524、528、532、536、540は必要なく、522は542としてモジュール544に直接渡される。PCAデコーダ170では、モジュール832、836、および850は必要とされず、復号された平均特徴マップ848は852として合計モジュール856に渡される。平均係数が「1」の値を割り当てられるとき、サブピクチャ1212は、一定の平均係数値を符号化する必要がないので、ピクチャ1200から省略され得る。サブピクチャ1212がピクチャ1200から省略されるとき、基底ベクトルのための係数を含むサブピクチャ1214は、係数をパッキングするために必要とされる最小幅と比較して、幅が増加され得る。サブピクチャ1214の幅は、パッキングされた係数の幅（すなわち、パッキングされたテンソル115のレイヤ中のチャンネルの最大数）の1.5倍に設定され得る。サブピクチャ1214の幅を増加させると、より広いアスペクト比を有するピクチャ1200が得られ、これは、個々の基底ベクトルの一般的に遭遇するアスペクト比により適しており、したがって、ピクチャ1200内の未使用エリアがより少なくなる。サブピクチャ1210、1214、および1215は、ピクチャ1200の全体がサブピクチャによって占有されるように、ピクチャ構造を維持するために幅が調整され得る。一定の平均係数の使用は、また、関連する量子化および逆量子化の必要性を除去する。一定の平均係数の使用は、依然として、順変換および逆変換（射影）の可逆動作を維持するために必要な、導出された基底ベクトルに対する順変換（ドット積）の前にテンソル115を「ゼロセンタリング」する目的に役立つ。

【0248】

本明細書で提示される方法は、VVCまたはHEVCなどの現代のブロックベースの圧縮規格を使用して、圧縮に適したフォーマットでテンソルを効率的に表現することを可能にする。ブロックベースの圧縮は、特徴マップを再構成するために基底ベクトルを投影するための係数などのデータに直感的に適用可能ではないが、トレーニングされた二次変換を含む様々な変換を使用することなどによって、ブロック内の追加の予期しない冗長性を明らかにする。本明細書に提示される方法は、「高速RCNN」および「YOLOv3」ネットワークアーキテクチャ、ならびに「バックボーン」および「ヘッド」部分へのこれらのネットワークの特定の分割を参照して説明されるが、本方法は多次元テンソルデータについて動作する任意のニューラルネットワークに適用可能であり、そのようなネットワークの「バックボーン」および「ヘッド」部分への異なる分割に適用可能である。

【0249】

産業上の利用可能性

記載された構成は、コンピュータおよびデータ処理産業に適用可能であり、特に、高い圧縮効率を達成する、ビデオおよび画像信号などの信号の符号化および復号のためのデジタル信号処理に適用可能である。

【0250】

チャンネルまたは特徴マップのグループ内の浮動小数点テンソルデータを量子化し、対数量子化領域を使用して、得られた整数値を平面フレームにパッキングするための構成も開示される。対数量子化領域を使用する量子化および逆量子化方法は、大きな大きさのテンソル値のための正確な値を符号化するのに費やされるビットがないことに起因して、より大きな圧縮効率を可能にし、そのような精度は、使用中のネットワークのためのタスク性能のさらなる改善をもたらさない。

【 0 2 5 1 】

上記は本発明のいくつかの実施形態のみを説明し、本発明の範囲および趣旨から逸脱することなく、修正および／または変更を行うことができ、実施形態は例示的であり、限定的ではない。

【 0 2 5 2 】

付録 A

ビットストリームにおける基底ベクトルパッキング（すなわち、1 3 9 1）に関連するメタデータを表すための例示的な S E I メッセージフォーマットおよび関連するセマンティクスは、以下の通りである：

【 0 2 5 3 】

10

20

30

40

50

【表 1】

basis_vector_packing_info(payloadSize) {	Descriptor
basis_vector_dimensions_update	u(1)
if(basis_vector_dimensions_update != 0) {	
layer_cnt	u(1)
for(layer_idx = 0; layer_idx < layer_cnt; layer_idx++) {	
bv_max_cnt [layer_idx]	ue(v)
bv_width [layer_idx]	ue(v)
bv_height [layer_idx]	ue(v)
}	
}	
basis_vector_cnt_update	u(1)
if(basis_vector_cnt_update)	
{	
for(layer_idx = 0; layer_idx < layer_cnt; layer_idx++) {	
bv_cnt [layer_idx]	ue(v)
}	
}	
layer_update_mapping_info	u(1)
if(layer_update_mapping_info)	
{	
layer_extraction_flag	u(1)
if(layer_extraction_flag)	
{	
layer_upsample_flag	u(1)
layer_downsampling_flag	u(1)
if(layer_upsample_flag layer_downsampling_flag)	
{	
resampling_filter_idx	ue(v)
resample_vertical_only_flag	u(1)
resample_horizontal_only_flag	u(1)
}	
}	
src_layer_idx	ue(v)
src_channel_offset	ue(v)
dst_layer_idx	ue(v)
}	
}	
}	

10

20

30

40

【0254】

基底ベクトルパッキング情報セマンティクス

上記のシンタックス構造は、基底ベクトル平面フレームをアンパックし、推論タスクを実行するために復号テンソルに投影するための基底ベクトルに変換するために必要な情報を指定する。

【0255】

記述子 u (n) を有するシンタックス要素は、シンタックス要素が n ビットを使用して符号化され、符号なし整数値として解釈されることを示す。記述子 u e (v) を有するシンタックス要素は、シンタックス要素が指数ゴロム値として符号化され、符号なし整数値

50

として解釈されることを示す。

【0256】

基底ベクトルパッキング情報 S E I メッセージの持続性は、関連する A U から、基底ベクトルパッキング情報 S E I メッセージの次の発生または C L V S のエンドまでである。基底ベクトルパッキング情報 S E I メッセージの次の発生時に、パラメータは、更新フラグに基づいてグループ化される。更新されない（すなわち、上記のシンタックステーブルのブランチ内に含まれ、そのブランチの更新フラグがゼロに設定されている）パラメータは、それらの以前の値（存在する場合）を保持する。

【0257】

1 に等しい `basis__vector__dimensions__update` は、現在および後続のフレームに適用可能な、すなわち 1 に等しい `basis__vector__dimensions__update` を有する基底ベクトルパッキング情報 S E I メッセージの次の発生までの、最大基底ベクトルカウントおよび基底ベクトル次元の更新を指定する。C L V S における基底ベクトルパッキング情報 S E I メッセージの第 1 のインスタンスが 1 に等しい `basis__vector__dimensions__update` を有することは、ビットストリーム適合性の要件である。

10

【0258】

メモ：通常、1 に等しい `basis__vector__dimensions__update` を有する基底ベクトルパッキング情報 S E I メッセージは、C L V S の第 1 ピクチャと共に送信される。

20

【0259】

1 に等しい `basis__vector__cnt__update` は、C L V S 内の現在および後続のフレームに適用可能な、使用される基底ベクトルカウントの更新を指定する。使用される基底ベクトルカウントは、最大基底ベクトルの基底ベクトルが何個使用されているかを示す。

【0260】

`layer__cnt` は、フレーム内に存在するレイヤの数を指定する。

【0261】

`bv__max__cnt[layer__idx]` は、`layer__idx` について存在する基底ベクトルの最大数の数を指定する。

30

【0262】

`bv__width[layer__idx]` は、`layer__idx` の基底ベクトルの幅を指定する。

【0263】

`bv__height[layer__idx]` は、`layer__idx` の基底ベクトルマップの高さを指定する。

【0264】

`bv__cnt[layer__idx]` は、`layer__idx` について存在する基底ベクトルの数の数を指定する。0 から `layer__cnt-1` までの範囲における各値 `layer__idx` について、`bv__cnt[layer__idx]` が、`bv__max__cnt[layer__idx]` 以下であることは、ビットストリーム適合性の要件である。`bv__cnt[layer__idx]` がビットストリームから復号されていないとき、値 `bv__cnt[layer__idx]` は、0 から `layer__cnt-1` までの範囲の `layer__idx` について `bv__max__cnt[layer__idx]` に等しく設定される。

40

【0265】

1 に等しい `layer__update__mapping__info` は、テンソルコンバイナ 162 によって決定され、テンソルセパレータ 172 によって実行されるテンソル再マッピングの構成に対する更新を指定する。`layer__update__mapping__info` がゼロに等しいとき、テンソル再マッピングの構成に対する更新は行われず、

50

テンソル再マッピングの任意の以前の構成が適用され続ける。

【0266】

1に等しい`layer_extraction_flag`は、テンソル内の最後のチャネルに対するチャネル`src_channel_offset`に対応するテンソルの一部分が、`dst_layer_idx`におけるテンソルのリストに挿入されることを指定し、新しいレイヤを生成するために、レイヤ`src_layer_idx`からのテンソルが抽出される。`layer_extraction_flag`がゼロに等しいとき、テンソルの抽出は実行されず、その結果、テンソル149はテンソル149aに等しく設定される。ビットストリーム`layer_extraction_flag`から以前に解析されていない場合、ゼロであると推論される。

10

【0267】

1に等しい`layer_upsample_flag`は、抽出されるテンソルの部分がアップサンプリングされて、より大きい、例えば、入力テンソルの2倍の幅および2倍の高さを有する新しいテンソルを生成することを指定する。

【0268】

1に等しい`layer_downsample_flag`は、抽出されるテンソルの部分がダウンサンプリングされて、より小さい、例えば、入力テンソルの幅の半分および高さの半分の有する新しいテンソルを生成することを指定する。

【0269】

`layer_upsample_flag`および`layer_downsample_flag`がともに、基底ベクトルパッキング情報SEIメッセージの1つのインスタンスにおける1つに等しくないことは、ビットストリーム適合性の要件である。

20

【0270】

`resampling_filter_idx`は、新しいテンソルを生成するときにテンソルの部分を再サンプリングする際に適用されるフィルタリングのタイプを指定する。再サンプリングは、`resample_vertical_only_flag`および`resample_horizontal_only_flag`に従って、水平または垂直または水平および垂直に適用される。次の表に、使用可能なフィルタを列挙する：

【0271】

【表2】

30

<code>resampling_filter_idx</code>	ダウンサンプリング時	アップサンプリング時
0	デシメーション	バイリニアフィルタ
1	アンチエイリアシング、デシメーション	バイキュービックフィルタ
2	将来使用するため予約	最近傍
3-	将来使用するため予約	将来使用するため予約

40

【0272】

1に等しい`resample_vertical_only_flag`は、リサンプリング動作が垂直方向にのみ適用されることを指定する。

【0273】

1に等しい`resample_horizontal_only_flag`は、リサンプリング動作が水平方向にのみ適用されることを指定する。

【0274】

`resample_vertical_only_flag`および`resample_horizontal_only_flag`は両方とも、基底ベクトルマッピング情報SEIメッセージのいずれのインスタンスにおいても1に等しくないことが、ビットストリ

50

ーム適合の要件である。

【0275】

`src_layer_idx`は、テンソル149に含まれる新しいテンソルを生成するために、再サンプリングのために、テンソル149aの部分のどのテンソルから抽出されるべきかを指定する。

【0276】

`src_channel_offset`は、`src_layer_idx`によって識別されるテンソルが分割されるチャンネルオフセットを指定する。`src_layer_idx`によって指定されたテンソルのチャンネル0～`src_channel_offset-1`が保持され、テンソル149の対応するテンソルとして出力される。`src_layer_idx`によって指定されるテンソルの最後のチャンネルのチャンネル`src_channel_offset`は、再サンプリングの対象となり、その結果は、テンソル149内の追加のテンソルとして出力される。

【0277】

`dst_layer_idx`は、リサンプリングされたテンソルがテンソル149aに挿入されてテンソル149が生成されるインデックスを指定する。インデックス値0は開始時の挿入を指定し、すなわち、新しい最初の場所を作成し、`layer_cnt`に等しいインデックス値は、最後の場所の後の挿入を指定し、すなわち、新しいテンソルを追加して、新しい最後の場所を作成する。1から`layer_cnt-1`までのインデックス値は、テンソル149を作成するために、位置`dst_layer_idx`および`dst_layer_idx+1`におけるテンソル149aの間の位置への再サンプリングされたテンソルの挿入を示す。

【図面】

【図1】

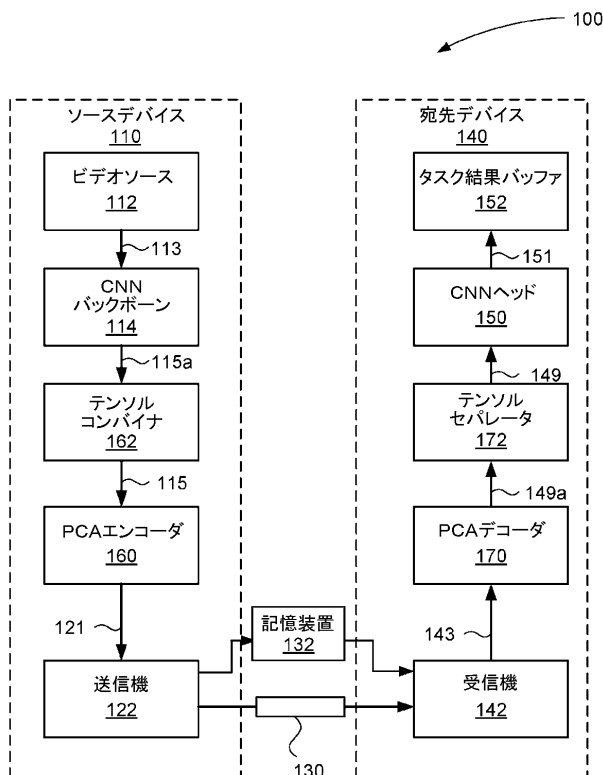


Fig. 1

【図2A】

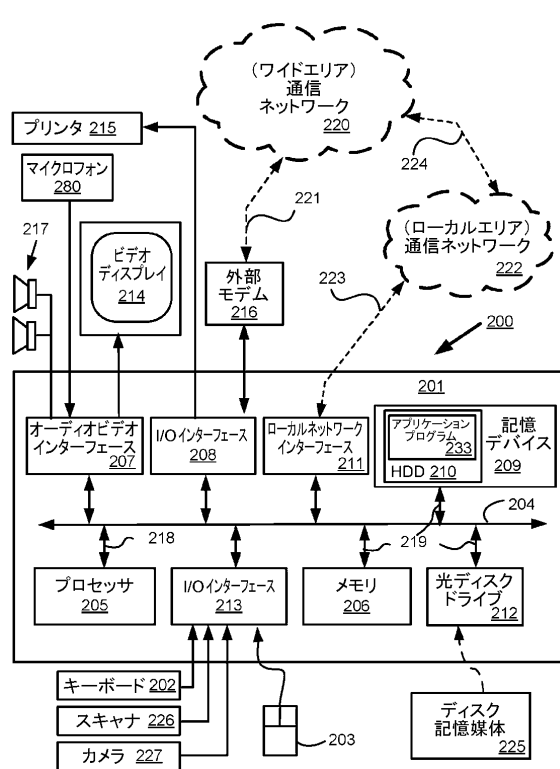


Fig. 2A

10

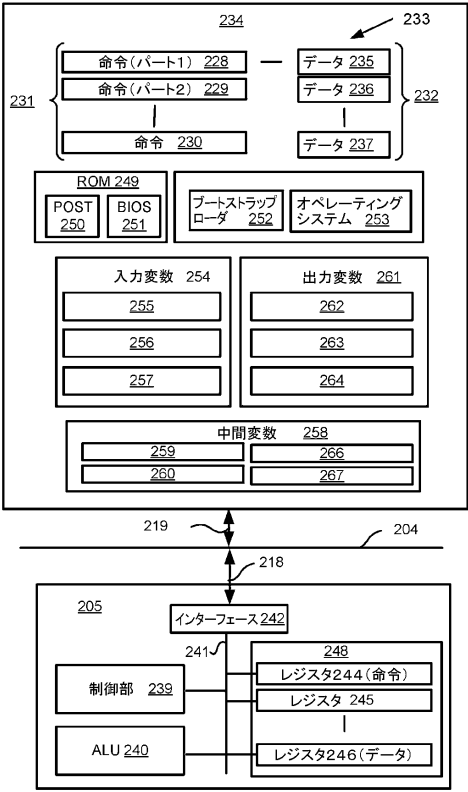
20

30

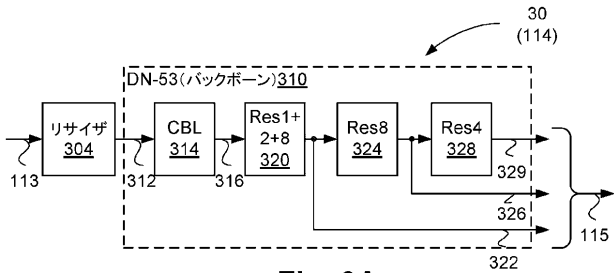
40

50

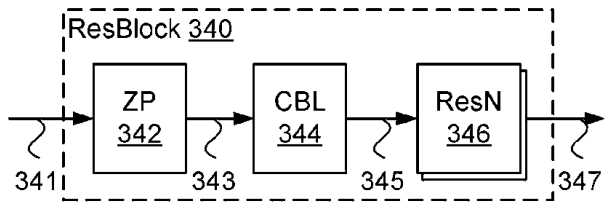
【図 2 B】



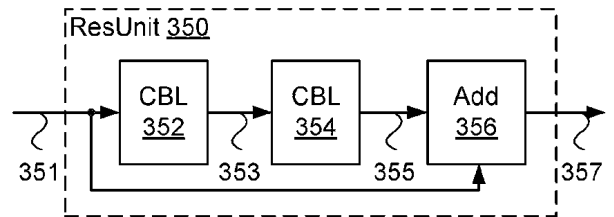
【図 3 A】



【図 3 B】



【図 3 C】



【図 3 D】

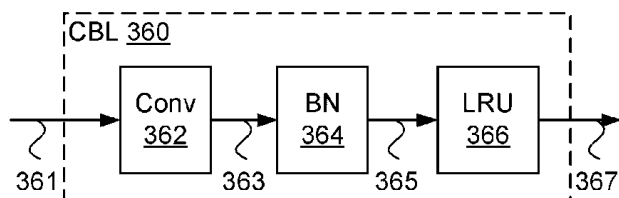


Fig. 3D

【圖 4】

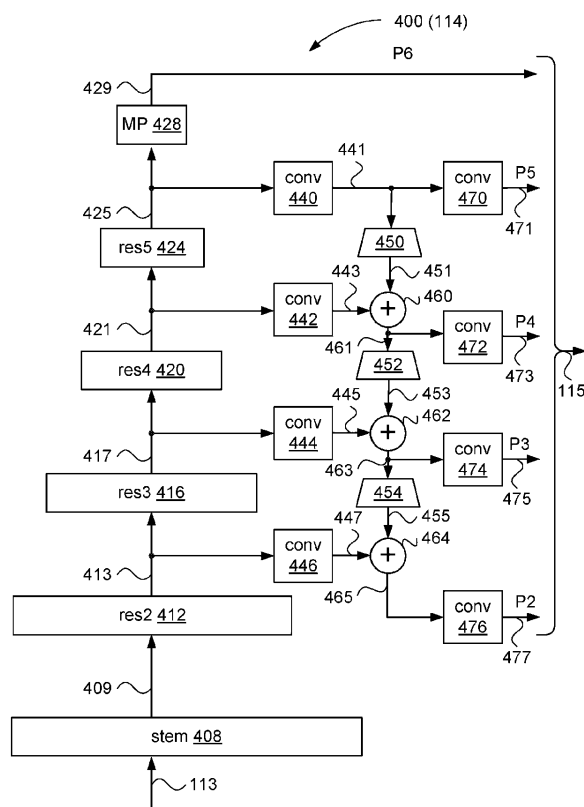


Fig. 4

【図 5】

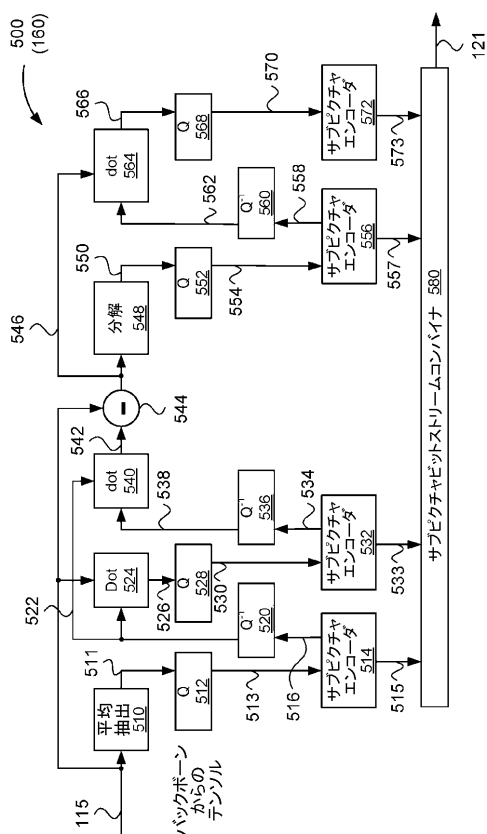


Fig. 5

【圖 6】

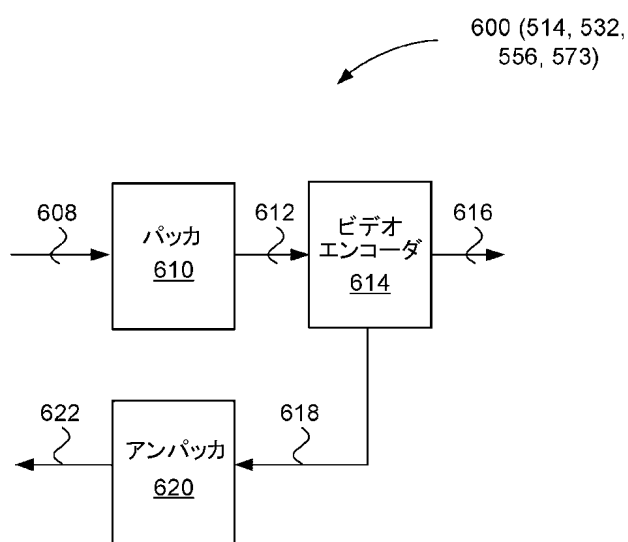


Fig. 6

10

20

30

40

50

【図 7】

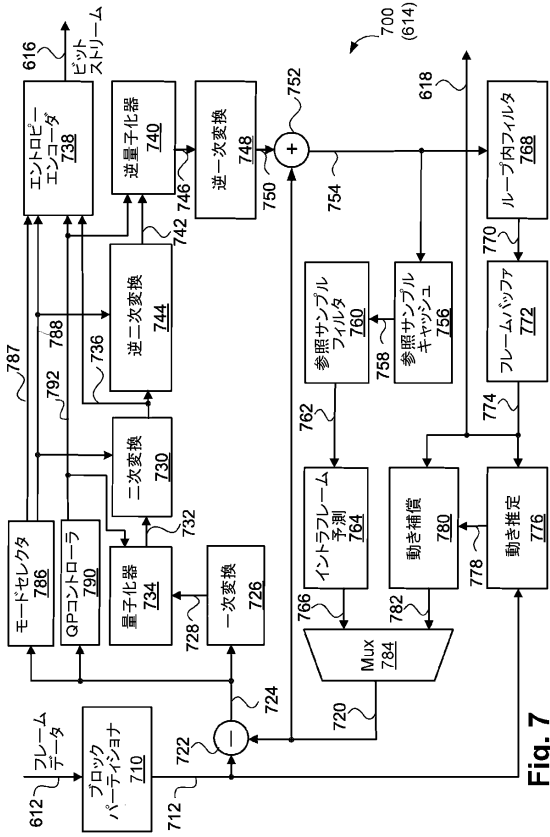


Fig. 7

【図 8】

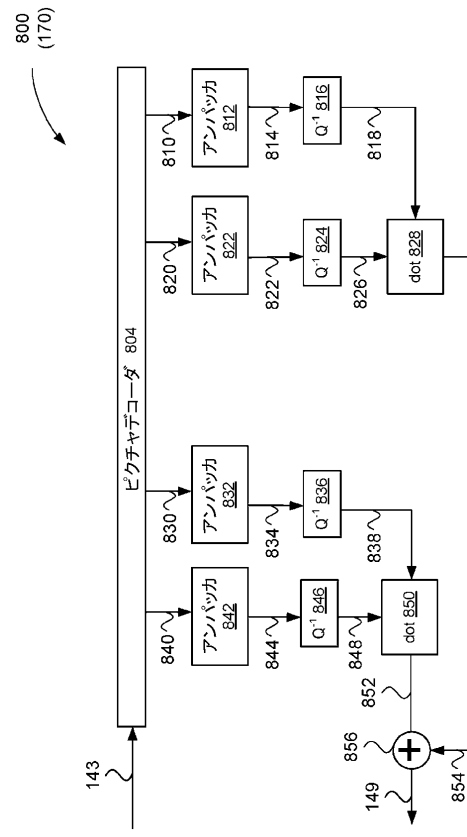


Fig. 8

【図 9】

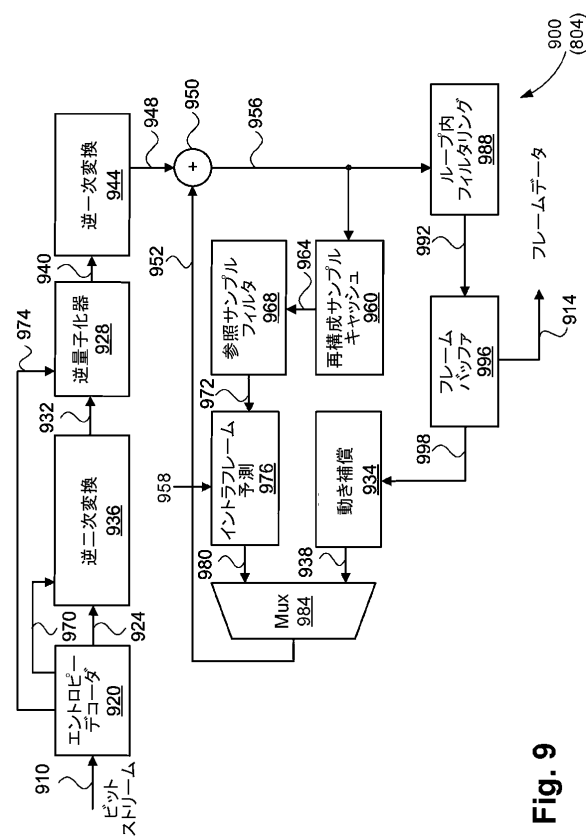


Fig. 9

【図 10 A】

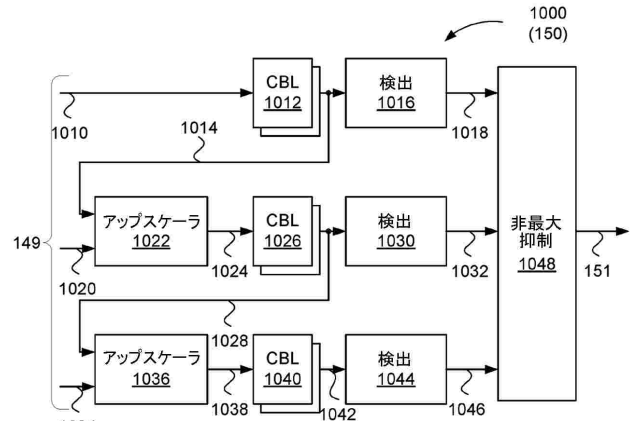


Fig. 10A

【図 1 0 B】

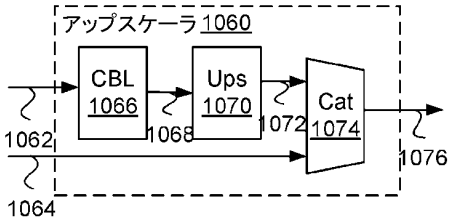


Fig. 10B

【図 1 0 C】

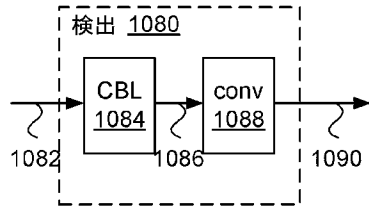


Fig. 10C

【図 1 1】

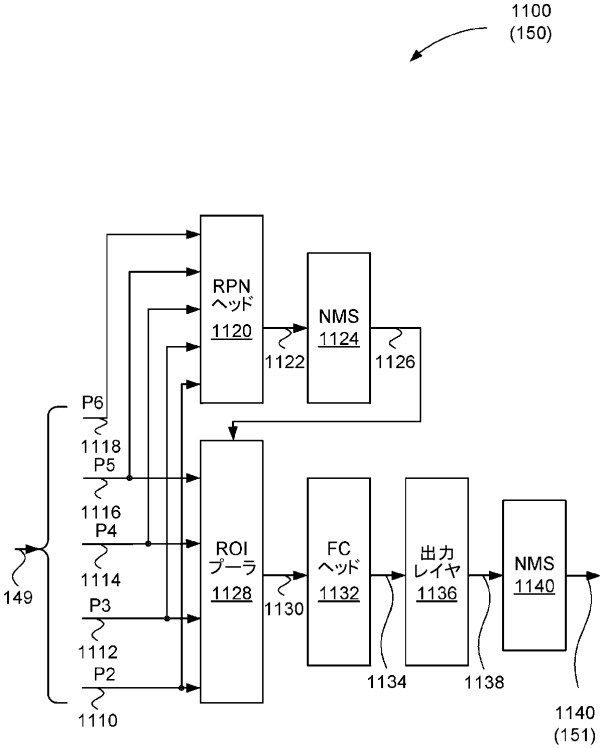


Fig. 11

【図 1 2 A】

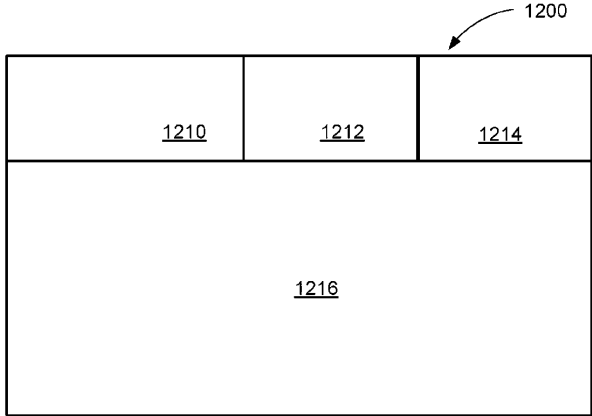


Fig. 12A

10

20

30

40

50

【図 1 2 B】

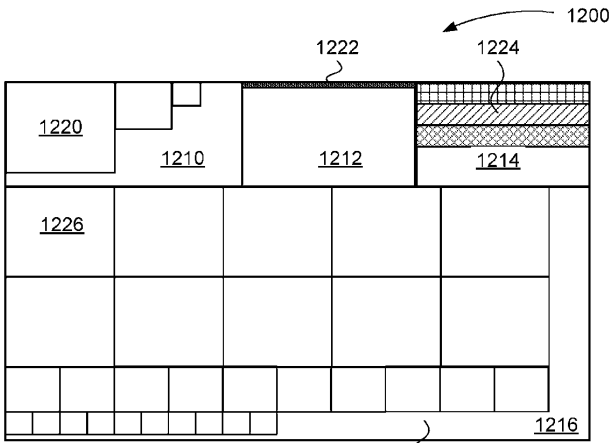


Fig. 12B

【図 1 3】

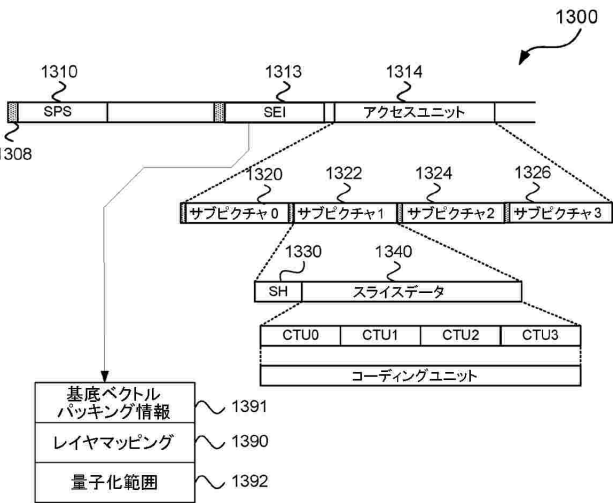


Fig. 13

【図 1 4】

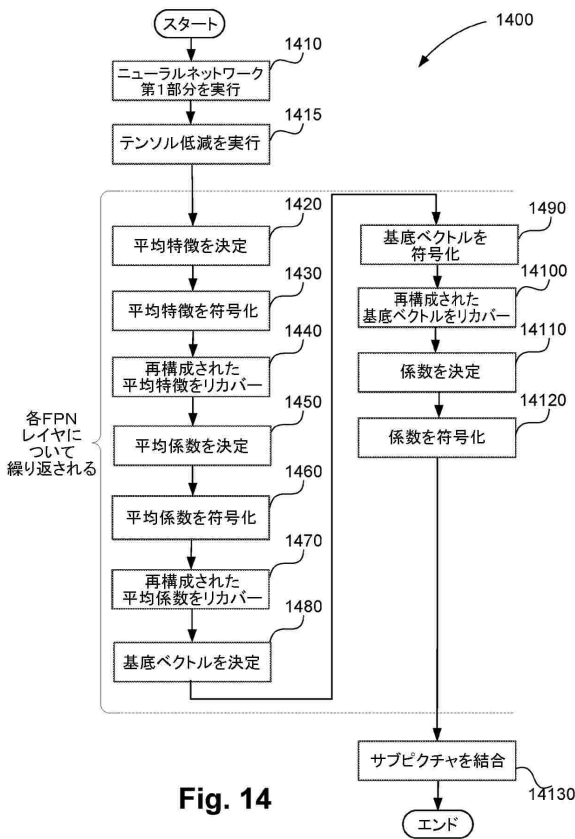


Fig. 14

【図 1 5】

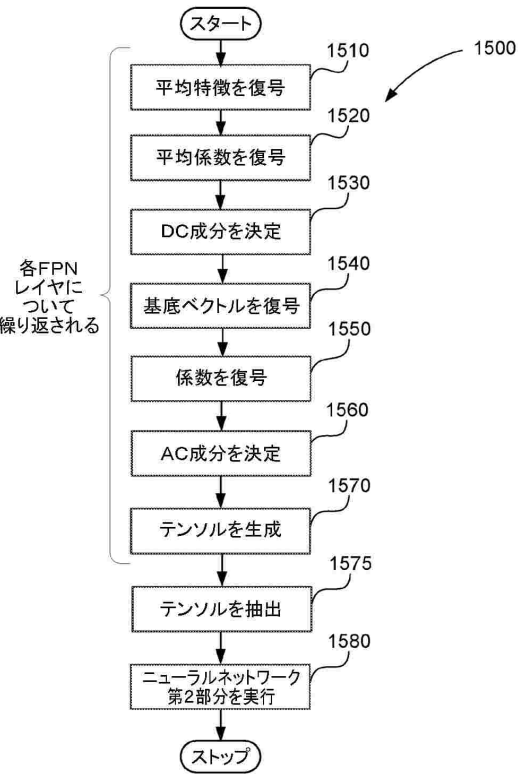


Fig. 15

【図 16 A】

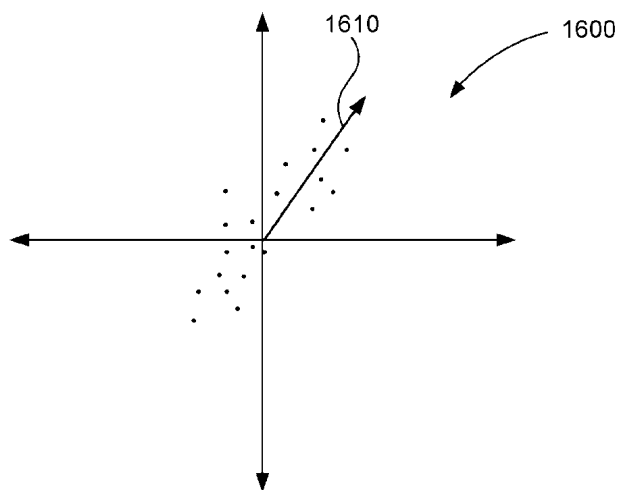


Fig. 16A

【図 16 B】

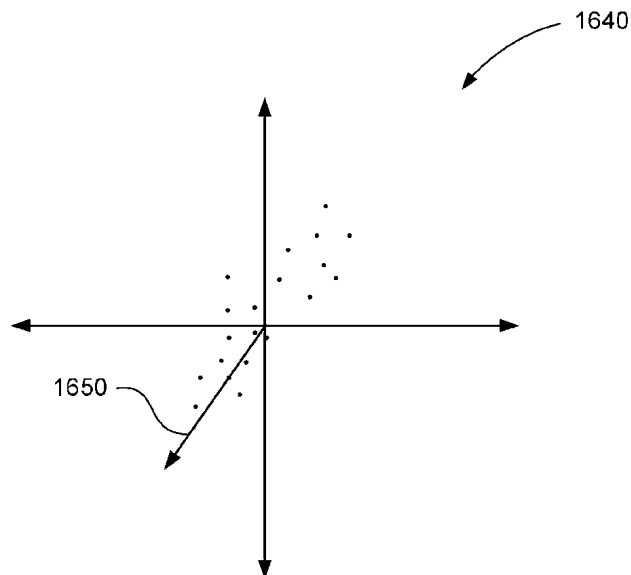


Fig. 16B

【圖 17】

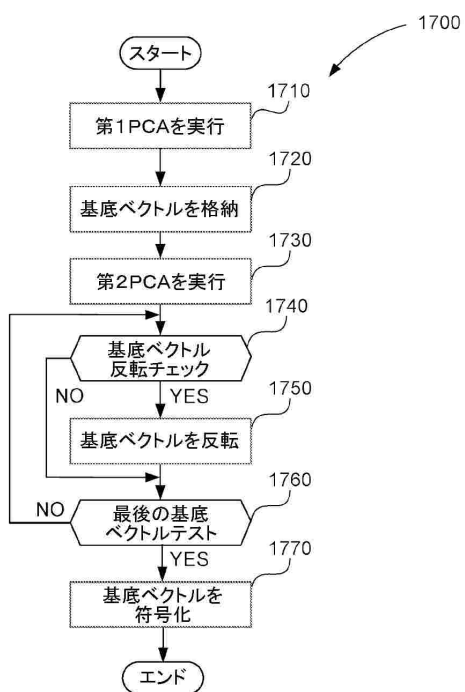


Fig. 17

【図 18 A】

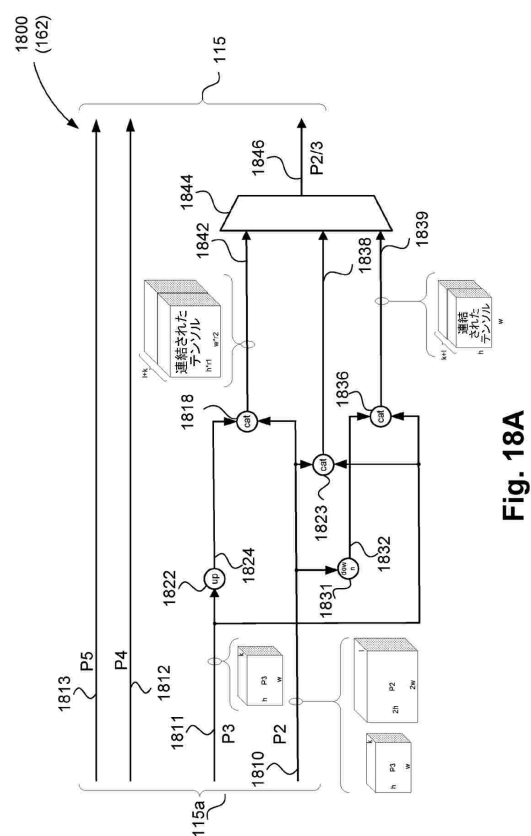
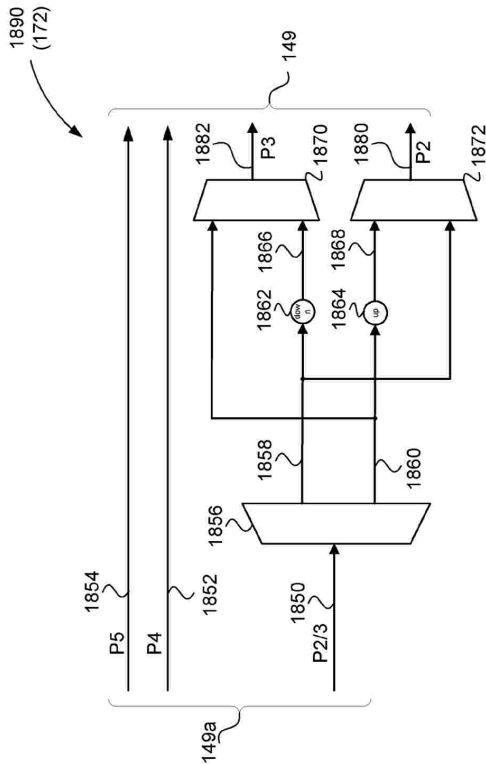


Fig. 18A

【図 18 B】



【図 19】

Fig. 18B

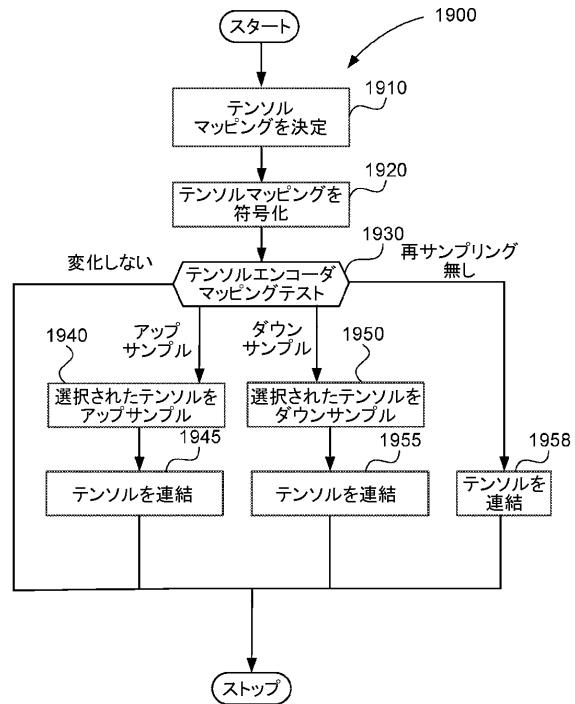


Fig. 19

【図 20】

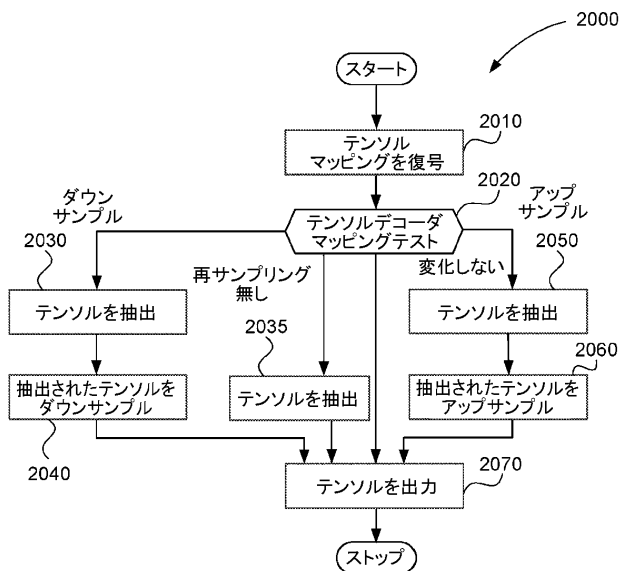


Fig. 20

【図 21】

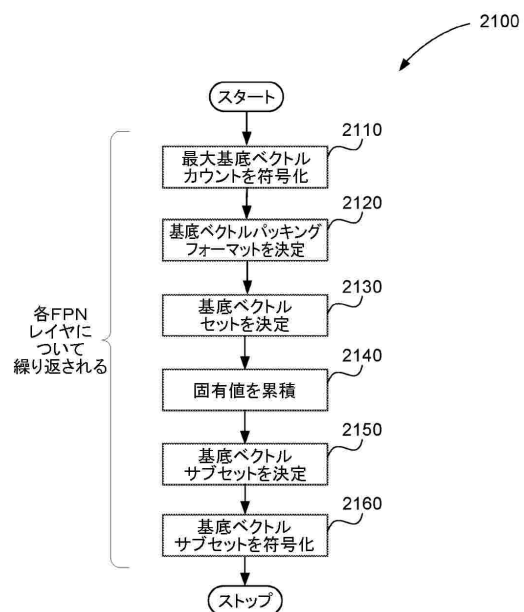


Fig. 21

10

20

30

40

50

【図 2 2】

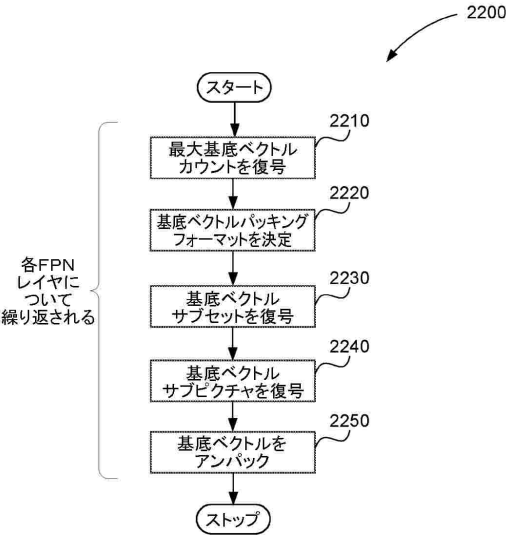


Fig. 22

【図 2 3】

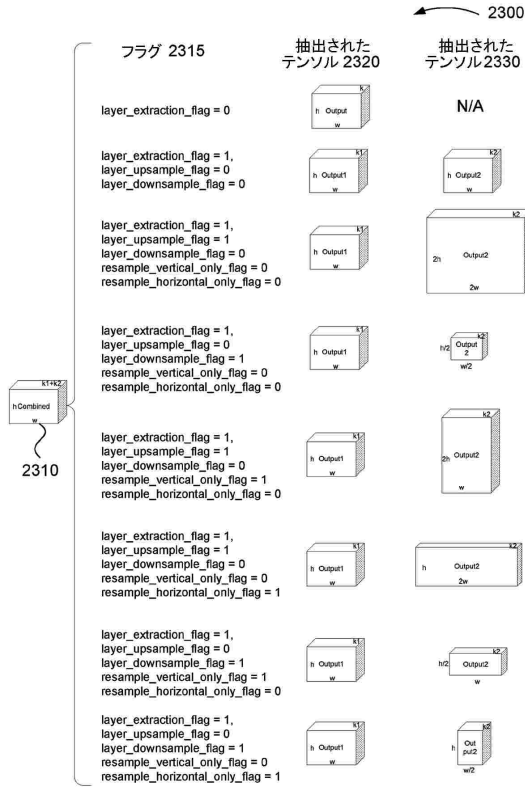


Fig. 23

【国際調査報告】

INTERNATIONAL SEARCH REPORT		International application No. PCT/AU2023/050291
A. CLASSIFICATION OF SUBJECT MATTER H04N 19/169 (2014.01) G06N 3/0464 (2023.01) H04N 19/159 (2014.01) H04N 19/172 (2014.01) H04N 19/18 (2014.01) H04N 19/46 (2014.01) H04N 19/50 (2014.01) H04N 19/91 (2014.01) H04N 19/96 (2014.01)		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) Databases: PATENW (EPODOC, WPIAP, TXTE & TXTMT), INSPEC, Google Patents, Google Scholar and JNET documents: IPC/CPC – G06N3/02, G06N3/0464, H04N19/46, H04N19/85, H04N19/90, and keywords: mean, average, feature, quantisation, integer, feature map, tensor, inverse-quantisation, basis vector, decode, encode, tensor, feature coefficients, subpicture, PCA and similar terms.		
Applicant/Inventor(s) name search was conducted in external databases DOCDB, DWPI and IP Australia internal databases.		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Documents are listed in the continuation of Box C	
☒ Further documents are listed in the continuation of Box C ☒ See patent family annex		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "D" document cited by the applicant in the international application earlier application or patent but published on or after the international filing date "E" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "L" document referring to an oral disclosure, use, exhibition or other means "O" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family		
Date of the actual completion of the international search 23 June 2023		Date of mailing of the international search report 23 June 2023
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA Email address: pct@ipaaustralia.gov.au		Authorised officer Jae-Hak Kim AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No. +61 2 6283 2251

INTERNATIONAL SEARCH REPORT		International application No.
C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		PCT/AU2023/050291
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2021/003210 A1 (VID SCALE, INC.) 07 January 2021 Title, Abstract, Figs. 7, 8 and 14, Paragraphs [0005], [0006], [0095], [0102], [0144], [0145], [0155], [0158], [0164], [0166] and [0187]	1-14
X	AIDINI, A. et al., "Tensor Decomposition Learning for Compression of Multidimensional Signals", IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 15, NO. 3, APRIL 2021, Pages 476-490 Title, Abstract, Figs. 1 & 2, Section I & III, Pages 476, 477, 479 & 480	1-14
X	WO 2021/254855 A1 (INTERDIGITAL VC HOLDINGS FRANCE, SAS) 23 December 2021 Figs. 1, 5 & 7, Page 2 lines 8-20; Page 5 lines 4-8 & 21-23; Page 6 lines 1-8 & 32-36; Page 9 lines 29-33; Page 28 lines 3-9 & 17-24	1-14

Form PCT/ISA/210 (fifth sheet) (July 2019)

INTERNATIONAL SEARCH REPORT		International application No.	
Information on patent family members		PCT/AU2023/050291	
This Annex lists known patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.			
Patent Document/s Cited in Search Report		Patent Family Member/s	
Publication Number	Publication Date	Publication Number	Publication Date
WO 2021/003210 A1	07 January 2021	WO 2021003210 A1	07 Jan 2021
		CN 114175056 A	11 Mar 2022
		EP 3994622 A1	11 May 2022
		US 2022261616 A1	18 Aug 2022
WO 2021/254855 A1	23 December 2021	WO 2021254855 A1	23 Dec 2021
		CN 116018757 A	25 Apr 2023
		EP 4168940 A1	26 Apr 2023
		IL 299171 A	01 Feb 2023
		KR 20230027152 A	27 Feb 2023
End of Annex			
Due to data integration issues this family listing may not include 10 digit Australian applications filed since May 2001. Form PCT/ISA/210 (Family Annex)(July 2019)			

10

20

30

40

50

フロントページの続き

,MC,ME,MK,MT,NL,NO,PL,PT,RO,RS,SE,SI,SK,SM,TR),OA(BF,BJ,CF,CG,CI,CM,GA,GN,GQ,GW,KM,
ML,MR,NE,SN,TD,TG),AE,AG,AL,AM,AO,AT,AU,AZ,BA,BB,BG,BH,BN,BR,BW,BY,BZ,CA,CH,CL,CN,C
O,CR,CU,CV,CZ,DE,DJ,DK,DM,DO,DZ,EC,EE,EG,ES,FI,GB,GD,GE,GH,GM,GT,HN,HR,HU,ID,IL,IN,IQ,I
R,IS,IT,JM,JO,JP,KE,KG,KH,KN,KP,KR,KW,KZ,LA,LC,LK,LR,LS,LU,LY,MA,MD,MG,MK,MN,MU,MW
,MX,MY,MZ,NA,NG,NI,NO,NZ,OM,PA,PE,PG,PH,PL,PT,QA,RO,RS,RU,RW,SA,SC,SD,SE,SG,SK,SL
,ST,SV,SY,TH,TJ,TM,TN,TR,TT,TZ,UA,UG,US,UZ,VC,VN,WS,ZA,ZM,ZW

ノン オーストラリア プロプライエタリー リミテッド内

F ターム (参考) 5C159 MA04 MA05 MA21 MC11 ME01 PP04 UA02 UA05