

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7403246号
(P7403246)

(45)発行日 令和5年12月22日(2023.12.22)

(24)登録日 令和5年12月14日(2023.12.14)

(51)国際特許分類

F I

H 0 4 N 19/70 (2014.01)

H 0 4 N 19/70

H 0 4 N 19/436 (2014.01)

H 0 4 N 19/436

請求項の数 7 (全27頁)

(21)出願番号	特願2019-115751(P2019-115751)	(73)特許権者	000001007
(22)出願日	令和1年6月21日(2019.6.21)		キヤノン株式会社
(65)公開番号	特開2021-2775(P2021-2775A)		東京都大田区下丸子3丁目30番2号
(43)公開日	令和3年1月7日(2021.1.7)	(74)代理人	110003281
審査請求日	令和4年6月21日(2022.6.21)		弁理士法人大塚国際特許事務所
		(72)発明者	大川 浩司
			東京都大田区下丸子3丁目30番2号
			キヤノン株式会社内
		(72)発明者	志摩 真悟
			東京都大田区下丸子3丁目30番2号
			キヤノン株式会社内
		審査官	鉢呂 健

最終頁に続く

(54)【発明の名称】 画像復号装置、画像復号方法

(57)【特許請求の範囲】

【請求項1】

画像を複数のブロックから成るブロック行を1つ以上含む矩形領域に分割して符号化されたビットストリームから前記画像を復号する画像復号装置であって、
並列処理の有効化に関する entropy_coding_sync_enabled_flag を復号し、スライスが矩形形状となる第1モードが使用されるか当該第1モードと異なるモードであってスライスに関する第2モードが使用されるかを示す rect_slice_flag を前記ビットストリームのピクチャパラメータセットから復号し、前記画像における矩形領域の垂直方向のブロック数に対応する第1情報を前記ピクチャパラメータセットから復号し、前記画像におけるスライスに含まれる矩形領域の数の第2情報を前記ビットストリームのスライスヘッダから復号し、前記画像が複数の矩形領域を含み且つ前記 rect_slice_flag が前記第2モードが使用されることを示す場合においてスライスのアドレスを前記スライスヘッダから復号する復号手段と、
前記 entropy_coding_sync_enabled_flag の値が1であり、前記 rect_slice_flag が前記第2モードが使用されることを示す場合において、前記スライスのアドレスと、前記矩形領域の垂直方向におけるブロックの数に対応する前記第1情報と、前記スライスに含まれる矩形領域の数の前記第2情報と、に基づいて、前記スライスに対し、ブロック行の符号データの先頭位置を特定するための情報の数を特定する特定手段と
を備え、

前記復号手段は、前記特定手段によって特定した前記先頭位置を特定するための情報の数と、前記先頭位置を特定するための情報とに少なくとも基づいて、前記ブロック行の符号データを復号する

ことを特徴とする画像復号装置。

【請求項 2】

前記復号手段は、

前記矩形領域におけるそれぞれのブロック行を並列に復号可能であることを特徴とする請求項 1 に記載の画像復号装置。

【請求項 3】

前記画像は、動画像における各フレームの画像もしくは静止画像であることを特徴とする請求項 1 又は 2 に記載の画像復号装置。

【請求項 4】

前記スライスは、ラスタ順に並ぶタイルを含むことを特徴とする請求項 1 乃至 3 の何れか 1 項に記載の画像復号装置。

【請求項 5】

前記矩形領域はブリックであることを特徴とする請求項 1 乃至 4 の何れか 1 項に記載の画像復号装置。

【請求項 6】

画像を複数のブロックから成るブロック行を 1 つ以上含む矩形領域に分割して符号化されたビットストリームから前記画像を復号する画像復号方法であって、
並列処理の有効化に関する `entropy_coding_sync_enabled_flag` を復号し、スライスが矩形形状となる第 1 モードが使用されるか当該第 1 モードと異なるモードであってスライスに関する第 2 モードが使用されるかを示す `rect_slice_flag` を前記ビットストリームのピクチャパラメータセットから復号し、前記画像における矩形領域の垂直方向のブロック数に対応する第 1 情報を前記ピクチャパラメータセットから復号し、前記画像におけるスライスに含まれる矩形領域の数の第 2 情報を前記ビットストリームのスライスヘッダから復号し、前記画像が複数の矩形領域を含み且つ前記 `rect_slice_flag` が前記第 2 モードが使用されることを示す場合においてスライスのアドレスを前記スライスヘッダから復号する復号工程と、
前記 `entropy_coding_sync_enabled_flag` の値が 1 であり、前記 `rect_slice_flag` が前記第 2 モードが使用されることを示す場合において、前記スライスのアドレスと、前記矩形領域の垂直方向におけるブロックの数に対応する前記第 1 情報と、前記スライスに含まれる矩形領域の数の前記第 2 情報と、に基づいて、前記スライスに対し、ブロック行の符号データの先頭位置を特定するための情報の数を特定する特定工程と

を備え、

前記復号工程において、前記特定工程によって特定した前記先頭位置を特定するための情報の数と、前記先頭位置を特定するための情報とに少なくとも基づいて、前記ブロック行の符号データを復号する

ことを特徴とする画像復号方法。

【請求項 7】

コンピュータに請求項 6 に記載の画像復号方法を実行させるためのコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像の復号技術に関するものである。

【背景技術】

【0002】

動画像の圧縮記録の符号化方式として、HEVC (High Efficiency V

10

20

30

40

50

ideo Coding) 符号化方式(以下、HEVCと記す)が知られている。HEVCでは符号化効率向上のため、従来のマクロブロック(16×16画素)より大きなサイズの基本ブロックが採用された。この大きなサイズの基本ブロックはCTU(Coding Tree Unit)と呼ばれ、そのサイズは最大64×64画素である。CTUはさらに予測や変換を行う単位となるサブブロックに分割される。

【0003】

またHEVCでは、ピクチャを複数のタイルまたはスライスに分割して符号化する事が可能である。各タイル間またはスライス間にはデータの依存性が少なく、並列に符号化・復号化処理を実施する事ができる。マルチコアのCPU等で並列に処理を実行し、処理時間を短縮できる事が、タイル、スライス分割の大きな利点の一つとして挙げられる。

10

【0004】

また、各スライスはHEVCに採用されている従来の2値算術符号化の手法によって符号化される。すなわち、各シンタックス要素が2値化され、2値信号が生成される。各シンタックス要素には、あらかじめ発生確率がテーブル(以下、発生確率テーブル)として与えられ、2値信号は発生確率テーブルに基づいて算術符号化される。この発生確率テーブルは復号時には復号情報として、続く符号の復号に使用される。符号化時には符号化情報として、続く符号化に使用される。そして符号化が行われる毎に、符号化された2値信号が発生確率の高い方のシンボルであったか否か、という統計情報に基づいて発生確率テーブルが更新される。

【0005】

20

またHEVCには、Wavefront Parallel Processing(以下、WPP)と呼ばれるエントロピー符号化・復号化を並列に処理するための手法がある。WPPでは、あらかじめ指定された位置のブロックを符号化処理した時点での発生確率のテーブルを、次の行の左端のブロックに適用することで、符号化効率の低下を抑制した上で行単位でのブロックの並列符号化処理が可能となる。ブロック行単位での並列処理を可能にする為、スライスヘッダにはビットストリーム中の各ブロック行の先頭位置を示すentry__point__offset__minus1及びその数を示すnum__entry__point__offsetsが符号化される。特許文献1では、WPPに関連する技術が開示されている。

【0006】

30

近年、HEVCの後継としてさらに高効率な符号化方式の国際標準化を行う活動が開始された。JVET(Joint Video Experts Team)がISO/IECとITU-Tの間で設立され、VVC(Versatile Video Coding)符号化方式(以下、VVC)として標準化が進められている。VVCでは、タイルを更に複数のブロック行から構成される矩形(ブリック)に分割する事が検討されている。そしてスライスは一つ以上のブリックを包含するように構成される。

【先行技術文献】

【特許文献】

【0007】

【文献】特開2014-11638号公報

40

【発明の概要】

【発明が解決しようとする課題】

【0008】

VVCにおいては、スライスを構成するブリックが予め導出可能であり、更にそのブリックに内包される基本ブロック行の数も他のシンタックスから導出可能である。そのため、該スライスに属する基本ブロック行の先頭位置を示すentry__point__offset__minus1の数を、num__entry__point__offsetを用いずに導出する事が可能である。そのためnum__entry__point__offsetは冗長なシンタックスとなる。本発明では、冗長なシンタックスを減らすことでビットストリームの符号量を減らす技術を提供する。

50

【課題を解決するための手段】

【0009】

本発明の一様態は、画像を複数のブロックから成るブロック行を1つ以上含む矩形領域に分割して符号化されたビットストリームから前記画像を復号する画像復号装置であって、並列処理の有効化に関する `entropy_coding_sync_enabled_flag` を復号し、スライスが矩形形状となる第1モードが使用されるか当該第1モードと異なるモードであってスライスに関する第2モードが使用されるかを示す `rect_slice_flag` を前記ビットストリームのピクチャパラメータセットから復号し、前記画像における矩形領域の垂直方向のブロック数に対応する第1情報を前記ピクチャパラメータセットから復号し、前記画像におけるスライスに含まれる矩形領域の数の第2情報を前記ビットストリームのスライスヘッダから復号し、前記画像が複数の矩形領域を含み且つ前記 `rect_slice_flag` が前記第2モードが使用されることを示す場合においてスライスのアドレスを前記スライスヘッダから復号する復号手段と、前記 `entropy_coding_sync_enabled_flag` の値が1であり、前記 `rect_slice_flag` が前記第2モードが使用されることを示す場合において、前記スライスのアドレスと、前記矩形領域の垂直方向におけるブロックの数に対応する前記第1情報と、前記スライスに含まれる矩形領域の数の前記第2情報と、に基づいて、前記スライスに対し、ブロック行の符号データの先頭位置を特定するための情報の数を特定する特定手段とを備え、前記復号手段は、前記特定手段によって特定した前記先頭位置を特定するための情報の数と、前記先頭位置を特定するための情報とに少なくとも基づいて、前記ブロック行の符号データを復号することを特徴とする。

10

20

【発明の効果】

【0010】

本発明の構成によれば、冗長なシンタクスを減らすことでビットストリームの符号量を減らすことができる。

【図面の簡単な説明】

【0011】

【図1】画像符号化装置の機能構成例を示すブロック図。

【図2】画像復号装置の機能構成例を示すブロック図。

【図3】画像符号化装置による入力画像の符号化処理のフローチャート。

30

【図4】画像復号装置によるビットストリームの復号処理のフローチャート。

【図5】コンピュータ装置のハードウェア構成例を示すブロック図。

【図6】ビットストリームのフォーマットの一例を示す図。

【図7】入力画像の分割例を示す図。

【図8】入力画像の分割例を示す図。

【図9】タイルとスライスとの関係を示す図。

【発明を実施するための形態】

【0012】

以下、添付図面を参照して実施形態を詳しく説明する。尚、以下の実施形態は特許請求の範囲に係る発明を限定するものでない。実施形態には複数の特徴が記載されているが、これらの複数の特徴の全てが発明に必須のものとは限らず、また、複数の特徴は任意に組み合わせられてもよい。さらに、添付図面においては、同一若しくは同様の構成に同一の参照番号を付し、重複した説明は省略する。

40

【0013】

〔第1の実施形態〕

先ず、本実施形態に係る画像符号化装置の機能構成例について、図1のブロック図を用いて説明する。画像分割部102には、符号化対象となる入力画像が入力される。入力画像は動画画像を構成する各フレームの画像であっても良いし、静止画像であっても良い。画像分割部102は、入力画像を「一つもしくは複数のタイル」に分割する。タイルは入力画像内の矩形領域を覆う、連続する基本ブロックの集合である。画像分割部102はさら

50

に、それぞれのタイルを一つもしくは複数のブリックに分割する。ブリックは、タイル内の一つもしくは複数の基本ブロックの行（基本ブロック行）で構成される矩形領域（タイル以下の大きさの複数のブロックから成るブロック行を1つ以上含む矩形領域）である。画像分割部102はさらに入力画像を、「一つまたは複数のタイル」あるいは「一つのタイル内の一つ以上のブリック」で構成されるスライスに分割する。スライスは符号化の基本単位であり、スライス毎にスライスの種類を示す情報等のヘッダ情報が付加される。入力画像を4個のタイル、4個のスライス、11個のブリックに分割する例を図7に示す。左上のタイルは1個のブリック、左下のタイルは2個のブリック、右上のタイルは5個のブリック、右下のタイルは3個のブリックにそれぞれ分割されている。そして左のスライスは3個のブリック、右上のスライスは2個のブリック、右中央のスライスは3個のブリック、右下のスライスは3個のブリックを包含するように構成されている。画像分割部102は、このようにして分割したタイル、ブリック、スライスのそれぞれについて、大きさに関する情報を分割情報として出力する。

10

【0014】

ブロック分割部103は、画像分割部102から出力された基本ブロック行の画像（基本ブロック行画像）を複数の基本ブロックに分割し、基本ブロック単位の画像（ブロック画像）を後段に出力する。

【0015】

予測部104は、基本ブロック単位の画像をサブブロックに分割し、サブブロック単位でフレーム内予測であるイントラ予測やフレーム間予測であるインター予測などを行い、予測画像を生成する。ブリックを跨いだイントラ予測（他のブリックのブロックの画素を用いたイントラ予測）、ブリックを跨いだ動きベクトルの予測（他のブリックのブロックの動きベクトルを用いた動きベクトルの予測）は行われない。さらに予測部104は、入力された画像と予測画像から予測誤差を算出して出力する。また予測部104は、予測に必要な情報（予測情報）、例えばサブブロック分割方法、予測モードや動きベクトル等の情報も予測誤差と併せて出力する。

20

【0016】

変換・量子化部105は、予測誤差をサブブロック単位で直交変換して変換係数を求め、該求めた変換係数を量子化して量子化係数を求める。逆量子化・逆変換部106は、変換・量子化部105から出力された量子化係数を逆量子化して変換係数を再生し、さらに該再生した変換係数を逆直交変換して予測誤差を再生する。

30

【0017】

フレームメモリ108は、再生された画像を格納しておくメモリとして機能する。画像再生部107は、予測部104から出力された予測情報に基づいてフレームメモリ108を適宜参照して予測画像を生成し、該予測画像と入力された予測誤差から再生画像を生成して出力する。

【0018】

インループフィルタ部109は、再生画像に対してデブロッキングフィルタやサンプルアダプティブオフセットなどのインループフィルタ処理を行い、該インループフィルタ処理が施された画像（フィルタ画像）を出力する。

40

【0019】

符号化部110は、変換・量子化部105から出力された量子化係数および予測部104から出力された予測情報を符号化することで符号データ（符号化データ）を生成し、該生成した符号データを出力する。

【0020】

統合符号化部111は、画像分割部102から出力された分割情報を用いてヘッダ符号データを生成し、該生成したヘッダ符号データと、符号化部110から出力された符号データと、を含むビットストリームを生成して出力する。制御部199は、画像符号化装置全体の動作制御を行うものであり、上記の画像符号化装置の各機能部の動作制御を行う。

【0021】

50

次に、図 1 に示した構成を有する画像符号化装置による入力画像に対する符号化処理について説明する。本実施形態では、説明を容易にするために、イントラ予測符号化の処理のみを説明するが、これに限定されずインター予測符号化の処理においても適用可能である。さらに本実施形態では具体的な説明を行うために、ブロック分割部 103 は、画像分割部 102 から出力された基本ブロック行画像を「 64×64 画素のサイズを有する基本ブロック」を単位に分割するものとして説明する。

【0022】

画像分割部 102 は入力画像をタイル、ブリック、スライスに分割する。画像分割部 102 による入力画像の分割例を図 8 に示す。本実施形態では図 8 (a) に示す如く、 1152×1152 画素のサイズを有する入力画像を 9 つのタイル (1 つのタイルのサイズは 384×384 画素) に分割する。各タイルには左上からラスト順に ID (タイル ID) が付与され、左上のタイルのタイル ID は 0、右下のタイルのタイル ID は 8 である。

【0023】

また、入力画像におけるタイル、ブリック、スライスの分割例を図 8 (b) に示す。図 8 (b) に示す如く、タイル ID = 0 のタイル及びタイル ID = 7 のタイルは 2 つのブリック (それぞれのブリックのサイズは 384×192 画素) に分割される。タイル ID = 2 のタイルは 2 つのブリック (上側のブリックのサイズは 384×128 画素、下側のブリックのサイズは 384×256 画素) に分割される。タイル ID = 3 のタイルは 3 つのブリック (それぞれのブリックのサイズは 384×128 画素) に分割される。タイル ID = 1, 4, 5, 6, 8 のタイルはブリックに分割しておらず (1 つのタイルを 1 つのブリックに分割することに等価)、その結果、タイル = ブリックとなっている。各ブリックにはラスト順のタイルの中で上から順に ID が付与される。図 8 (b) に示す BID がブリックの ID である。また入力画像は、スライス 0 ~ 4 に分割されている。スライス 0 は BID = 0 に対応するブリックを含むスライス、スライス 1 は BID = 1 に対応するブリックを含むスライス、スライス 2 は BID = 2 に対応するブリックを含むスライスである。スライス 3 は BID = 3 ~ 10 に対応するブリックを含むスライス、スライス 4 は BID = 11 ~ 13 に対応するブリックを含むスライスである。なお、各スライスにも、ラスト順のスライスの中で上から順に ID が付与されており、例えばスライス 0 は ID = 0 のスライス、スライス 4 は ID = 4 のスライスを指すものとする。

【0024】

そして画像分割部 102 は、分割したタイル、ブリック、スライスのそれぞれについて、大きさに関する情報を分割情報として統合符号化部 111 に出力する。また画像分割部 102 は、各ブリックを基本ブロック行画像に分割し、該分割した基本ブロック行画像をブロック分割部 103 に出力する。

【0025】

ブロック分割部 103 は、画像分割部 102 から出力された基本ブロック行画像を複数の基本ブロックに分割し、基本ブロック単位の画像であるブロック画像 (64×64 画素) を、後段の予測部 104 に対して出力する。

【0026】

予測部 104 は、基本ブロック単位の画像をサブブロックに分割し、サブブロック単位で水平予測や垂直予測などのイントラ予測モードを決定し、該決定したイントラ予測モードおよび符号化済の画素から予測画像を生成する。さらに予測部 104 は、入力された画像と予測画像から予測誤差を算出し、該算出した予測誤差を変換・量子化部 105 に対して出力する。また予測部 104 は、サブブロック分割方法やイントラ予測モードなどの情報を予測情報として、符号化部 110 および画像再生部 107 に対して出力する。

【0027】

変換・量子化部 105 は、予測部 104 から出力された予測誤差をサブブロック単位で直交変換 (サブブロックのサイズに対応した直交変換処理) して変換係数 (直交変換係数) を求める。そして変換・量子化部 105 は、該求めた変換係数を量子化して量子化係数を求める。そして変換・量子化部 105 は、該求めた量子化係数を符号化部 110 および

10

20

30

40

50

逆量子化・逆変換部 106 に対して出力する。

【0028】

逆量子化・逆変換部 106 は、変換・量子化部 105 から出力された量子化係数を逆量子化して変換係数を再生し、さらに該再生した変換係数を逆直交変換して予測誤差を再生する。そして逆量子化・逆変換部 106 は、該再生した予測誤差を画像再生部 107 に対して出力する。

【0029】

画像再生部 107 は、予測部 104 から出力された予測情報に基づいてフレームメモリ 108 を適宜参照して予測画像を生成し、該予測画像と、逆量子化・逆変換部 106 から入力された予測誤差と、から再生画像を生成する。そして画像再生部 107 は、該生成した再生画像をフレームメモリ 108 に格納する。

10

【0030】

インループフィルタ部 109 は、フレームメモリ 108 から再生画像を読み出し、該読み出した再生画像に対してデブロッキングフィルタやサンプルアダプティブオフセットなどのインループフィルタ処理を行う。そしてインループフィルタ部 109 は、該インループフィルタ処理が施された画像を再びフレームメモリ 108 に格納（再格納）する。

【0031】

符号化部 110 は、変換・量子化部 105 から出力された量子化係数および予測部 104 から出力された予測情報をエントロピー符号化することで符号データを生成する。エントロピー符号化の方法は特に指定しないが、ゴロム符号化、算術符号化、ハフマン符号化などを用いることができる。そして符号化部 110 は、該生成した符号データを統合符号化部 111 に対して出力する。

20

【0032】

統合符号化部 111 は、画像分割部 102 から出力された分割情報を用いてヘッダ符号データを生成し、該生成したヘッダ符号データと、符号化部 110 から出力された符号データと、を多重化することでビットストリームを生成して出力する。ビットストリームの出力先は特定の出力先に限るものではなく、画像符号化装置の内部若しくは外部のメモリに出力（格納）しても良いし、LAN やインターネットなどのネットワークを介して画像符号化装置と通信可能な外部装置に対して送信しても良い。

【0033】

30

次に、統合符号化部 111 が出力するビットストリーム（画像符号化装置が符号化する VVC による符号データ）のフォーマットの一例を図 6 に示す。図 6 のビットストリームには、シーケンスの符号化に関わる情報が含まれたヘッダ情報であるシーケンス・パラメータ・セット（SPS）が含まれている。また図 6 のビットストリームには、ピクチャの符号化に関わる情報が含まれたヘッダ情報であるピクチャ・パラメータ・セット（PPS）が含まれている。また図 6 のビットストリームには、スライスの符号化に関わる情報が含まれたヘッダ情報であるスライスヘッダ（SLH）が含まれている。また図 6 のビットストリームには、各ブリック（図 6 ではブリック 0 ～ブリック（N - 1））の符号データが含まれている。

【0034】

40

SPS には画像サイズ情報と基本ブロックデータ分割情報とが含まれている。PPS には、タイルの分割情報であるタイルデータ分割情報と、ブリックの分割情報であるブリックデータ分割情報と、スライスの分割情報であるスライスデータ分割情報 0 と、基本ブロック行データ同期化情報と、が含まれている。SLH には、スライスデータ分割情報 1 と基本ブロック行データ位置情報とが含まれている。

【0035】

まず、SPS について説明する。SPS には画像サイズ情報として情報 601 である `pic_width_in_luma_samples` 及び情報 602 である `pic_height_in_luma_samples` が含まれている。`pic_width_in_luma_samples` は入力画像の水平方向のサイズ（画素数）を表しており、p

50

`pic_height_in_luma_samples`は入力画像の垂直方向のサイズ(画素数)を表している。本実施形態では、入力画像として図8の入力画像を用いるため、`pic_width_in_luma_samples = 1152`、`pic_height_in_luma_samples = 1152`となる。またSPSには基本ブロックデータ分割情報として情報603である`log2_ctu_size_minus2`が含まれている。`log2_ctu_size_minus2`は、基本ブロックのサイズを表す。基本ブロックの垂直方向及び水平方向の画素数は $1 < (\log2_ctu_size_minus2 + 2)$ で示される。本実施形態では基本ブロックのサイズは 64×64 画素であるため、`log2_ctu_size_minus2`の値は4となる。

【0036】

次に、PPSについて説明する。PPSにはタイルデータ分割情報として情報604～607が含まれている。情報604は、入力画像が複数のタイルに分割されて符号化されているか否かを示す`single_tile_in_pic_flag`である。`single_tile_in_pic_flag = 1`の場合は、入力画像が複数のタイルに分割されて符号化されていないことを示す。一方、`single_tile_in_pic_flag = 0`の場合は、入力画像が複数のタイルに分割されて符号化されていることを示す。

【0037】

情報605は、`single_tile_in_pic_flag = 0`の場合にタイルデータ分割情報に含められる情報である。情報605は、各タイルが同一のサイズを持つか否かを示す`uniform_tile_spacing_flag`である。`uniform_tile_spacing_flag = 1`の場合は、各タイルが同一のサイズを持つことを示し、`uniform_tile_spacing_flag = 0`の場合は、サイズが同一ではないタイルが存在することを示す。

【0038】

情報606および情報607は、`uniform_tile_spacing_flag = 1`の場合にタイルデータ分割情報に含められる情報である。情報606は、(タイルの水平方向の基本ブロック数 - 1)を示す`tile_cols_width_minus1`である。情報607は、(タイルの垂直方向の基本ブロック数 - 1)を示す`tile_rows_height_minus1`である。入力画像の水平方向のタイル数は、入力画像の水平方向の基本ブロック数をタイルの水平方向の基本ブロック数で除算した場合の商として得られる。この除算により余りが生じた場合は、商に1を加えた数を「入力画像の水平方向のタイル数」とする。また入力画像の垂直方向のタイル数は、入力画像の垂直方向の基本ブロック数をタイルの垂直方向の基本ブロック数で除算した場合の商として得られる。この除算により余りが生じた場合は、商に1を加えた数を「入力画像の垂直方向のタイル数」とする。また、入力画像内の総タイル数は、入力画像の水平方向のタイル数 × 入力画像の垂直方向のタイル数を計算することで求めることができる。

【0039】

なお、`uniform_tile_spacing_flag = 0`の場合、他とサイズの異なるタイルが含まれているため、入力画像の水平方向におけるタイルの数、入力画像の垂直方向におけるタイルの数、各タイルの縦横のサイズ、を符号にする。

【0040】

またPPSにはブリックデータ分割情報として情報608～613が含まれている。情報608は、`brick_splitting_present_flag`である。`brick_splitting_present_flag = 1`の場合は、入力画像内における1つ以上のタイルが複数のブリックに分割されていることを示す。一方、`brick_splitting_present_flag = 0`の場合は、入力画像内におけるそれぞれのタイルが単一のブリックで構成されていることを示す。

【0041】

情報609は、`brick_splitting_present_flag = 1`の場合

10

20

30

40

50

合にブリックデータ分割情報に含められる情報である。情報 609 は、それぞれのタイルについて、該タイルが複数のブリックに分割されているか否かを示す `brick_split_flag[]` である。 i 番目のタイルが複数のブリックに分割されているかを示す `brick_split_flag[]` を `brick_split_flag[i]` と表記する。`brick_split_flag[i] = 1` の場合、 i 番目のタイルが複数のブリックに分割されていることを示し、`brick_split_flag[i] = 0` の場合、 i 番目のタイルが単一のブリックで構成されていることを示す。

【0042】

情報 610 は、`brick_split_flag[i] = 1` の場合に、 i 番目のタイルを構成するそれぞれのブリックのサイズが同一であるか否かを示す `uniform_brick_spacing_flag[i]` である。全ての i について `brick_split_flag[i] = 0` であれば、情報 610 はブリックデータ分割情報には含まれていない。情報 610 は、`brick_split_flag[i] = 1` を満たす i について、`uniform_brick_spacing_flag[i]` を含む。`uniform_brick_spacing_flag[i] = 1` の場合は、 i 番目のタイルを構成するそれぞれのブリックのサイズが同一であることを示す。一方、`uniform_brick_spacing_flag[i] = 0` の場合は、 i 番目のタイルを構成するブリックのうち他とサイズが異なるブリックが存在することを示す。

【0043】

情報 611 は、`uniform_brick_spacing_flag[i] = 1` の場合にブリックデータ分割情報に含められる情報である。情報 611 は、(i 番目のタイルにおけるブリックの垂直方向の基本ブロック数 - 1) を示す `brick_height_minus1[i]` である。

【0044】

なお、ブリックの垂直方向の基本ブロック数は、該ブリックの垂直方向の画素数を、基本ブロックの垂直方向の画素数（本実施形態では 64 画素）で除算することで求めることができる。また、タイルを構成するブリック数は、タイルの垂直方向の基本ブロック数をブリックの垂直方向の基本ブロック数で除算した場合の商として得られる。この除算により余りが生じた場合は、商に 1 を加えた数を「タイルを構成するブリック数」とする。例えばタイルの垂直方向の基本ブロック数が 10 で、`brick_height_minus1` の値が 2 であるとする。このとき、このタイルは上から順に、基本ブロック行数が 3 のブリック、基本ブロック行数が 3 のブリック、基本ブロック行数が 3 のブリック、基本ブロック行数が 1 のブリック、の 4 つのブリックに分割されることになる。

【0045】

情報 612 は、`uniform_brick_spacing_flag[i] = 0` を満たす i について (i 番目のタイルを構成するブリック数 - 1) を示す `num_brick_rows_minus1[i]` である。

【0046】

なお、本実施形態では、`uniform_brick_spacing_flag[i] = 0` の場合は、(i 番目のタイルを構成するブリック数 - 1) を示す `num_brick_rows_minus1[i]` をブリックデータ分割情報に含めた。しかし、これに限定されるものではない。

【0047】

例えば `brick_split_flag[i] = 1` の時点で i 番目のタイルを構成するブリックの数が 2 以上であると想定する。そして、(該タイルを構成するブリックの数 - 2) を示す `num_brick_rows_minus2[i]` を `num_brick_rows_minus1[i]` の代わりに符号化しても良い。このようにすることで、該タイルを構成するブリックの数を示すシンタクスのビット数を減らすことができる。例えば、該タイルが 2 個のブリックにより構成され、`num_brick_rows_minus1[i]` をゴロム符号化する場合、「1」を示す「010」の 3 ビットのデータが

符号化される。一方で、(該タイルを構成するブリックの数 - 2)を示す `num_brick_rows_minus 2[i]` をゴロム符号化する場合、0を示す「0」の1ビットのデータが符号化される。

【0048】

情報613は `uniform_brick_spacing_flag[i] = 0` を満たす *i* について (*i* 番目のタイルにおける *j* 番目のブリックの垂直方向の基本ブロック数 - 1)を示す `brick_row_height_minus 1[i][j]` である。`brick_row_height_minus 1[i][j]` は `num_brick_rows_minus 1[i]` の数だけ符号化される。なお、上述の `num_brick_rows_minus 2[i]` を用いた場合は、`brick_row_height_minus 1[i][j]` は `num_brick_rows_minus 2[i] + 1` の数だけ符号化される。タイルにおける下端のブリックの垂直方向の基本ブロック数は、該タイルの垂直方向の基本ブロック数から「`brick_row_height_minus 1 + 1`」の総和を減算することで求めることができる。例えば、タイルの垂直方向の基本ブロック数 = 10、`num_brick_rows_minus 1 = 3`、`brick_row_height_minus 1 = 2`、1、2であるとする。このとき、該タイルにおける下端のブリックの垂直方向の基本ブロック数は $10 - (3 + 2 + 3) = 2$ になる。

【0049】

またPPSにはスライスデータ分割情報0として情報614～618が含まれている。情報614は、`single_brick_per_slice_flag` である。`single_brick_per_slice_flag = 1` の場合、入力画像内の全てのスライスが単一のブリックで構成されていることを示す。つまり、それぞれのスライスが1つだけのブリックで構成されることを示す。一方、`single_brick_per_slice_flag = 0` の場合、入力画像において1つ以上のスライスが複数のブリックで構成されていることを示す。

【0050】

情報615は `rect_slice_flag` であり、`single_brick_per_slice_flag = 0` の場合にスライスデータ分割情報0に含められる情報である。`rect_slice_flag` は、スライスが含むタイルがラスタ順か矩形かを示す。図9(a)は、`rect_slice_flag = 0` の場合におけるタイルとスライスとの関係を示しており、スライス内のタイルがラスタ順で符号化されることを示している。一方、図9(b)は、`rect_slice_flag = 1` の場合におけるタイルとスライスとの関係を示しており、スライス内の複数のタイルが矩形形状であることを示している。

【0051】

情報616は `num_slices_in_pic_minus 1` であり、`rect_slice_flag = 1` かつ `single_brick_per_slice_flag = 0` の場合にスライスデータ分割情報0に含められる情報である。`num_slices_in_pic_minus 1` は、(入力画像におけるスライス数 - 1)を示す。

【0052】

情報617は、入力画像におけるそれぞれのスライスについて、該スライス (*i* 番目のスライス) の左上のブリックのインデックスを示す `top_left_brick_idx[i]` である。

【0053】

情報618は、入力画像中 *i* 番目のスライスにおける左上のブリックのインデックスと右下のブリックのインデックスとの差分を示す `bottom_right_brick_idx_delta[i]` である。ここで、「入力画像中 *i* 番目のスライスにおける左上のブリック」とは、該スライスにおいて最初に処理されるブリックである。また、「入力画像中 *i* 番目のスライスにおける右下のブリック」とは、該スライスにおいて最後に処理されるブリックである。ただし、フレーム内の最初のスライスの左上のブリックのインデ

10

20

30

40

50

ックスは0と決まっているため、最初のスライスの `top_left_brick_idx[0]` は符号化されない。

【0054】

またPPSには基本ブロック行データ同期化情報として情報619が符号化されて含まれている。情報619は `entropy_coding_sync_enabled_flag` である。`entropy_coding_sync_enabled_flag = 1` の場合は、上に隣接する基本ブロック行の所定位置の基本ブロックを処理した時点での発生確率のテーブルを左端のブロックに適用する。これにより、基本ブロック行単位でエントロピー符号化・復号化の並列処理が可能になる。

【0055】

次に、SLHについて説明する。SLHにはスライスデータ分割情報1として情報620～621が符号化されて含まれている。情報620は、`rect_slice_flag = 1` もしくは入力画像中のブリックの数が2以上の場合にスライスデータ分割情報1に含められる `slice_address` である。`slice_address` は、`rect_slice_flag = 0` の場合はスライスの先頭のBIDを示し、`rect_slice_flag = 1` の場合は、現在のスライスの番号を示す。

【0056】

情報621は、`rect_slice_flag = 0` かつ `single_brick_per_slice_flag = 0` の場合にスライスデータ分割情報1に含められる `num_bricks_in_slice_minus1` である。`num_bricks_in_slice_minus1` は(スライス中のブリック数 - 1)を示す。

【0057】

SLHには基本ブロック行データ位置情報として情報622が含まれている。情報622は `entry_point_offset_minus1[]` である。`entry_point_offset_minus1[]` は、`entropy_coding_sync_enabled_flag = 1` の場合に、(スライス中の基本ブロック行数 - 1)の数だけ基本ブロック行データ位置情報に符号化されて含められる。

【0058】

`entry_point_offset_minus1[]` は、基本ブロック行の符号データのエントリポイント、即ち基本ブロック行の符号データの先頭位置を表す。`entry_point_offset_minus1[j - 1]` はj番目の基本ブロック行の符号データのエントリポイントを示す。0番目の基本ブロック行の符号データの先頭位置は、該基本ブロック行が属するスライスの符号データの先頭位置と同じであるため省略される。そして、{(j - 1)番目の基本ブロック行の符号データの大きさ - 1}が `entry_point_offset_minus1[j - 1]` として符号化される。

【0059】

次に、本実施形態に画像符号化装置による入力画像の符号化処理(図6の構成を有するビットストリームの生成処理)について、図3のフローチャートに従って説明する。

【0060】

まず、ステップS301では、画像分割部102は入力画像をタイル、ブリック、スライスに分割する。そして画像分割部102は、分割したタイル、ブリック、スライスのそれぞれについて、大きさに関する情報を分割情報として統合符号化部111に出力する。また画像分割部102は、各ブリックを基本ブロック行画像に分割し、該分割した基本ブロック行画像をブロック分割部103に出力する。

【0061】

ステップS302では、ブロック分割部103は、基本ブロック行画像を複数の基本ブロックに分割し、基本ブロック単位の画像であるブロック画像を、後段の予測部104に対して出力する。

【0062】

ステップS303では、予測部104は、ブロック分割部103から出力された基本ブ

10

20

30

40

50

ロック単位の画像をサブブロックに分割し、サブブロック単位でイントラ予測モードを決定し、該決定したイントラ予測モードおよび符号化済の画素から予測画像を生成する。さらに予測部 104 は、入力された画像と予測画像から予測誤差を算出し、該算出した予測誤差を変換・量子化部 105 に対して出力する。また予測部 104 は、サブブロック分割方法やイントラ予測モードなどの情報を予測情報として、符号化部 110 および画像再生部 107 に対して出力する。

【0063】

ステップ S304 では、変換・量子化部 105 は、予測部 104 から出力された予測誤差をサブブロック単位で直交変換して変換係数（直交変換係数）を求める。そして変換・量子化部 105 は、該求めた変換係数を量子化して量子化係数を求める。そして変換・量子化部 105 は、該求めた量子化係数を符号化部 110 および逆量子化・逆変換部 106 に対して出力する。

10

【0064】

ステップ S305 では、逆量子化・逆変換部 106 は、変換・量子化部 105 から出力された量子化係数を逆量子化して変換係数を再生し、さらに該再生した変換係数を逆直交変換して予測誤差を再生する。そして逆量子化・逆変換部 106 は、該再生した予測誤差を画像再生部 107 に対して出力する。

【0065】

ステップ S306 では、画像再生部 107 は、予測部 104 から出力された予測情報に基づいてフレームメモリ 108 を適宜参照して予測画像を生成し、該予測画像と、逆量子化・逆変換部 106 から入力された予測誤差と、から再生画像を生成する。そして画像再生部 107 は、該生成した再生画像をフレームメモリ 108 に格納する。

20

【0066】

ステップ S307 では、符号化部 110 は、変換・量子化部 105 から出力された量子化係数および予測部 104 から出力された予測情報をエントロピー符号化することで符号データを生成する。

【0067】

ここで、`entropy_coding_sync_enabled_flag = 1` の場合、上に隣接する基本ブロック行の所定の位置の基本ブロックを処理した時点での発生確率テーブルを、次の基本ブロック行の左端の基本ブロックを処理する前に適用する。本実施形態では、`entropy_coding_sync_enabled_flag = 1` であるものとして説明する。

30

【0068】

ステップ S308 では、制御部 199 は、スライス内の全ての基本ブロックの符号化が完了したか否かを判断する。この判断の結果、スライス内の全ての基本ブロックの符号化が完了した場合には、処理はステップ S309 に進む。一方、スライス内の基本ブロックのうちまだ符号化していない基本ブロック（未符号化の基本ブロック）が残っている場合には、該未符号化の基本ブロックを符号化するべく、処理はステップ S303 に進む。

【0069】

ステップ S309 では、統合符号化部 111 は、画像分割部 102 から出力された分割情報を用いてヘッダ符号データを生成し、該生成したヘッダ符号データと、符号化部 110 から出力された符号データと、を含むビットストリームを生成して出力する。

40

【0070】

入力画像を図 8 に示す如く分割した場合、タイルデータ分割情報の、`single_tile_in_pic_flag` は 0、`uniform_tile_spacing_flag` は 1 になる。また `tile_cols_width_minus1` は 5、`tile_rows_height_minus1` は 5 である。

【0071】

ブリックデータ分割情報の `brick_splitting_present_flag` は 1 である。また、タイル ID = 1, 4, 5, 6, 8 に対応するタイルはブリックに分

50

割されていない。然るに `brick_split_flag[1]`、`brick_split_flag[4]`、`brick_split_flag[5]`、`brick_split_flag[6]`、`brick_split_flag[8]` は 0 である。またタイル ID = 0, 2, 3, 7 に対応するタイルはブリックに分割されている。然るに `brick_split_flag[0]`、`brick_split_flag[2]`、`brick_split_flag[3]`、`brick_split_flag[7]` は 1 である。
【0072】

また、タイル ID = 0, 3, 7 に対応するタイルは何れも同一サイズのブリックに分割されている。然るに `uniform_brick_spacing_flag[0]`、`uniform_brick_spacing_flag[3]`、`uniform_brick_spacing_flag[7]` は何れも 1 である。タイル ID = 2 に対応するタイルについては、`BID = 3` のブリックのサイズは、`BID = 4` のブリックのサイズとは異なる。然るに、`uniform_brick_spacing_flag[2]` は 0 である。

【0073】

`brick_height_minus1[0]` は 2 であり、`brick_height_minus1[3]` は 1 である。また、`brick_height_minus1[7]` は 2 である。なお、`brick_height_minus1` は `uniform_brick_spacing` が 1 の場合に符号化される。

【0074】

`brick_row_height_minus1[2][0]` は 1 である。なお、`num_brick_rows_minus1[2]` ではなく、代わりに上述の `num_brick_rows_minus2[2]` のシンタクスを符号化する場合は、値は 0 になる。

【0075】

また、本実施形態では、1つのスライスが複数のブリックを含むため、スライスデータ分割情報 0 における `single_brick_per_slice_flag` は 0 となる。また本実施形態では、スライスが複数のタイルをラスタ順で内包する（スライスはラスタ順に並ぶタイルを含む）ため、`rect_slice_flag` は 0 である。

【0076】

この場合、SLH におけるスライスデータ分割情報として、まずスライス中の最初の BID（該スライス中のブリックの BID のうち最小の BID）が `slice_address` として符号化される。

【0077】

スライス 0 については、該スライス 0 中の最初のブリックの BID である「0」が `slice_address` として符号化される。スライス 1 については、該スライス 1 中の最初のブリックの BID である「1」が `slice_address` として符号化される。スライス 2 については、該スライス 2 中の最初のブリックの BID である「2」が `slice_address` として符号化される。スライス 3 については、該スライス 3 中の最初のブリックの BID である「3」が `slice_address` として符号化される。スライス 4 については、該スライス 4 中の最初のブリックの BID である「11」が `slice_address` として符号化される。

【0078】

そして `num_bricks_in_slice_minus1` として各スライスがいくつのブリックを内包しているかが符号化される。スライス 0 に含まれるブリックは BID = 0 に対応するブリックのみ（単一のブリックのみ）であるから、スライス 0 に対応する `num_bricks_in_slice_minus1` は 0 となる。スライス 1 に含まれるブリックは BID = 1 に対応するブリックのみ（単一のブリックのみ）であるから、スライス 1 に対応する `num_bricks_in_slice_minus1` は 0 となる。スライス 2 に含まれるブリックは BID = 2 に対応するブリックのみ（単一のブリ

10

20

30

40

50

ックのみ)であるから、スライス2に対応するnum_brick_s_in_slice_minus1は0となる。スライス3に含まれるブリックはBID=3~10に対応するブリック(8個のブリック)であるから、スライス3に対応するnum_brick_s_in_slice_minus1は7となる。スライス4に含まれるブリックはBID=11~13に対応するブリック(3個のブリック)であるから、スライス4に対応するnum_brick_s_in_slice_minus1は2となる。

【0079】

次に、基本ブロック行データ位置情報は、符号化部110より送られた、(スライス中の(j-1)番目の基本ブロック行の符号データの大きさ-1)の値をentry_point_offset_minus1[j-1]として符号化する。スライス中のentry_point_offset_minus1の数は、(スライス中の基本ブロック行数-1)と等しい。

10

【0080】

スライス0は単一のブリックで構成されており(対応するnum_brick_s_in_slice_minus1=0)、brick_height_minus1[0]=2である。然るに、該ブリックの基本ブロック行の数は、brick_height_minus1[0]+1=3となる(jの範囲は0~1)。

【0081】

スライス1は単一のブリックで構成されており(対応するnum_brick_s_in_slice_minus1=0)、brick_height_minus1[0]=2、tile_rows_height_minus1=5である。然るに、該ブリックの基本ブロック行の数は、tile_rows_height_minus1-brick_height_minus1[0]=3となる(jの範囲は0~1)。

20

【0082】

スライス2は単一のブリックで構成されており(対応するnum_brick_s_in_slice_minus1=0)、brick_height_minus1[1]=5である。然るに、該ブリックの基本ブロック行の数は、brick_height_minus1[1]+1=6となる(jの範囲は0~4)。

【0083】

スライス3の場合、対応するslice_address=3、対応するnum_brick_s_in_slice_minus1=7である。然るに、スライス3は、BID=3~10(=slice_address+num_brick_s_in_slice_minus1)に対応するブリック(8個のブリック)で構成されている。スライス3における基本ブロック行の数は、BID=3~10のそれぞれに対応するブリックの基本ブロック行の数の合計値となる。

30

【0084】

BID=3に対応するブリックの基本ブロック行の数は、brick_row_height_minus1[2][0]=1であるから、brick_row_height_minus1[2][0]+1=2である。

【0085】

40

BID=4に対応するブリックの基本ブロック行の数は、次のようにして導出することができる。brick_row_height_minus1[2][0]=1、num_brick_rows_minus1[2]=1、tile_rows_height_minus1=5である。然るに、BID=4に対応するブリックの基本ブロック行の数は、tile_rows_height_minus1-brick_row_height_minus1[2][0]=4である。

【0086】

BID=5~7のそれぞれに対応するブリックの基本ブロック行の数は、brick_height_minus1[3]=1であることから、brick_height_minus1[3]+1=2であることが分かる。

50

【 0 0 8 7 】

B I D = 8 ~ 1 0 のそれぞれに対応するブリックの基本ブロック行の数は、 $t i l e _ r o w s _ h e i g h t _ m i n u s \ 1 = 5$ であることから、 $t i l e _ r o w s _ h e i g h t _ m i n u s \ 1 + 1 = 6$ であることがわかる。

【 0 0 8 8 】

よって、スライス3における基本ブロック行の数は、B I D = 3 ~ 1 0 のそれぞれに対応するブリックの基本ブロック行の数の合計値 ($2 + 4 + 2 + 2 + 2 + 6 + 6 + 6$) = 30と導出することができる。j の範囲は0 ~ 2 8 となる。

【 0 0 8 9 】

スライス4の場合、対応する $s l i c e _ a d d r e s s = 1 1$ 、対応する $n u m _ b r i c k s _ i n _ s l i c e _ m i n u s \ 1 = 2$ である。然るに、スライス4は、B I D = 1 1 ~ 1 3 (= $s l i c e _ a d d r e s s + n u m _ b r i c k s _ i n _ s l i c e _ m i n u s \ 1$) に対応するブリック (3個のブリック) で構成されている。スライス4における基本ブロック行の数は、B I D = 1 1 ~ 1 3 のそれぞれに対応するブリックの基本ブロック行の数の合計値となる。

10

【 0 0 9 0 】

B I D = 1 1 ~ 1 2 のそれぞれに対応するブリックの基本ブロック行の数は、 $b r i c k _ h e i g h t _ m i n u s \ 1 [7] = 2$ であることから、 $b r i c k _ h e i g h t _ m i n u s \ 1 [7] + 1 = 3$ であることが分かる。

【 0 0 9 1 】

20

B I D = 1 3 に対応するブリックの基本ブロック行の数は、 $t i l e _ r o w s _ h e i g h t _ m i n u s \ 1 = 5$ であることから、 $t i l e _ r o w s _ h e i g h t _ m i n u s \ 1 + 1 = 6$ であることがわかる。

【 0 0 9 2 】

よって、スライス4における基本ブロック行の数は、B I D = 1 1 ~ 1 3 のそれぞれに対応するブリックの基本ブロック行の数の合計値 ($3 + 3 + 6$) = 1 2 と導出することができる。j の範囲は0 ~ 1 0 となる。

【 0 0 9 3 】

このような処理により、各スライスにおける基本ブロック行数が確定する。本実施形態では、他のシンタクスから $e n t r y _ p o i n t _ o f f s e t _ m i n u s \ 1$ の数を導出することができるので、従来のように $n u m _ e n t r y _ p o i n t _ o f f s e t$ を符号化してヘッダに含める必要はない。よって、本実施形態によれば、ビットストリームのデータ量を削減することができる。

30

【 0 0 9 4 】

ステップ S 3 1 0 では、制御部 1 9 9 は、入力画像における全ての基本ブロックの符号化が完了したか否かを判断する。この判断の結果、入力画像における全ての基本ブロックの符号化が完了した場合には、処理はステップ S 3 1 1 に進む。一方、入力画像において未だ符号化されていない基本ブロックが残っている場合には、処理はステップ S 3 0 3 に進み、未だ符号化されていない基本ブロックについて以降の処理を行う。

【 0 0 9 5 】

40

ステップ S 3 1 1 では、インループフィルタ部 1 0 9 は、ステップ S 3 0 6 で生成された再生画像に対してインループフィルタ処理を行い、該インループフィルタ処理が施された画像を出力する。

【 0 0 9 6 】

このように本実施形態によれば、ブリックが有する基本ブロック行の符号データの先頭位置を示す情報がいくつ符号化されているのかを示す情報を符号化してビットストリームに含めなくても良く、該情報が導出可能なビットストリームを生成することができる。

【 0 0 9 7 】

[第 2 の実施形態]

本実施形態では、第 1 の実施形態に係る画像符号化装置によって生成されたビットスト

50

リームを復号する画像復号装置について説明する。なお、ビットストリームの構成など、第 1 の実施形態と共通する要件については第 1 の実施形態にて説明したとおりであるため、説明は省略する。

【0098】

本実施形態に係る画像復号装置の機能構成例について、図 2 のブロック図を用いて説明する。分離復号部 202 は、第 1 の実施形態に係る画像符号化装置によって生成されたビットストリームを取得する。ビットストリームの取得方法は特定の取得方法に限らない。例えば、LAN やインターネットなどのネットワークを介して画像符号化装置から直接的もしくは間接的にビットストリームを取得しても良いし、画像復号装置の内部もしくは外部に保存しておいたビットストリームを取得しても良い。そして分離復号部 202 は、該取得したビットストリームから、復号処理に関する情報や係数に関する符号データを分離し、復号部 203 へ送る。また分離復号部 202 は、ビットストリームのヘッダの符号データを復号する。本実施形態では、タイル、ブリック、スライス、基本ブロックの大きさ等の画像の分割に関するヘッダ情報を復号して分割情報を生成し、該生成した分割情報を画像再生部 205 に出力する。つまり分離復号部 202 は、図 1 の統合符号化部 111 と逆の動作を行う。

【0099】

復号部 203 は、分離復号部 202 から出力された符号データを復号して量子化係数および予測情報を再生する。逆量子化・逆変換部 204 は、量子化係数に対して逆量子化を行って変換係数を生成し、該生成した変換係数に対して逆直交変換を行うことで予測誤差を再生する。

【0100】

フレームメモリ 206 は、再生されたピクチャの画像データを格納するためのメモリである。画像再生部 205 は、入力された予測情報に基づいてフレームメモリ 206 を適宜参照して予測画像を生成する。そして画像再生部 205 は、該生成した予測画像と逆量子化・逆変換部 204 で再生された予測誤差から再生画像を生成する。そして画像再生部 205 は、再生画像について、分離復号部 202 から入力された分割情報に基づいてタイル、ブリック、スライスの入力画像中の位置を特定して出力する。

【0101】

インループフィルタ部 207 は、上記のインループフィルタ部 109 と同様、再生画像に対してデブロッキングフィルタなどのインループフィルタ処理を行い、インループフィルタ処理が施された画像を出力する。制御部 299 は、画像復号装置全体の動作制御を行うものであり、上記の画像復号装置の各機能部の動作制御を行う。

【0102】

次に、図 2 に示した構成を有する画像復号装置によるビットストリームの復号処理について説明する。以下では、ビットストリームをフレーム単位で画像復号装置に入力するものとして説明するが、1 フレーム分の静止画像のビットストリームを画像復号装置に入力するようにしても良い。また、本実施形態では説明を容易にするために、イントラ予測復号処理のみを説明するが、これに限定されずインター予測復号処理においても適用可能である。

【0103】

分離復号部 202 は、入力されたビットストリームから、復号処理に関する情報や係数に関する符号データを分離し、復号部 203 へ送る。また分離復号部 202 は、ビットストリームのヘッダの符号データを復号する。より具体的には分離復号部 202 は、図 6 における基本ブロックデータ分割情報、タイルデータ分割情報、ブリックデータ分割情報、スライスデータ分割情報 0、基本ブロック行データ同期化情報、基本ブロック行データ位置情報等を復号して分割情報を生成する。そして分離復号部 202 は、該生成した分割情報を画像再生部 205 に出力する。また分離復号部 202 は、ピクチャデータの基本ブロック単位の符号データを再生し、復号部 203 に出力する。

【0104】

復号部 203 は、分離復号部 202 から出力された符号データを復号して量子化係数および予測情報を再生する。再生された量子化係数は逆量子化・逆変換部 204 に出力され、再生された予測情報は画像再生部 205 に出力される。

【0105】

逆量子化・逆変換部 204 は、入力された量子化係数に対して逆量子化を行って変換係数を生成し、該生成した変換係数に対して逆直交変換を行うことで予測誤差を再生する。再生された予測誤差は画像再生部 205 に出力される。

【0106】

画像再生部 205 は、分離復号部 202 から入力された予測情報に基づいてフレームメモリ 206 を適宜参照して予測画像を生成する。そして画像再生部 205 は、該生成した予測画像と逆量子化・逆変換部 204 で再生された予測誤差から再生画像を生成する。そして画像再生部 205 は、再生画像について、分離復号部 202 から入力された分割情報に基づいて、例えば図 7 のようなタイル、ブリック、スライスの形状および入力画像中の位置を特定してフレームメモリ 206 に出力（格納）する。フレームメモリ 206 に格納された画像は予測の際の参照に用いられる。

【0107】

インループフィルタ部 207 は、フレームメモリ 206 から読み出した再生画像に対してデブロッキングフィルタなどのインループフィルタ処理を行い、インループフィルタ処理が施された画像をフレームメモリ 206 に出力（格納）する。

【0108】

制御部 299 は、フレームメモリ 206 に格納された再生画像を出力する。再生画像の出力先は特定の出力先に限らない。例えば制御部 299 は、画像復号装置が有する表示装置に該再生画像を出力して該表示装置に該再生画像を表示させても良い。また例えば制御部 299 は、LAN やインターネットなどのネットワークを介して再生画像を外部の装置に対して送信しても良い。

【0109】

次に、本実施形態に係る画像復号装置によるビットストリームの復号処理（図 6 の構成を有するビットストリームの復号処理）について、図 4 のフローチャートに従って説明する。

【0110】

ステップ S401 では、分離復号部 202 は、入力されたビットストリームから、復号処理に関する情報や係数に関する符号データを分離し、復号部 203 へ送る。また分離復号部 202 は、ビットストリームのヘッダの符号データを復号する。より具体的には、分離復号部 202 は、図 6 における基本ブロックデータ分割情報、タイルデータ分割情報、ブリックデータ分割情報、スライスデータ分割情報、基本ブロック行データ同期化情報、基本ブロック行データ位置情報等を復号して分割情報を生成する。そして分離復号部 202 は、該生成した分割情報を画像再生部 205 に出力する。また分離復号部 202 は、ピクチャデータの基本ブロック単位の符号データを再生し、復号部 203 に出力する。

【0111】

本実施形態では、ビットストリームの符号化元である入力画像の分割は図 8 に示した分割とする。ビットストリームの符号化元である入力画像やその分割に係る情報は分割情報から導出することができる。

【0112】

画像サイズ情報に含まれている `pic_width_in_luma_samples` から、入力画像の水平方向のサイズ（横サイズ）が 1152 画素であることを特定することができる。また、画像サイズ情報に含まれている `pic_height_in_luma_samples` から、入力画像の垂直方向のサイズ（縦サイズ）が 1152 画素であることを特定することができる。

【0113】

また、基本ブロックデータ分割情報の `log2_ctu_size_minus2 = 4`

10

20

30

40

50

であることから、基本ブロックのサイズを、 $1 < \log_2 \text{ctu_size_minus} 2 + 2$ より、 64×64 画素と導出することができる。

【0114】

また、タイルデータ分割情報の `single_tile_in_pic_flag = 0` であることから、入力画像が複数のタイルに分割されていることを特定することができる。そして、`uniform_tile_spacing_flag = 1` であることから、各タイルが（端部を除いて）同一のサイズを有することを特定することができる。

【0115】

また、`tile_cols_width_minus 1 = 5`、`tile_rows_height_minus 1 = 5` であることから、各タイルが 6×6 個の基本ブロックで構成されていることを特定することができる。つまり、各タイルが 384×384 画素で構成されていることを特定することができる。入力画像が 1152×1152 画素であることから、該入力画像は水平方向に3つ、垂直方向に3つの9つのタイルに分割されて符号化されていることがわかる。

10

【0116】

また、ブリックデータ分割情報の `brick_splitting_present_flag = 1` であることから、入力画像中の少なくとも一つのタイルが複数のブリックに分割されていることを特定することができる。

【0117】

また、`brick_split_flag[1]`、`brick_split_flag[4]`、`brick_split_flag[5]`、`brick_split_flag[6]`、`brick_split_flag[8]` は0である。これにより、タイルID = 1, 4, 5, 6, 8 に対応するタイルはブリックに分割されていないことを特定することができる。本実施形態では、すべてのタイルの基本ブロック行数が6であることから、タイルID = 1, 4, 5, 6, 8 に対応するタイルのブリックの基本ブロック行数は6であることがわかる。

20

【0118】

一方で、`brick_split_flag[0]`、`brick_split_flag[2]`、`brick_split_flag[3]`、`brick_split_flag[7]` は1である。これにより、タイルID = 0, 2, 3, 7 に対応するタイルはブリックに分割されていることを特定することができる。また、`uniform_brick_spacing_flag[0]`、`uniform_brick_spacing_flag[3]`、`uniform_brick_spacing_flag[7]` は何れも1である。これにより、タイルID = 0, 3, 7 に対応するタイルは何れも同一サイズのブリックに分割されていることを特定することができる。

30

【0119】

また、`brick_height_minus 1[0]`、`brick_height_minus 1[7]` は何れも2である。然るに、タイルID = 0 に対応するタイル及びタイルID = 7 に対応するタイルは何れも、タイル内のブリックの垂直方向の基本ブロック数が3であることを特定することができる。また、タイルID = 0 に対応するタイル及びタイルID = 7 に対応するタイルは何れも、該タイル内のブリック数が2（＝タイルの基本ブロック行数（6）／タイル内のブリックの垂直方向の基本ブロック数（3））であることを特定することができる。

40

【0120】

また、`brick_height_minus 1[3]` は1である。然るに、タイルID = 3 に対応するタイル内のブリックの垂直方向の基本ブロック数が2であることを特定することができる。また、タイルID = 3 に対応するタイル内のブリック数が3（＝タイルの基本ブロック行数（6）／タイル内のブリックの垂直方向の基本ブロック数（2））であることを特定することができる。

【0121】

50

タイルID = 2に対応するタイルについては、`num_brick_rows_minus1[2] = 1`であることから、該タイルは2つのブリックで構成されていることを特定することができる。また、`uniform_brick_spacing_flag[2] = 0`である。これにより、タイルID = 2に対応するタイルは、他とサイズが異なるブリックが存在することを特定することができる。そして、`brick_row_height_minus1[2][0] = 1`、`brick_row_height_minus1[2][1] = 3`、全てのタイルの垂直方向の基本ブロック数は6である。これにより、タイルID = 2に対応するタイルは上から順に、垂直方向の基本ブロック数が2のブリック、垂直方向の基本ブロック数が4のブリック、で構成されていることを特定することができる。なお、`brick_row_height_minus1[2][1] = 3`は符号化しないようにしてもよい。タイルにおけるブリックの数が2つの場合、タイルの高さと、タイル内の1つ目のブリックの高さ(`brick_row_height_minus1[2][0] = 1`)とから2つ目のブリックの高さを求めることが可能であるためである。

10

【0122】

また、スライスデータ分割情報0における`single_brick_per_slice_flag = 0`であることから、少なくとも1つのスライスが複数のブリックで構成されていることを特定することができる。本実施形態では、`uniform_brick_spacing_flag[i] = 0`の場合は、(*i*番目のタイルを構成するブリック数 - 1)を示す`num_brick_rows_minus1[i]`をブリックデータ分割情報に含めた。しかし、これに限定されるものではない。

20

【0123】

例えば`brick_split_flag[i] = 1`の時点で*i*番目のタイルを構成するブリックの数が2以上であると想定する。そして、(該タイルを構成するブリックの数 - 2)を示す`num_brick_rows_minus2[i]`を`num_brick_rows_minus1[i]`の代わりに復号しても良い。このようにすることで、該タイルを構成するブリックの数を示すシンタックスのビット数を減らしたビットストリームを復号することができる。

【0124】

次に、各ブリックの左上及び右下の境界の座標を求める。座標は入力画像の左上を原点とし、基本ブロックの水平位置と垂直位置で示される。例えば、左から3番目、上から2番目の基本ブロックの左上の境界の座標は(3, 2)、右下の境界の座標は(4, 3)になる。

30

【0125】

タイルID = 0に対応するタイル内のBID = 0のブリックの左上の境界の座標は(0, 0)である。BID = 0のブリックの基本ブロック行数は3で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(3, 3)になる。

【0126】

タイルID = 0に対応するタイル内のBID = 1のブリックの左上の境界の座標は(0, 3)である。BID = 1のブリックの基本ブロック行数は3で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(6, 6)になる。

40

【0127】

タイルID = 1に対応するタイル(BID = 2のブリック)の左上の境界の座標は(6, 0)である。BID = 2のブリックの基本ブロック行数は6で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(12, 6)になる。

【0128】

タイルID = 2に対応するタイル内のBID = 3のブリックの左上の境界の座標は(12, 0)である。BID = 3のブリックの基本ブロック行数は2であるから、右下の境界の座標は(18, 2)になる。

【0129】

50

タイルID = 2に対応するタイル内のBID = 4のブリックの左上の境界の座標は(12, 2)である。BID = 4のブリックの基本ブロック行数は4であるから、右下の境界の座標は(18, 6)になる。

【0130】

タイルID = 3に対応するタイル内のBID = 5のブリックの左上の境界の座標は(0, 6)である。BID = 5のブリックの基本ブロック行数は2であるから、右下の境界の座標は(6, 8)になる。

【0131】

タイルID = 3に対応するタイル内のBID = 6のブリックの左上の境界の座標は(0, 8)である。BID = 6のブリックの基本ブロック行数は2であるから、右下の境界の座標は(6, 10)になる。

10

【0132】

タイルID = 3に対応するタイル内のBID = 7のブリックの左上の境界の座標は(0, 10)である。BID = 7のブリックの基本ブロック行数は2であるから、右下の境界の座標は(6, 12)になる。

【0133】

タイルID = 4に対応するタイル(BID = 8のブリック)の左上の境界の座標は(6, 6)である。BID = 8のブリックの基本ブロック行数は6で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(12, 12)になる。

【0134】

20

タイルID = 5に対応するタイル(BID = 9のブリック)の左上の境界の座標は(12, 6)である。BID = 9のブリックの基本ブロック行数は6で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(18, 12)になる。

【0135】

タイルID = 6に対応するタイル(BID = 10のブリック)の左上の境界の座標は(0, 12)である。BID = 10のブリックの基本ブロック行数は6で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(6, 18)になる。

【0136】

タイルID = 7に対応するタイル内のBID = 11のブリックの左上の境界の座標は(6, 12)である。BID = 11のブリックの基本ブロック行数は3で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(12, 15)になる。

30

【0137】

タイルID = 7に対応するタイル内のBID = 12のブリックの左上の境界の座標は(6, 15)である。BID = 12のブリックの基本ブロック行数は3で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(12, 18)になる。

【0138】

タイルID = 8に対応するタイル(BID = 13のブリック)の左上の境界の座標は(12, 12)である。BID = 13のブリックの基本ブロック行数は6で、全てのタイルの水平方向の基本ブロック数は6であるから、右下の境界の座標は(18, 18)になる。

【0139】

40

次に、各スライスに含まれるブリックを特定する。スライス0については、slice__addressは0、num__bricks__in__slice__minus1は0であるから、スライス0はBID = 0のブリックのみで構成されていることが分かる。

【0140】

スライス1については、slice__addressは1、num__bricks__in__slice__minus1は0であるから、スライス1はBID = 1のブリックのみで構成されていることが分かる。

【0141】

スライス2については、slice__addressは2、num__bricks__in__slice__minus1は0であるから、スライス2はBID = 2のブリックのみ

50

で構成されていることが分かる。

【0142】

スライス3については、`slice__address`は3、`num__bricks__in__slice__minus1`は7である。然るに、スライス3は $BID = 3 \sim 10 (= slice_address + num_bricks_in_slice_minus1)$ のブリックで構成されていることが分かる。

【0143】

スライス4については、`slice__address`は11、`num__bricks__in__slice__minus1`は2である。然るに、スライス4は $BID = 11 \sim 13 (= slice_address + num_bricks_in_slice_minus1)$ のブリックで構成されていることが分かる。

10

【0144】

また、`entropy__coding__sync__enabled__flag`が1である。よって、(スライス中の $(j - 1)$ 番目の基本ブロック行の符号データの大きさ - 1) が`entry__point__offset__minus1[j - 1]`として符号化されていることが分かる。その個数は、(処理するスライスの基本ブロック行数 - 1)である。

【0145】

スライス0については、該スライス0を構成する $BID = 0$ のブリックの基本ブロック行数 = 3であるから、 $3 - 1 = 2$ となり、 j の範囲は $0 \sim 1$ となる。スライス1については、該スライス1を構成する $BID = 1$ のブリックの基本ブロック行数 = 3であるから、 $3 - 1 = 2$ となり、 j の範囲は $0 \sim 1$ となる。スライス2については、該スライス2を構成する $BID = 2$ のブリックの基本ブロック行数 = 6であるから、 $6 - 1 = 5$ となり、 j の範囲は $0 \sim 4$ となる。スライス3については、該スライス3を構成する $BID = 3 \sim 10$ のブリックの基本ブロック行数の合計値は $2 + 4 + 2 + 2 + 2 + 6 + 6 + 6 = 30$ であるから、 j の範囲は $0 \sim 28$ となる。

20

【0146】

スライス4については、該スライス4を構成する $BID = 11 \sim 13$ のブリックの基本ブロック行数の合計値は $3 + 3 + 6 = 12$ であるから、 j の範囲は $0 \sim 10$ となる。

【0147】

以上により、従来のように`num__entry__point__offset`を符号化せずとも、他のシンタクスから`entry__point__offset__minus1`の数を導出できる。そして各基本ブロック行のデータの先頭位置がわかるため、基本ブロック行毎に並列して復号処理を行うことができる。分離復号部202より導出された分割情報は画像再生部205に送られ、ステップS404で入力画像内における処理対象の位置の特定に用いられる。

30

【0148】

ステップS402では、復号部203は、分離復号部202で分離された符号データを復号して量子化係数および予測情報を再生する。ステップS403では、逆量子化・逆変換部204は、入力された量子化係数に対して逆量子化を行って変換係数を生成し、該生成した変換係数に対して逆直交変換を行うことで予測誤差を再生する。

40

【0149】

ステップS404では、画像再生部205は、復号部203から入力された予測情報に基づいてフレームメモリ206を適宜参照して予測画像を生成する。そして画像再生部205は、該生成した予測画像と逆量子化・逆変換部204で再生された予測誤差から再生画像を生成する。そして画像再生部205は、再生画像について、分離復号部202から入力された分割情報に基づいて、タイル、ブリック、スライスの入力画像中の位置を特定してその位置に合成し、フレームメモリ206に出力(格納)する。

【0150】

ステップS405では、制御部299は、入力画像の全ての基本ブロックを復号したか否かを判断する。この判断の結果、入力画像の全ての基本ブロックを復号した場合には、

50

処理はステップ S 4 0 6 に進む。一方、入力画像において未だ復号していない基本ブロックが残っている場合には、処理はステップ S 4 0 2 に進み、未だ復号していない基本ブロックについて復号処理を行う。

【 0 1 5 1 】

ステップ S 4 0 6 では、インループフィルタ部 2 0 7 は、フレームメモリ 2 0 6 から読み出した再生画像に対してインループフィルタ処理を行い、インループフィルタ処理が施された画像をフレームメモリ 2 0 6 に出力（格納）する。

【 0 1 5 2 】

このように本実施形態によれば、第 1 の実施形態に係る画像符号化装置によって生成された「ブリックが有する基本ブロック行の先頭位置を示す情報がいくつ符号化されているのかを示す情報を含まないビットストリーム」から入力画像を復号することができる。

10

【 0 1 5 3 】

なお、第 1 の実施形態に係る画像符号化装置と第 2 の実施形態に係る画像復号装置とは別個の装置であっても良い。また、第 1 の実施形態に係る画像符号化装置と第 2 の実施形態に係る画像復号装置とを 1 つの装置に統合しても良い。

【 0 1 5 4 】

[第 3 の実施形態]

図 1 や図 2 に示した各機能部はハードウェアで実装しても良いが、一部をソフトウェアで実装しても良い。後者の場合、フレームメモリ 1 0 8 やフレームメモリ 2 0 6 を除く各機能部をソフトウェア（コンピュータプログラム）で実装しても良い。このようなコンピュータプログラムを実行可能なコンピュータ装置は、上記の画像符号化装置や画像復号装置に適用可能である。

20

【 0 1 5 5 】

上記の画像符号化装置や画像復号装置に適用可能なコンピュータ装置のハードウェア構成例について、図 5 のブロック図を用いて説明する。なお、図 5 に示したハードウェア構成は、上記の画像符号化装置や画像復号装置に適用可能なコンピュータ装置のハードウェア構成の一例に過ぎず、適宜変更 / 変形が可能である。

【 0 1 5 6 】

C P U 5 0 1 は R A M 5 0 2 や R O M 5 0 3 に格納されているコンピュータプログラムやデータを用いて各種の処理を実行する。これにより C P U 5 0 1 はコンピュータ装置全体の動作制御を行うと共に、上記の画像符号化装置や画像復号装置が行うものとして説明した各処理を実行若しくは制御する。すなわち C P U 5 0 1 は、図 1 や図 2 に示した各機能部（フレームメモリ 1 0 8 やフレームメモリ 2 0 6 を除く）として機能することができる。

30

【 0 1 5 7 】

R A M 5 0 2 は、R O M 5 0 3 や外部記憶装置 5 0 6 からロードされたコンピュータプログラムやデータを格納するためのエリア、I / F 5 0 7 を介して外部から受信したデータを格納するためのエリアを有する。また R A M 5 0 2 は、C P U 5 0 1 が各種の処理を実行する際に用いるワークエリアを有する。このように R A M 5 0 2 は、各種のエリアを適宜提供することができる。R O M 5 0 3 には、コンピュータ装置の設定データや起動プログラムなどが格納されている。

40

【 0 1 5 8 】

操作部 5 0 4 は、キーボード、マウス、タッチパネル画面などのユーザインターフェースであり、ユーザが操作することで各種の指示を C P U 5 0 1 に対して入力することができる。

【 0 1 5 9 】

表示部 5 0 5 は、液晶画面やタッチパネル画面などにより構成されており、C P U 5 0 1 による処理結果を画像や文字などでもって表示することができる。なお、表示部 5 0 5 は、画像や文字を投影するプロジェクタなどの装置であっても良い。

【 0 1 6 0 】

50

外部記憶装置 506 は、ハードディスクドライブ装置などの大容量情報記憶装置である。外部記憶装置 506 には、OS（オペレーティングシステム）や、上記の画像符号化装置や画像復号装置が行うものとして上述した各処理を CPU 501 に実行若しくは制御させるためのコンピュータプログラムやデータが保存されている。

【0161】

外部記憶装置 506 に保存されているコンピュータプログラムには、図 1 や図 2 においてフレームメモリ 108 やフレームメモリ 206 を除く各機能部の機能を CPU 501 に実行若しくは制御させるためのコンピュータプログラムが含まれている。また、外部記憶装置 506 に保存されているデータには、上記の説明において既知の情報として説明したものや、符号化や復号に関連する様々な情報が含まれている。

10

【0162】

外部記憶装置 506 に保存されているコンピュータプログラムやデータは、CPU 501 による制御に従って適宜 RAM 502 にロードされ、CPU 501 による処理対象となる。

【0163】

図 1 の画像符号化装置が有するフレームメモリ 108 や図 2 の画像符号化装置が有するフレームメモリ 206 は、上記の RAM 502 や外部記憶装置 506 などのメモリ装置を用いて実装可能である。

【0164】

I/F 507 は、外部の装置との間のデータ通信を行うためのインターフェースである。例えば、コンピュータ装置を画像符号化装置に適用した場合、画像符号化装置は生成したビットストリームを I/F 507 を介して外部に出力することができる。また、コンピュータ装置を画像復号装置に適用した場合、画像復号装置はビットストリームを I/F 507 を介して受信することができる。また画像復号装置は、ビットストリームを復号した結果を I/F 507 を介して外部に送信することができる。CPU 501、RAM 502、ROM 503、操作部 504、表示部 505、外部記憶装置 506、I/F 507 は何れもバス 508 に接続されている。

20

【0165】

なお、上記の説明において使用した具体的な数値は、具体的な説明を行うために使用したものであって、上記の各実施形態がこれらの数値に限定されることを意図したものではない。また、以上説明した各実施形態の一部若しくは全部を適宜組み合わせても構わない。また、以上説明した各実施形態の一部若しくは全部を選択的に用いても構わない。

30

【0166】

（その他の実施形態）

本発明は、上述の実施形態の 1 以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける 1 つ以上のプロセッサがプログラムを読出し実行する処理でも実現可能である。また、1 以上の機能を実現する回路（例えば、ASIC）によっても実現可能である。

【0167】

発明は上記実施形態に制限されるものではなく、発明の精神及び範囲から離脱することなく、様々な変更及び変形が可能である。従って、発明の範囲を公にするために請求項を添付する。

40

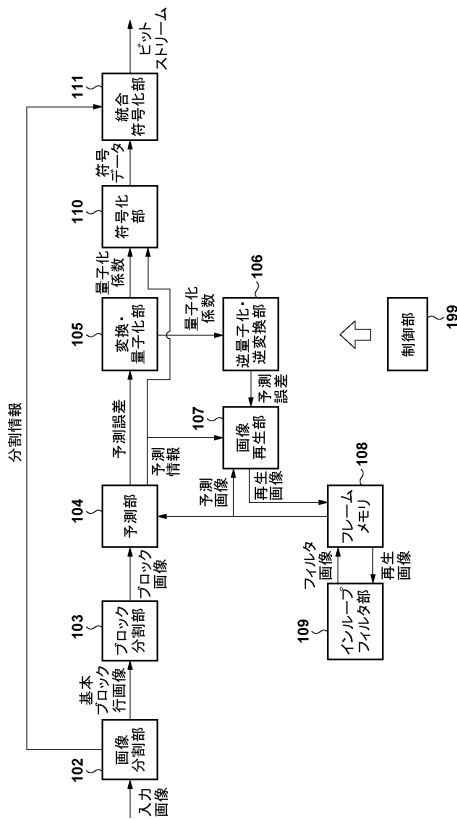
【符号の説明】

【0168】

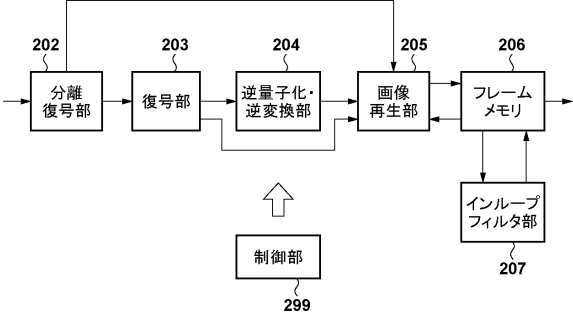
102：画像分割部 103：ブロック分割部 104：予測部 105：変換・量子化部 106：逆量子化・逆変換部 107：画像再生部 108：フレームメモリ 109：インループフィルタ部 110：符号化部 111：統合符号化部

【図面】

【図 1】



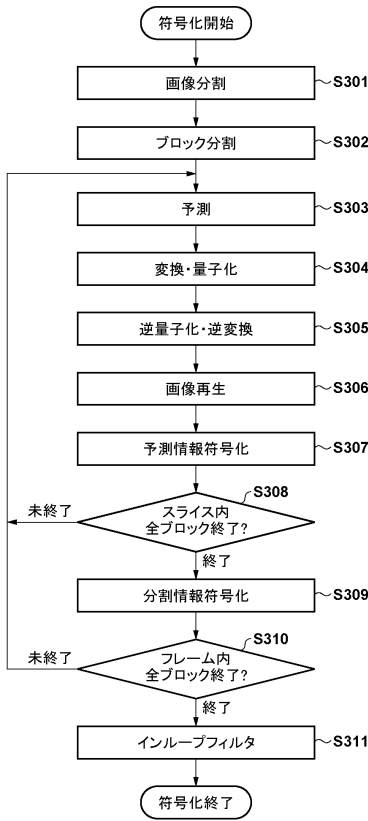
【図 2】



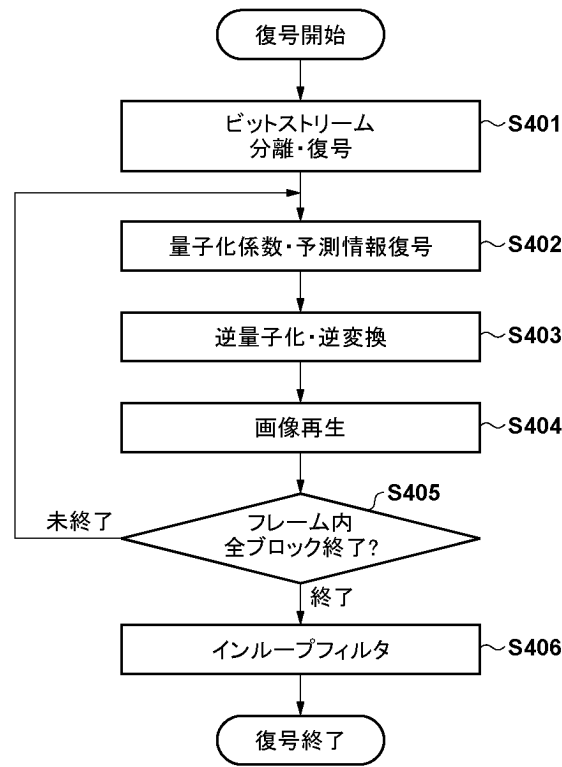
10

20

【図 3】



【図 4】

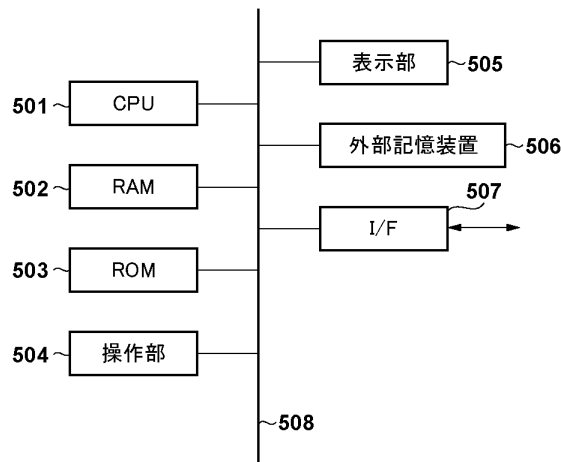


30

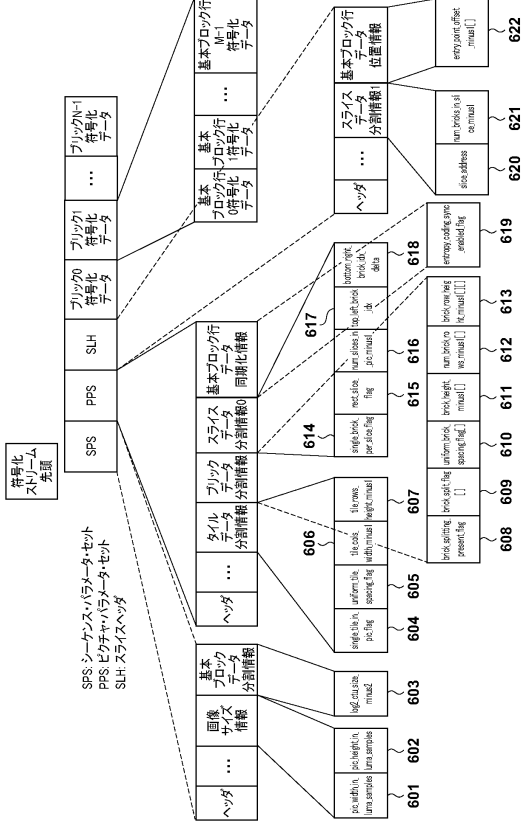
40

50

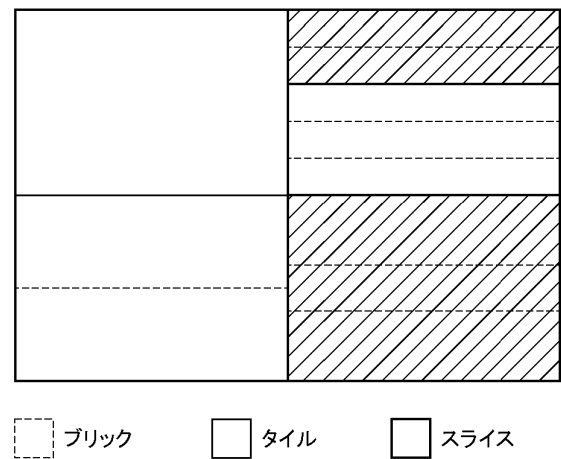
【図 5】



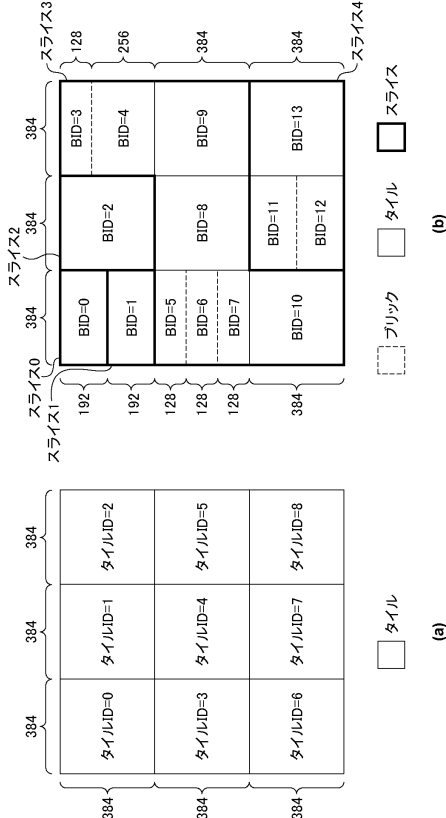
【図 6】



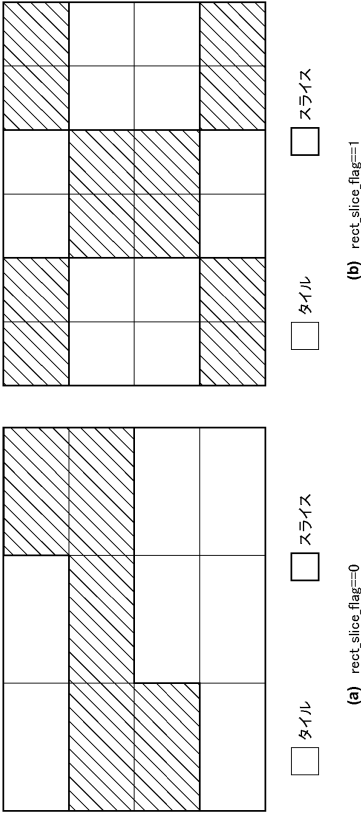
【図 7】



【図 8】



【図 9】



10

20

30

40

50

 フロントページの続き

- (56)参考文献 国際公開第 2 0 1 9 / 0 7 8 1 6 9 (W O , A 1)
 BROSS, Benjamin et al. , Versatile Video Coding (Draft 5) , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19-27 Mar. 2019, [JVET-N1001-v8] , JVET-N1001 (version 8) , ITU-T , 2019年06月11日 , < URL:https://jvet-experts.org/doc_end_user/documents/14_Geneva/wg11/JVET-N1001-v8.zip > : JVET-N1001-v8.docx: pp.18-20,35-37,40-43,81-85,89-97
 WANG, Ye-Kui and Hendry , AHG12: Miscellaneous AHG12 topics , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3-12 July 2019, [JVET-O0145-v1] , JVET-O0145 (version 1) , ITU-T , 2019年06月20日 , < URL:https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O0145-v1.zip > : JVET-O0145-v1.docx: pp.1-3
 BROSS, Benjamin et al. , Versatile Video Coding (Draft 6) , Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3-12 July 2019, [JVET-O2001-v7] , JVET-O2001 (version 7) , ITU-T , 2019年07月15日 , < URL:https://jvet-experts.org/doc_end_user/documents/15_Gothenburg/wg11/JVET-O2001-v7.zip > : JVET-O2001-v7.docx: pp.19-21,37-40,45-49,90-96,108-117
- (58)調査した分野 (Int.Cl. , D B 名)
 H 0 4 N 1 9 / 0 0 - 1 9 / 9 8