(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2021-48569
(P2021-48569A)
(43)公開日　令和3年3月25日(2021.3.25)

(51)Int.Cl.		F I		テーマコード（参考）
H04N 19/70	(2014.01)	H04N	19/70	5C159
H04N 19/436	(2014.01)	H04N	19/436	

審査請求 未請求　請求項の数 2　ＯＬ　（全 18 頁）

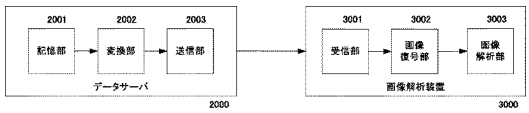
(21)出願番号　特願2019-171790 (P2019-171790)
(22)出願日　令和1年9月20日 (2019.9.20)

(71)出願人　308036402
株式会社ＪＶＣケンウッド
神奈川県横浜市神奈川区守屋町３丁目１２番地
(72)発明者　竹原　英樹
神奈川県横浜市神奈川区守屋町３丁目１２番地
(72)発明者　中村　博哉
神奈川県横浜市神奈川区守屋町３丁目１２番地
(72)発明者　坂爪　智
神奈川県横浜市神奈川区守屋町３丁目１２番地

最終頁に続く

(54)【発明の名称】画像復号装置、画像復号方法及び画像復号プログラム

(57)【要約】
【課題】画像符号化及び復号化に適したブロック分割を行うことにより、符号化効率を向上させる技術を提供する。
【解決手段】ブロックを１ブロック行以上まとめたブリックを整数個含むスライスの符号列を復号し、復号スライスの符号列はＮＡＬユニットに含まれ、スライスに含まれる２番目以降のブリックの開始バイト位置を示すエントリポイントを前記スライスが含まれるＮＡＬユニットとは別のＮＡＬユニットから復号する。
【選択図】図１



【特許請求の範囲】
【請求項１】
画像を所定サイズのブロックに分割して、前記ブロックを１ブロック行以上まとめたブリックを整数個含むスライスの符号列を復号する画像復号装置であって、
前記スライスの符号列はＮＡＬユニットに含まれ、
スライスに含まれる２番目以降のブリックの開始バイト位置を示すエントリポイントを前記スライスが含まれるＮＡＬユニットとは別のＮＡＬユニットから復号する画像復号装置。
【請求項２】
画像を所定サイズのブロックに分割して、前記ブロックを１ブロック行以上まとめたブリックを整数個含むスライスの符号列を復号する画像復号装置であって、
スライスが２以上のブリックを含む可能性があることを示すフラグが、スライスが２以上のブリックを含む可能性があることを示す場合、スライスに含まれる２番目以降のブリックの開始バイト位置を示すエントリポイントをスライスヘッダに含めるか否かを示すフラグを、ピクチャのパラメータを設定するピクチャパラメータセットから復号する画像復号装置。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、並列処理可能な画像符号化及び復号技術に関する。
【背景技術】
【０００２】
ＨＥＶＣでは並列処理のツールとしてタイルとエントロピー同期符号化が定義されている。ＪＶＥＴ（Ｊｏｉｎｔ　Ｖｉｄｅｏ　Ｅｘｐｅｒｔ　Ｔｅａｍ）にて策定中の次世代の映像符号化規格であるＶＶＣ（Ｖｅｒｓａｔｉｌｅ　Ｖｉｄｅｏ　Ｃｏｄｉｎｇ）では並列処理のツールの改善が検討されている。
【先行技術文献】
【非特許文献】
【０００３】
Versatile Video Coding (Draft 6)
【発明の概要】
【発明が解決しようとする課題】
【０００４】
非特許文献１の技術には規格策定途上にあり、並列処理のツールの処理効率に課題がある。
【課題を解決するための手段】
【０００５】
上記課題を解決する本発明のある態様では、画像を所定サイズのブロックに分割して、前記ブロックを１ブロック行以上まとめたブリックを整数個含むスライスの符号列を復号し、前記スライスの符号列はＮＡＬユニットに含まれ、スライスに含まれる２番目以降のブリックの開始バイト位置を示すエントリポイントを前記スライスが含まれるＮＡＬユニットとは別のＮＡＬユニットから復号する。
【発明の効果】
【０００６】
本発明によれば、並列処理のツールを高効率に実現することができる。
【図面の簡単な説明】
【０００７】
【図１】第１の実施の形態の構成を説明する図である。
【図２】符号化ストリーム中のピクチャがブリックに分割されている例を説明する図であ

る
【図３】タイルへのＣＴＵの割り当て情報とブリックへのＣＴＵ行の割り当て情報に関するＰＰＳのシンタックスを説明する図である。
【図４】ブリックインデックスの例を説明する図である。
【図５】ＰＰＳの一部のシンタックスを説明する図である。
【図６】スライスヘッダの一部のシンタックスを説明する図である。
【図７】変換部の動作を説明するフローチャートである。
【図８】ＳＥＩメッセージのシンタックスを示す図である。
【図９】ＳＥＩペイロードのシンタックスを示す図である。
【図１０】エントリポイントＳＥＩメッセージのシンタックスを説明する図である
【図１１】変換部が出力する符号化ストリームを説明する図である。
【図１２】実施の形態１の変形例のＰＰＳの一部のシンタックスを説明する図である。
【図１３】第２の実施の形態の構成を説明する図である。
【図１４】第１の実施の形態の符号化復号装置のハードウェア構成の一例を説明するための図である。
【発明を実施するための形態】
【０００８】
本発明の実施の形態は、ＪＶＥＴ（Ｊｏｉｎｔ Ｖｉｄｅｏ Ｅｘｐｅｒｔ Ｔｅａｍ）にて策定中の次世代の映像符号化規格であるＶＶＣ（Ｖｅｒｓａｔｉｌｅ Ｖｉｄｅｏ Ｃｏｄｉｎｇ）で検討されている技術に基づくものとする。ＶＶＣでは、従来のＡＶＣやＨＥＶＣと同様に、ブロック分割、イントラ予測、インター予測、変換、量子化、エントロピー符号、ループフィルタ等の技術が用いられる。
【０００９】
以下、本発明の実施の形態において使用する技術、及び技術用語を定義する。以降、特に断らない限り画像とピクチャは同じ意味で用いる。本発明の実施の形態において画像符号化部では符号化が実施され、画像復号部では復号が実施される。
【００１０】
＜符号化ツリーユニットと符号化ブロック＞
所定サイズのブロックでピクチャを均等分割する。この所定サイズのブロックを符号化ツリーブロック（ＣＴＢ）と定義する。各符号化ツリーブロックの内部は、再帰的な分割が可能である。符号化ツリーブロック分割後の、符号化・復号の対象となるブロックを符号化ブロック（ＣＵ）と定義する。イントラ予測やインター予測は符号化ブロック単位で符号化（復号）される。また、符号化ツリーブロック、符号化ブロックを総称してブロックと定義する。適切なブロック分割を行うことにより効率的な符号化（復号）が可能となる。符号化ツリーブロックのサイズは、エンコーダとデコーダで予め取り決めた固定値とすることもできるし、エンコーダが決定した符号化ツリーブロックのサイズをデコーダに伝送するような構成をとることもできる。
【００１１】
また、ピクチャにおいて、１つの輝度の符号化ツリーブロックと２つの色差の符号化ツリーブロックを符号化の単位として符号化ツリーユニット(ＣＴＵ)と定義する。ＣＴＵは、１つの輝度の符号化ツリーブロックを単位とするものや、３つのカラープレーンの符号化ツリーブロックを単位とするものがある。ＣＴＵ（ＣＴＢ）のサイズはＳＰＳ（Ｓｅｑｕｅｎｃｅ Ｐａｒａｍｅｔｅｒ Ｓｅｔ）で符号化（復号）される。
【００１２】
＜タイルとブリック＞
タイル（Ｔｉｌｅ）は、ピクチャ内の複数のＣＴＵで構成された矩形領域である。ブリック（Ｂｒｉｃｋ）は、タイル内の１以上のＣＴＵ行で構成された矩形領域である。１つのＣＴＵは複数のタイルに属することはなく、１つのタイルのみに属する。１つのＣＴＵは複数のブリックに属することはなく、１つのブリックのみに属する。１つのブリックは複数のタイルに属することはなく、１つのタイルのみに属する。タイルへのＣＴＵの割り

当てと、ブリックへのＣＴＵ行の割り当ては、それぞれピクチャ単位で設定される。そして、タイルへのＣＴＵの割り当て情報と、ブリックへのＣＴＵ行の割り当て情報はそれぞれＰＰＳ（Ｐｉｃｔｕｒｅ　Ｐａｒａｍｅｔｅｒ　Ｓｅｔ）で符号化（復号）される。以上のように、ピクチャは領域が重複しない複数のタイルに分割される。また、タイルは領域が重複しない複数のブリックに分割される。

【００１３】

符号化ストリームのパラメータを設定するＳＰＳや符号化ピクチャのパラメータを設定するＰＰＳのパラメータ情報とスライスヘッダがあれば、各符号化ブリックは単独で復号が可能である。そのため、ＳＰＳ、ＰＰＳ、スライスヘッダがあれば、複数の符号化ブリックを同時に並列に符号化（復号）することができる。符号化ブリックはブリックを符号化して生成されるブリックの符号列であるとする。以下、符号化ピクチャはピクチャを符号化して生成されるピクチャの符号列とし、符号化スライスはスライスを符号化して生成されるスライスの符号列とする。

【００１４】

図３はタイルへのＣＴＵの割り当て情報とブリックへのＣＴＵ行の割り当て情報に関するＰＰＳのシンタックスを説明する図である。brick_splitting_present_flagはタイルを２以上のブリックに分割するか否かを示すフラグである。brick_splitting_present_flagが１であれば、１以上のタイルが２以上のブリックに分割される。brick_splitting_present_flagが０であれば、全てのタイルは２以上のブリックに分割されない。brick_splitting_present_flagが１であれば、タイル毎にブリックへの分割情報を符号化（復号）する。

【００１５】

タイルがブリックで分割されなかった場合、１タイルは１ブリックとして扱われる。タイルインデックスはピクチャ内のタイルに対してラスタスキャン順に割り当てられるインデックスである。ブリックインデックスはピクチャ内のブリックに割り当てられるインデックスである。ピクチャが複数のタイルに分割される場合、ブリックインデックスはピクチャ内のタイルに対して、タイルインデックス順のタイル内の各ブリックにラスタスキャン順に割り当てられるインデックスである。ブリックインデックスは０以上でピクチャ内のブリック数（NumBricksInPic）未満の整数値である。ラスタスキャン順とは最初に水平方向にスキャンし、その後垂直方向に移動して水平方向にスキャンする順序である。図４はブリックインデックスの例を説明する図である。図４（ａ）はピクチャ内のブリック数８、図４（ｂ）はピクチャ内のブリック数５の一例をそれぞれ示す。

【００１６】

＜ＣＴＵの処理順序＞

ＣＴＵはブリック内でラスタスキャン順に処理され、ブリックはタイル内でラスタスキャン順に処理される。

【００１７】

＜スライスとスライスモード＞

スライスは、ブリックを整数個まとめた領域となる。スライスにはブロックをまとめるモードとして、矩形スライスモードとラスタスキャンスライスモードがある。図５はＰＰＳの一部のシンタックスを説明する図である。図５を用いて矩形スライスモードとラスタスキャンスライスモードを説明する。スライスが２以上のブリックを含む可能性があることを示すフラグ（single_brick_per_slice_flag）が、スライスが２以上のブリックを含む可能性があることを示す場合（single_brick_per_slice_flagが０）、rect_slice_flagが符号化（復号）される。スライスが１ブリックしか含まない場合（single_brick_per_slice_flagが１）、rect_slice_flagは０となる。rect_slice_flagが１であれば矩形スライスモード、rect_slice_flagが０であればラスタスキャンスライスモードとなる。

【００１８】

＜矩形スライスモード＞

矩形スライスモードでは、矩形領域内のブリックをまとめてスライスとする。矩形スラ

イスモードでは、ブリックの処理の順序がラスタスキャン順ではないため、スライスとブリックの対応関係が一意に決まらない。そこで、スライスとブリックの対応関係を示す情報をＰＰＳで符号化（復号）することで、スライスとブリックの対応関係を取得することができる。

【００１９】

ピクチャ内のスライス数から１を減算した値であるnum_slices_in_pic_minus1がＰＰＳ内で符号化（復号）されて、ピクチャ内のスライス数を取得する。ピクチャ内の各スライスに含まれる右下のブリックインデックスを示すbottom_right_brick_idx_deltaとbrick_idx_delta_sign_flagがＰＰＳ内で符号化（復号）されて、ピクチャ内のスライスとブリックの対応関係が導出される。

【００２０】

以上のように、矩形スライスモードでは、スライスとブリックの対応関係を示す情報をＰＰＳで符号化（復号）することで、ブリックの処理の順序に自由度を与えることができる。例えば、図４（ｂ）のように、ブリック０とブリック１が属するタイルに大画面をその他のブリックに小画面を割り当てることで、大画面の処理順序を優先することができる。

【００２１】

＜ラスタスキャンスライスモード＞

ラスタスキャンスライスモードは、ラスタスキャン順にブリックをスライスとする。そのため、ラスタスキャンスライスモードでは、ブリックとスライスの位置の対応関係をＰＰＳで符号化（復号）しなくてもよい。ラスタスキャンスライスモードでは、スライスとブリックの対応関係を示す情報をＰＰＳで符号化（復号）をしないことで、スライスとブリックの対応関係に自由度を与えることができる。例えば、図４（ｂ）で、スライスをブリック０のみで構成するのかブリック０とブリック１の２つで構成するのか等、スライスに含まれるブリックの数をエンコード中に調整することができる。これにより、スライスの符号量を調整することが可能になる。

【００２２】

＜スライスＩＤ＞

スライスＩＤはスライスを一意に識別するためのＩＤである。ラスタスキャンスライスモードの場合、ラスタスキャン順に並んだｉ番目のスライス［ｉ］のスライスＩＤはｉであり、スライスＩＤは０以上でnum_slices_in_pic_minus1以下の値となる。矩形スライスモードの場合、スライスＩＤはＰＰＳ内で任意の値として明示的に符号化（復号）することもできる。矩形スライスモードの場合でスライスＩＤはＰＰＳ内で明示的に符号化（復号）しなければ、ラスタスキャン順に並んだｉ番目のスライス［ｉ］のスライスＩＤはｉであり、スライスＩＤは０以上でnum_slices_in_pic_minus1以下の値となる。

【００２３】

＜ループフィルタの境界制御フラグ＞

ここで、ループフィルタの境界制御フラグを図５のシンタックスを用いて説明する。loop_filter_across_bricks_enabled_flagはループフィルタをブリック境界に適用するか否かを示すフラグであり、１であればループフィルタを適用し、０であればループフィルタを適用しない。loop_filter_across_slices_enabled_flagはループフィルタをスライス境界に適用するか否かを示すフラグであり、１であればループフィルタを適用し、０であればループフィルタを適用しない。ブリック境界とスライス境界が重なる場合は、両方のフラグが１であればループフィルタを適用し、いずれかのフラグまたは両方のフラグが０であればループフィルタを適用しない。

【００２４】

＜スライスヘッダ＞

次に、スライスヘッダについて説明する。図６はスライスヘッダの一部のシンタックスを説明する図である。矩形スライスモードであるか、又は、ピクチャ内のブリック数であるNumBricksInPicが１より大きい場合、スライスアドレス（slice_address）が符号化（

復号）される。また、ラスタスキャンスライスモードであり、且つ、single_brick_per_slice_flagが０であれば、スライスに含まれるスライス数を示すnum_bricks_in_slice_minus1が符号化（復号）される。

【００２５】

ラスタスキャンスライスモードの場合、スライスアドレスはスライスに含まれる最初のブリックＩＤを示す。ブリックＩＤはピクチャ内の各ブリックに含まれるＣＴＵに対して割り当てられるＩＤである。同じブリックに属するＣＴＵは同一のブリックＩＤを有する。ブリックＩＤは０以上でピクチャ内のブリック数（NumBricksInPic）未満の整数値である。ラスタスキャンスライスモードの場合、ピクチャ内のブリックの位置はＰＰＳで定義されているため、スライスアドレスはスライスに含まれる最初のブリックＩＤにすることで、ピクチャ内におけるスライスの開始位置を最も早く確定することができる。slice_addressのビット長はCeil(Log2(NumBricksInPic))とし、slice_addressは0以上で（NumBricksInPic-1）以下の値となる。ここで、ラスタスキャンスライスモードの場合、スライスとブリックの対応関係を示す情報がＰＰＳで符号化（復号）されていないため、スライスアドレスをスライスに含まれる最初のブリックＩＤとする。ラスタスキャンスライスモードの場合、ブリックをラスタスキャン順に連続してスライスに含めるため、最初のブリックＩＤとスライスに含まれるスライス数がわかれば、スライスとブリックの関係を一意に定めることができる。なお、Ceil(x)は x 以上の最小の整数値とする関数である。

【００２６】

矩形スライスモードの場合、スライスとブリックの対応関係を示す情報がＰＰＳで符号化（復号）されている。そのため、スライスＩＤからブリックＩＤを導出することができ、スライスアドレスはスライスＩＤを示すものとする。ここでは、矩形スライスモードの場合はスライスアドレスをスライスＩＤにするとしたが、ラスタスキャンスライスモードと同様にスライスに含まれる最初のブリックＩＤとしてもよい。矩形スライスモードの場合にもラスタスキャンスライスモードと同様にスライスアドレスをスライスに含まれる最初のブリックＩＤとすることで、余計な回路やモジュールなどが不要になり、エンコーダやデコーダのメモリ量や回路規模を削減できる。また、ＰＰＳのsignalled_slice_id_flagが０であれば、スライスＩＤの管理が不要になり、エンコーダやデコーダのメモリ量を更に削減できる。

【００２７】

＜エントリポイント＞

次に、図６を用いてスライスデータ中の符号化ブリックの開始バイト位置について説明する。図６で示されるように、スライスデータ中の符号化ブリックの開始バイト位置はエントリポイント（entry_point_offset_minus1[i]）として、スライスヘッダに符号化（復号）されるため、デコーダでは、スライスヘッダを復号することで符号化ブリックの開始バイト位置を取得することができる。ここで、NumEntryPointsはスライス内のブリック数から１を減算した値である。スライスデータ内の最初の符号化ブリックの開始バイト位置は０で自明であるから符号化（復号）されず、スライスデータ内の２番目以降の符号化ブリックの開始バイト位置が符号化（復号）される。entry_point_offset_present_flagはスライスヘッダにエントリポイントを符号化（復号）するか否かを判定するフラグである。entry_point_offset_present_flagが１であればエントリポイントを符号化（復号）し、entry_point_offset_present_flagが０であればエントリポイントを符号化（復号）しない。offset_len_minus1はentry_point_offset_minus1のビット長を示す値である。なお、entry_point_offset_present_flagは図５で示すようにsingle_tile_in_pic_flagが０またはentropy_coding_sync_enabled_flagが１である時にＰＰＳで符号化（復号）される。ここで、single_tile_in_pic_flagはピクチャが１タイルであるか否かを示すフラグである。single_tile_in_pic_flagが１であれば、ピクチャは１タイルで構成され、single_tile_in_pic_flagが０であれば、ピクチャは複数タイルで構成される。ピクチャを１タイルで構成する場合、タイルを複数のブリックで構成することを許容せず、１タイルは１ブリックで構成される。entropy_coding_sync_enabled_flagはブリック内のＣＴＵ列の並列処

理を許可するか否かを示すフラグである。entropy_coding_sync_enabled_flagが１であれば、ブリック内のＣＴＵ列の並列処理を許可し、entropy_coding_sync_enabled_flagが０であれば、ブリック内の並列処理を許可しない。

【００２８】

＜ＳＥＩ＞

ＳＥＩ（Supplemental enhancement information）は、復号画像の画素を生成するためのプロセスには不要な情報であるが、デコーダの動作に必要な情報である。例えば、ＳＥＩには、仮想参照デコーダであるＨＲＤ（Hypothetical Reference Decoder）を正確に動作させるための情報であるBuffering Period SEIや復号画像の出力タイミングを定めるPicture Timing SEIが含まれる。ＳＥＩはＳＥＩメッセージとして符号化（復号）される。

【００２９】

＜ＮＡＬユニット＞

ＮＡＬユニットはヘッダとペイロードで構成される。ヘッダでペイロードタイプが符号化（復号）されて、ペイロードタイプで示されるタイプの符号列がペイロードに含まれる。ペイロードタイプには、ＳＰＳ、ＰＰＳ、符号化スライス、ＳＥＩメッセージ等がある。例えば、ペイロードタイプがスライスを含むことを示すペイロードタイプであれば、スライスの符号列がペイロードに格納され、ペイロードタイプがＳＥＩメッセージを含むことを示すペイロードタイプであれば、ＳＥＩメッセージの符号列がペイロードに格納される。

【００３０】

＜符号化ストリーム＞

ＶＶＣやＨＥＶＣでは、ＳＰＳ、ＰＰＳ、符号化スライス、ＳＥＩメッセージ等がＮＡＬユニットに格納されて符号化ストリームとなる。符号化スライスは、スライスヘッダとスライスデータで構成される。スライスデータに符号化ツリーブロックが符号化される。符号化ストリームには１以上の符号化ピクチャが含まれる。

【００３１】

＜ＨＲＤ＞

ＨＲＤは符号化ストリームが符号化規格に適合していることを検査するための仮想参照デコーダである。ＨＲＤは符号化ストリームを復号するためのＣＰＢ（符号化ピクチャバッファ）と復号したピクチャを出力するためのＤＰＢ（復号ピクチャバッファ）が定義されている。ＨＲＤでは、符号化ストリームに符号化されたＣＰＢとＤＰＢの入出力タイミングに基づいて符号化ストリームを復号した場合にＣＰＢやＤＰＢが破綻しないことを検査する。ＣＰＢとＤＰＢが破綻しなければ、符号化ストリームは符号化規格に適合している。ＨＲＤのタイプ１では符号化ストリームの符号化スライスの規格適合が検査される。ＨＲＤのタイプ１では、ＳＰＳ、ＰＰＳ、ＳＥＩメッセージが含まれるＮＡＬは検査対象外である。ＣＰＢとＤＰＢの入出力タイミングはＳＥＩメッセージで符号化（復号）される。

【００３２】

（第１の実施の形態）

＜第１の実施の形態の構成＞

本発明の第１の実施の形態について説明する。最初に、本実施の形態の構成を説明する。図１は第１の実施の形態の構成を説明する図である。本実施の形態は、データサーバ２０００と画像解析装置３０００で構成される。データサーバ２０００は、記憶部２００１、変換部２００２、及び送信部２００３で構成される。データサーバ２０００は、符号化ストリームを変換する機能を有するため、画像変換装置とも呼ぶ。画像解析装置３０００は、受信部３００１、画像復号部３００２及び画像解析部３００３で構成される。

【００３３】

記憶部２００１には符号化ストリームが記憶されている。画像復号部３００２は記憶部２００１に記憶された符号化ストリームを復号可能である。画像復号部３００２は並列に処理できるブリックの最大数（以降、最大ブリック並列処理数）は８とする。

【００３４】
＜第１の実施の形態の動作＞
次に、本実施の形態の動作を説明する。図１に基づいて各構成要素の動作を説明する。まず、データサーバ２０００と画像解析装置３０００の動作について説明する。
【００３５】
データサーバ２０００は、画像解析装置３０００の要求に基づいて、記憶部２００１から符号化ストリームと画像復号部３００２の特性パラメータを読み出して変換部２００２に入力する。特性パラメータは画像復号部３００２がエントリポイントＳＥＩに対応しているか否かの情報であり、特性パラメータは予め記憶部２００１に保持されているものとする。変換部２００２は、入力された符号化ストリームを変換し、変換した符号化ストリームを送信部２００３に入力する。変換部２００２の詳細な動作は後述する。送信部２００３は、入力された符号化ストリームを画像解析装置３０００に送信する。
【００３６】
受信部３００１は送信部２００３から入力された符号化ストリームを画像復号部３００２に入力する。画像復号部３００２は入力された符号化ストリームを最大ブリック並列処理数で並列に復号して復号画像を出力し、画像解析部３００３に入力する。ここで、画像復号部３００２は入力された符号化ストリームに復号画像を出力するフレームレートが設定されている場合でもフレームレートにとらわれることなく復号画像を出力する。画像解析部３００３は入力された復号画像を解析して解析結果を出力する。例えば、画像解析部３００３は入力された復号画像について、画質の計測、顔や人物等の認識等を行う。
【００３７】
以上のように、画像解析部３００３で復号画像を解析するような場合、符号化ストリームのフレームレートにとらわれることなく画像復号部３００２から復号画像を出力することで画像解析部３００３から出力される解析結果を短時間で取得することができる。
【００３８】
＜符号化ストリーム＞
記憶部２００１に記憶されている符号化ストリームについて説明する。図２は符号化ストリーム中のピクチャがブリックに分割されている例を説明する図である。ここでは説明を簡単にするためにピクチャをタイル分割し、タイルはブリックに分割しないものとする。つまり、１タイルが１ブリックとなる。図２（ａ）は１ＣＴＵ行毎に画面を水平方向にＢ０（Ｔ０）からＢ７（Ｔ７）の８つのブリック（タイル）に分割した例である。図２（ｂ）は１ＣＴＵ列毎に画面を垂直方向にＢ０（Ｔ０）からＢ７（Ｔ７）の８つのブリック（タイル）に分割した例である。ブリックＢ０からブリックＢ７にはそれぞれブリックＩＤとして０から７が割り当てられる。ここでは、ピクチャ内のブリック数NumBricksInPicは８である。
【００３９】
ここでは、１ピクチャは１スライスで構成されるとする。従って、スライス０にＢ０からＢ７の８つのブリックが含まれる。そのため、ＰＰＳのsingle_brick_per_slice_flagは０である。符号化ストリームはラスタスキャンスライスモードでも矩形スライスモードでもよい。ただし、スライスヘッダにエントリポイントを符号化するか否かのフラグであるentry_point_offset_present_flagは０とする。
【００４０】
ここで、entry_point_offset_present_flagを０にする効果を説明する。entry_point_offset_present_flagを１にすると、スライスヘッダにブリックの符号化した結果で得られるエントリポイントを符号化する必要があり、スライス内のブリックを並列に符号化して符号化ストリームを生成したとしても、ブリックのサイズを示すバイト数が確定するまで符号化ストリームを送信できない。そのため、符号化ストリームをスライス単位で送信する必要があり、スライス単位でしか低遅延化できない。一方、entry_point_offset_present_flagを０にすると、スライスヘッダにブリックのエントリポイントを符号化する必要はなく、符号化ストリームをブリック単位で送信でき、ブリック単位で低遅延化できる。

【００４１】
＜変換部の詳細な動作＞
続いて、変換部２００２の詳細な動作を説明する。図７は変換部の動作を説明するフローチャートである。図７は符号化ストリームに含まれる全ての符号化ピクチャに対して実行される。
【００４２】
ループフィルタを適用するか否かがスライス境界とブリック境界で同一でないかを検査する（Ｓ１００）。具体的には、loop_filter_across_bricks_enabled_flagとloop_filter_across_slices_enabled_flagの値が同一でないか検査する。ループフィルタを適用するか否かがスライス境界とブリック境界で同一でない場合（Ｓ１００のＹＥＳ）、エントリポイントＳＥＩ対応デコーダであるか検査する（Ｓ１０１）。ループフィルタを適用するか否かがスライス境界とブリック境界で同一である場合（Ｓ１００のＮＯ）、スライスを再構成する（Ｓ１０４）。スライスを再構成する詳細な動作は後述する。
【００４３】
エントリポイントＳＥＩ対応デコーダであれば（Ｓ１０１のＹＥＳ）、エントリポイントをエントリポイントＳＥＩに追加する（Ｓ１０２）。エントリポイントをエントリポイントＳＥＩに追加する詳細な動作は後述する。エントリポイントＳＥＩ対応デコーダでなければ（Ｓ１０１のＮＯ）、エントリポイントをスライスヘッダに追加する（Ｓ１０３）。
【００４４】
ここで、ループフィルタを適用するか否かがスライス境界とブリック境界で同一でない場合、各スライスに含まれるブリックを変更すると、復号画像が変わってしまうため、再符号化が必要となる。そのため、ループフィルタを適用するか否かがスライス境界とブリック境界で同一でない場合、各スライスに含まれるブリックの構成を変更しないように符号化ストリームを変換する。
【００４５】
一方、ループフィルタを適用するか否かがスライス境界とブリック境界で同一である場合、各スライスに含まれるブリックを変更しても復号画像が変わらないため、再符号化が不要である。そのため、ループフィルタを適用するか否かがスライス境界とブリック境界で同一である場合、ブリックとスライスの構成を変更するように符号化ストリームを変換する。
【００４６】
ここでは、スライスの並列処理に適したデコーダを考慮してloop_filter_across_bricks_enabled_flagとloop_filter_across_slices_enabled_flagの値が同一でないか検査する（Ｓ１００）ステップを設置した。一般的なデコーダを対象とする場合、loop_filter_across_bricks_enabled_flagとloop_filter_across_slices_enabled_flagの値が同一でないか検査する（Ｓ１００）処理を行わず、エントリポイントＳＥＩ対応デコーダであるか検査する（Ｓ１０１）処理から開始するのが良い。
【００４７】
＜エントリポイントのスライスヘッダへの追加＞
ここで、エントリポイントをスライスヘッダに追加する動作について説明する。エントリポイントのスライスヘッダへの追加は、図６のシンタックスに従って、ブリックＢ１からブリックＢ７のエントリポイントの情報が符号化ストリームに符号化（復号）される。
【００４８】
＜スライスの再構成＞
続いて、スライスを再構成する詳細な動作について説明する。記憶部２００１に記憶されている符号化ストリームは１つのスライスが８つのブリックを含んでいた。これを１つのスライスが１つのブリックを含むように再構成する。なお、ＰＰＳのsingle_brick_per_slice_flagを１にする。
【００４９】

＜エントリポイントのエントリポイントＳＥＩへの追加＞

続いて、エントリポイントをエントリポイントＳＥＩに追加する詳細な動作について説明する。エントリポイントのエントリポイントＳＥＩへの追加は、図８、図９、図１０のシンタックスに従って、ブリックＢ１からブリックＢ７のエントリポイントの情報が符号化（復号）される。

【００５０】

図８はＳＥＩメッセージのシンタックスを示す図である。ペイロードタイプ（payloadType）がpayload_type_byteから導出される。ペイロードタイプが１２８である場合、エントリポイントＳＥＩメッセージがＳＥＩペイロード（sei_payload）に符号化（復号）される。エントリポイントＳＥＩメッセージはエントリポイントＳＥＩと同意である。

【００５１】

図９はＳＥＩペイロードのシンタックスを示す図である。ＳＥＩペイロードにはエントリポイントＳＥＩメッセージのほか、Buffering Period ＳＥＩメッセージやPicture Timing、Picture Timing ＳＥＩメッセージ等のＳＥＩメッセージが符号化（復号）される。ＳＥＩメッセージを格納するＮＡＬユニットタイプは２つあり、符号化スライスよりも前に符号化（復号）するＰＲＥＦＩＸ＿ＳＥＩ＿ＮＵＴと、符号化スライスよりも後に符号化（復号）するＳＵＦＦＩＸ＿ＳＥＩ＿ＮＵＴである。

【００５２】

図１０はエントリポイントＳＥＩメッセージのシンタックスを説明する図である。図１０を用いてエントリポイントＳＥＩメッセージのシンタックスを説明する。NumSlicesInPicはピクチャ内のスライス数であり、矩形スライスモードの場合はＰＰＳで取得でき、ラスタスキャンスライスモードの場合はスライス数をカウントすることで取得できる。Ｓ番目のスライスに関するoffset_len_minus1[s]とNumEntryPoints個のentry_point_offset_minus1が符号化（復号）される。offset_len_minus1とentry_point_offset_minus1は前述と同義である。

【００５３】

＜変換後の符号化ストリーム＞

図１１は変換部２００２が出力する符号化ストリームを説明する図である。図１１（ｉ）は、入力された符号化ストリームの一例である。ＳＰＳ、ＰＰＳ、符号化ピクチャ０を構成する符号化スライス、符号化ピクチャ１を構成する符号化スライス０、・・・、符号化ピクチャＮを構成する符号化スライスの順で符号化されている。

【００５４】

図１１（ａ）は、エントリポイントをスライスヘッダに追加した符号化ストリームの一例である。以降、図１１（ａ）について説明する。ＳＰＳ、ＰＰＳ、符号化スライス０のそれぞれのＮＡＬユニットが符号化ストリームとなる。符号化スライス０は、エントリポイントが追加されたスライスヘッダ（図１１（ａ）のModified S-Header）と符号化ブリック０（図１１（ａ）のBrick 0）から符号化ブリック７（図１１（ａ）のBrick 7）の８つの符号化ブリックを含むスライスデータで構成される。ここで、エントリポイントが追加されたスライスヘッダは入力された符号化ストリームのスライスヘッダとは異なる。なお、符号化ブリック０から符号化ブリック７を復号するにはエントリポイントが追加されたスライスヘッダが必要になる。

【００５５】

図１１（ｂ）は、スライスを再構成した符号化ストリームの一例である。以降、図１１（ｂ）について説明する。ＳＰＳ、ＰＰＳ、符号化スライス０から符号化スライス７のそれぞれのＮＡＬユニットが符号化ストリームとなる。符号化スライスｉ（ｉ＝０、１、・・・７）は各々スライスヘッダｉ（図１１（ｂ）のNew S-Header i）と符号化ブリックｉ（図１１（ｂ）のBrick i）を含むスライスデータで構成される。ここで、スライスヘッダｉは入力された符号化ストリームのスライスヘッダとは異なる。なお、符号化ブリックｉを復号するにはスライスヘッダｉが必要になる。

【００５６】

このように、１つの符号化スライスを１つの符号化ブリックを含む符号化ピクチャに変換し、ＰＰＳのsingle_brick_per_slice_flagを１とすることで、符号化スライス単位での並列化を容易に実現できる。この場合、エントリポイントは不要であり、ＮＡＬユニット単位でアクセスできればよい。

【００５７】

ここでは、１つの符号化スライスが１つのブリックを含むように符号化ピクチャを変換しているが、画像復号部３００２の最大ブリック並列処理数に応じてスライスに含まれるブリックを再構成すればよく、これに限定されない。１つの符号化スライスが複数のブリックを含むように符号化ピクチャを変換してもよい。例えば、画像復号部３００２の最大ブリック並列処理数が４である場合、１つの符号化スライスが２つのブリックを含むように符号化ピクチャを変換する。

【００５８】

図１１（ｃ）は、エントリポイントをエントリポイントＳＥＩに追加した符号化ストリームの一例である。以降、図１１（ｃ）について説明する。ＳＰＳ、ＰＰＳ、符号化スライス０、ＳＥＩのそれぞれのＮＡＬユニットが符号化ストリームとなる。符号化スライス０は、スライスヘッダ（図１１（ｃ）のS-Header）と符号化ブリック０（図１１（ｃ）のBrick 0）から符号化ブリック７（図１１（ｃ）のBrick 7）の８つの符号化ブリックを含むスライスデータで構成される。ここで、スライスヘッダは入力された符号化ストリームのスライスヘッダと同一である。なお、符号化ブリック０から符号化ブリック７を復号するにはスライスヘッダが必要になる。

【００５９】

エントリポイントをスライスヘッダに追加した符号化ストリームの変換後の符号化ストリームの大きさは、その他の方法に比べて最小になる。ただし、スライスヘッダをビット精度で再符号化する必要があり、符号化スライスの大きさが変わる。

【００６０】

符号化スライスを再構成した符号化ストリームは、８つの符号化スライスで構成されているため、変換後の符号化ストリームの大きさがその他の方法に比べて大きくなるが、エントリポイントは不要になるため符号化ストリームの構造はシンプルになる。そのため、符号化ブリック単位の並列化に対応していないが、符号化スライス単位の並列化に対応しているデコーダでも並列に復号できる。

【００６１】

エントリポイントをエントリポイントＳＥＩに追加した符号化ストリームは、入力された符号化ストリームは変更することなく、符号化ピクチャ内の最後の符号化スライスの後にエントリポイントＳＥＩを挿入するだけでよく、変換後の符号化ストリームを用意に生成することができる。このように、エントリポイントＳＥＩを符号化スライスの後で符号化（復号）することで、ラスタスキャンスライスモードでスライスとブリックの関係に自由度を与えながら、符号化スライス内のエントリポイントをデコーダに通知することができる。また、符号化ストリームを変更しないため、ＨＲＤのタイプ１の符号化ストリームコンフォーマンスに影響を与えない。そのため、ＨＲＤの適合性を検証しながら符号化ストリームを変換する必要がなく、確実且つ容易に符号化ストリームを変換することができる。

【００６２】

＜変形例＞

実施の形態１の変形例について説明する。本変形例は、本実施の形態とはＰＰＳのシンタックスが異なる。図１２は実施の形態１の変形例のＰＰＳの一部のシンタックスを説明する図である。実施の形態１の図５とはentry_point_offset_present_flag を符号化（復号）する条件が異なる。図１２で示すように、entry_point_offset_present_flagはスライスが２以上のブリックを含む可能性があることを示すフラグが、スライスが２以上のブリックを含む可能性があることを示す場合（single_brick_per_slice_flagが０である）またはentropy_coding_sync_enabled_flagが１である場合にＰＰＳで符号化（復号）され

る。
【００６３】
以上のように、single_brick_per_slice_flagが０である場合にentry_point_offset_pr
esent_flagを符号化（復号）することで、ピクチャが複数タイルで構成されている場合（
single_tile_in_pic_flagが０）でもスライスが１つのブリックしか含まない場合にはent
ry_point_offset_present_flagを符号化する必要がなくなる。そのため、スライスが１つ
のブリックしか含まない場合にentry_point_offset_present_flagが１である状態を解消
し、符号量を削減することができる。また、entry_point_offset_present_flagが存在し
ない場合には暗示的にentry_point_offset_present_flagは０であるとすることで、スラ
イスが１つのブリックしか含まない場合にはエントリポイントが存在しないことを定義す
ることができる。また、スライスヘッダでentry_point_offset_present_flagが０であれ
ばスライスに含まれるブリック数を検査する必要がないため処理量が削減できる。
【００６４】
（第２の実施の形態）
＜第２の実施の形態の構成＞
本発明の第２の実施の形態について説明する。図１３は第２の実施の形態の構成を説明
する図である。本発明の第２の実施の形態は、第１の実施の形態とはデータサーバ２００
０が画像符号化装置１０００であることが異なる。以降、第１の実施の形態とはとは異な
る点のみ説明する。
【００６５】
画像復号部３００２は画像符号化部１００２で符号化された符号化ストリームを復号可
能である。
【００６６】
画像符号化装置１０００は、画像解析装置３０００の要求に基づいて、記憶部１００１
から画像データと画像復号部３００２の特性パラメータを読み出して画像符号化部１００
２に入力する。画像符号化部１００２は、入力された画像データを符号化し、符号化した
符号化ストリームを送信部１００３に入力する。
【００６７】
画像符号化部１００２は変換部２００２とは出力する符号化ストリームを画像データか
ら符号化することが異なる。画像符号化部１００２の出力する符号化ストリームの構造は
変換部２００２の出力する符号化ストリームと同一である。
【００６８】
以下、画像符号化部１００２の動作を説明する。変換部２００２では、loop_filter_ac
ross_bricks_enabled_flagとloop_filter_across_slices_enabled_flagは符号化ストリー
ムから復号して取得するが、画像符号化部１００２では、loop_filter_across_bricks_en
abled_flagとloop_filter_across_slices_enabled_flagは画像符号化部１００２が決定す
る。
【００６９】
ループフィルタを適用するか否かがスライス境界とブリック境界で同一でない場合、画
像復号部３００２がエントリポイントＳＥＩ対応デコーダであるか検査する。ループフィ
ルタを適用するか否かがスライス境界とブリック境界で同一である場合、１つのスライス
が１つのブリックを含むように符号化ピクチャを符号化する。
【００７０】
画像復号部３００２がエントリポイントＳＥＩ対応デコーダであれば、エントリポイン
トをエントリポイントＳＥＩとして符号化ピクチャを符号化する。画像復号部３００２が
エントリポイントＳＥＩ対応デコーダでなければ、エントリポイントをスライスヘッダに
符号化して符号化ピクチャを符号化する。
【００７１】
以上のように、画像符号化装置１０００は、画像解析装置３０００の要求に基づいた画
像復号部３００２が復号可能な符号化ストリームを出力することができる。また、符号化

ストリームの構造が有する効果は実施の形態でも実施の形態１と同様である。

【００７２】

以上に述べた全ての実施の形態において、画像符号化装置が出力する符号化ストリームは、実施の形態で用いられた符号化方法に応じて復号することができるように特定のデータフォーマットを有している。符号化ストリームは、ＨＤＤ、ＳＳＤ、フラッシュメモリ、光ディスク等のコンピュータ等で読み解き可能な記録媒体に記録して提供しても良いし、有線あるいは無線のネットワークを通してサーバから提供しても良い。従って、この画像符号化装置に対応する画像復号装置は、提供手段によらず、この特定のデータフォーマットの符号化ストリームを復号することができる。

【００７３】

画像符号化装置と画像復号装置の間で符号化ストリームをやりとりするために、有線または無線のネットワークが用いええられる場合、通信路の伝送形態に適したデータ形式に符号化ストリームを変換して伝送してもよい。その場合、画像符号化装置が出力する符号化ストリームを通信路の伝送形態に適したデータ形式の符号化データに変換してネットワークに送信する送信装置と、ネットワークから符号化データを受信して符号化ストリームに復元して画像復号装置に供給する受信装置とが設けられる。　送信装置は、画像符号化装置が出力する符号化ストリームをバッファするメモリと、符号化ストリームをパケット化するパケット処理部と、ネットワークを介してパケット化された符号化データを送信する送信部とを含む。受信装置は、ネットワークを介してパケット化された符号化データを受信する受信部と、受信された符号化データをバッファするメモリと、符号化データをパケット処理して符号化ストリームを生成し、画像復号装置に提供するパケット処理部とを含む。

【００７４】

画像符号化装置と画像復号装置の間で符号化ストリームをやりとりするために、有線または無線のネットワークが用いられる場合、送信装置、受信装置に加え、さらに、送信装置が送信する符号化データを受信し、受信装置に供給する中継装置が設けられても良い。中継装置は、送信装置が送信するパケット化された符号化データを受信する受信部と、受信された符号化データをバッファするメモリと、パケットされた符号化データとネットワークに送信する送信部とを含む。さらに、中継装置は、パケット化された符号化データをパケット処理して符号化ストリームを生成する受信パケット処理部と、符号化ストリームを蓄積する記録媒体と、符号化ストリームをパケット化する送信パケット処理部を含んでも良い。

【００７５】

また、画像復号装置で復号された画像を表示する表示部を構成に追加することで、表示装置としても良い。

【００７６】

また、撮像部を構成に追加し、撮像した画像を画像符号化装置に入力することで、撮像装置としても良い。

【００７７】

図１４に、本願の符号化復号装置のハードウェア構成の一例を示す。符号化復号装置は、本発明の実施の形態に係る画像符号化装置、および画像復号装置の構成を包含する。係る符号化復号装置９０００は、ＣＰＵ９００１、コーデックＩＣ９００２、Ｉ／Ｏインターフェース９００３、メモリ９００４、光学ディスクドライブ９００５、ネットワークインターフェース９００６、ビデオインターフェース９００９を有し、各部はバス９０１０により接続される。

【００７８】

画像符号化部９００７と画像復号部９００８は、典型的にはコーデックＩＣ９００２として実装される。本発明の実施の形態に係る画像符号化装置の画像符号化処理は、画像符号化部９００７により実行され、本発明の実施の形態に係る画像復号装置における画像復号処理は、画像符号化部９００７により実行される。Ｉ／Ｏインターフェース９００３は

、例えばＵＳＢインターフェースにより実現され、外部のキーボード９１０４、マウス９１０５等と接続する。ＣＰＵ９００１は、Ｉ／Ｏインターフェース９００３を介して入力したユーザー操作に基づき、ユーザーの所望する動作を実行するように符号化復号装置９０００を制御する。キーボード９１０４、マウス９１０５等によるユーザーの操作としては、符号化、復号のどちらの機能を実行するかの選択、符号化品質の設定、符号化ストリームの入出力先、画像の入出力先等がある。

【００７９】

ユーザーがディスク記録媒体９１００に記録された画像を再生する操作を所望する場合、光学ディスクドライブ９００５は、挿入されたディスク記録媒体９１００から符号化ビットストリームを読出し、読み出した符号化ストリームを、バス９０１０を介してコーデックＩＣ９００２の画像復号部９００８に送る。画像復号部９００８は入力した符号化ビットストリームに対して本発明の実施の形態に係る画像復号装置における画像復号処理を実行し、復号画像を、ビデオインターフェース９００９を介して外部のモニタ９１０３へ送る。また、符号化復号装置９０００は、ネットワークインターフェース９００６を有し、ネットワーク９１０１を介して、外部の配信サーバ９１０６や、携帯端末９１０７と接続可能である。ユーザーがディスク記録媒体９１００に記録された画像に変えて、配信サーバ９１０６や携帯端末９１０７に記録された画像を再生することを所望する場合は、ネットワークインターフェース９００６は、入力されたディスク記録媒体９１００から符号化ビットストリームを読出すことに変えて、ネットワーク９１０１より符号化ストリームを取得する。また、ユーザーがメモリ９００４に記録された画像を再生することを所望する場合は、メモリ９００４に記録された符号化ストリームに対して、本発明の実施の形態に係る画像復号装置における画像復号処理を実行する。

【００８０】

ユーザーが外部のカメラ９１０２で撮像した画像を符号化しメモリ９００４に記録する操作を所望する場合、ビデオインターフェース９００９は、カメラ９１０２から画像を入力し、バス９０１０を介し、コーデックＩＣ９００２の画像符号化部９００７に送る。画像符号化部９００７は、ビデオインターフェース９００９を介して入力した画像に対して本発明の実施の形態に係る画像符号化装置における画像符号化処理を実行し、符号化ビットストリームを作成する。そして符号化ビットストリームを、バス９０１０を介し、メモリ９００４へ送る。ユーザーがメモリ９００４に変えて、ディスク記録媒体９１００に符号化ストリームを記録することを所望する場合は、光学ディスクドライブ９００５は、挿入されたディスク記録媒体９１００に対し符号化ストリームの書き出しを行う。

【００８１】

画像符号化装置を有し画像復号装置を有さないハードウェア構成や、画像復号装置を有し画像符号化装置を有さないハードウェア構成を実現することも可能である。そのようなハードウェア構成は、例えばコーデックＩＣ９００２が、画像符号化部９００７、または画像復号部９００８にそれぞれ置き換わることにより実現される。

【００８２】

以上の符号化及び復号に関する処理は、ハードウェアを用いた伝送、蓄積、受信装置として実現しても良いのは勿論のこと、ＲＯＭ（リード・オンリー・メモリ）やフラッシュメモリ等に記憶されているファームウェアや、コンピュータ等のソフトウェアによって実現しても良い。そのファームウェアプログラム、ソフトウェアプログラムをコンピュータ等で読み取り可能な記録媒体に記録して提供しても良いし、有線あるいは無線のネットワークを通してサーバから提供しても良いし、地上波あるいは衛星ディジタル放送のデータ放送として提供しても良い。

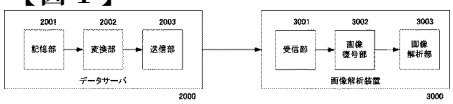
【００８３】

以上、本発明を実施の形態をもとに説明した。実施の形態は例示であり、それらの各構成要素や各処理プロセスの組み合わせにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

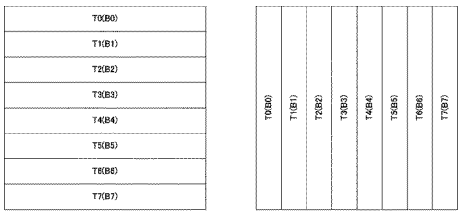
【符号の説明】

【００８４】
１０００　画像符号化装置、１００１　記憶部、１００２　画像符号化部、１００３　送信部、２０００　画像変換装置、２００１　記憶部、２００２　変換部、２００３　送信部、３０００　画像解析装置、３００１　受信部、３００２　画像復号部、３００３　画像解析部

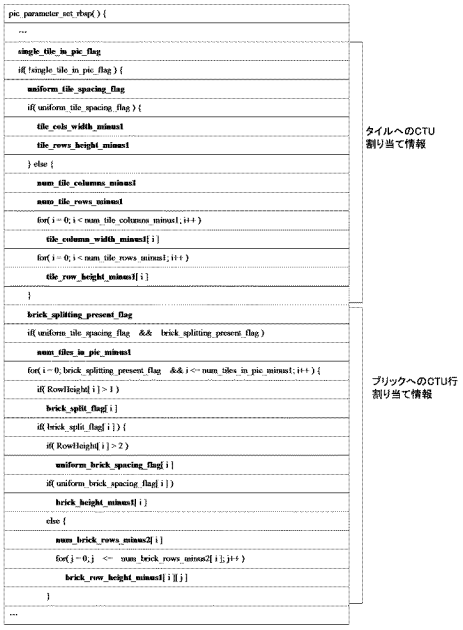
【図１】



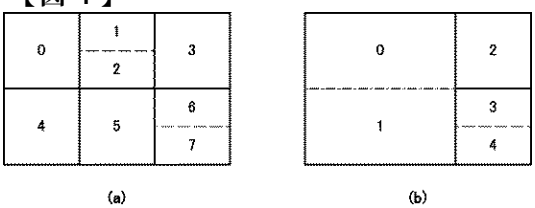
【図２】



【図３】



【図４】



【図５】

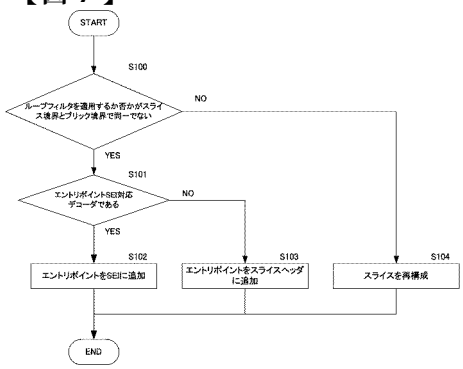
```
pic_parameter_set_rbsp( ) {
  ...
  single_tile_in_pic_flag
  if( !single_tile_in_pic_flag ) {
    ...
    single_brick_per_slice_flag
    if( !single_brick_per_slice_flag )
      rect_slice_flag
    if( rect_slice_flag  &&  !single_brick_per_slice_flag ) {
      num_slices_in_pic_minus1
      bottom_right_brick_idx_length_minus1
      for( i = 0; i < num_slices_in_pic_minus1; i++ ) {
        bottom_right_brick_idx_delta[ i ]
        brick_idx_delta_sign_flag[ i ]
      }
    }
    loop_filter_across_bricks_enabled_flag
    if( loop_filter_across_bricks_enabled_flag )
      loop_filter_across_slices_enabled_flag
  }
  if( rect_slice_flag ) {
    signalled_slice_id_flag
    if( signalled_slice_id_flag ) {
      signalled_slice_id_length_minus1
      for( i = 0; i  <=  num_slices_in_pic_minus1; i++ )
        slice_id[ i ]
    }
  }
  entropy_coding_sync_enabled_flag
  if( !single_tile_in_pic_flag  ||  entropy_coding_sync_enabled_flag )
    entry_point_offsets_present_flag
  cabac_init_present_flag
  ...
```

スライスとブリックの関係を示す情報

【図６】

```
slice_header( ) {
  slice_pic_parameter_set_id
  if( rect_slice_flag  ||  NumBricksInPic > 1 )
    slice_address
  if( !rect_slice_flag  &&  !single_brick_per_slice_flag )
    num_bricks_in_slice_minus1
  slice_type
  ...
  if( entry_point_offsets_present_flag  &&  NumEntryPoints > 0 ) {
    offset_len_minus1
    for( i = 0; i < NumEntryPoints; i++ )
      entry_point_offset_minus1[ i ]
  }
  if( slice_header_extension_present_flag ) {
    slice_header_extension_length
    for( i = 0; i < slice_header_extension_length; i++)
      slice_header_extension_data_byte[ i ]
  }
  byte_alignment( )
}
```

【図７】



【図８】

```
sei_message( ) {
  payloadType = 0
  do {
    payload_type_byte
    payloadType  +=  payload_type_byte
  } while (payload_type_byte  ==  0xFF )
  payloadSize = 0
  do {
    payload_size_byte
    payloadSize  +=  payload_size_byte
  } while (payload_size_byte  ==  0xFF )
  sei_payload( payloadType, payloadSize )
}
```

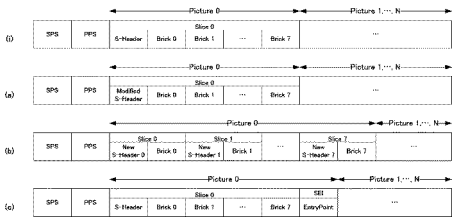
【図９】

```
sei_payload( payloadType, payloadSize ) {
  if( nal_unit_type  ==  PREFIX_SEI_NUT )
    if( payloadType  ==  0 )
      buffering_period( payloadSize )
    else if( payloadType  ==  1 )
      pic_timing( payloadSize )
    ...
  else /* nal_unit_type  ==  SUFFIX_SEI_NUT */
    if( payloadType  ==  128 )
      entry_point ( payloadSize )
    ...
  ...
}
```

【図１０】

```
entry_ponit( payloadSize ) {
{
  for( s = 0; s < NumSlicesInPic; s++ ) {
    offset_len_minus1[s]
    for( i = 0; i < NumEntryPoints[s]; i++ )
      entry_point_offset_minus1[ i ]
  }
}
```

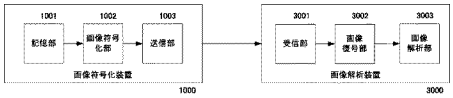
【図１１】



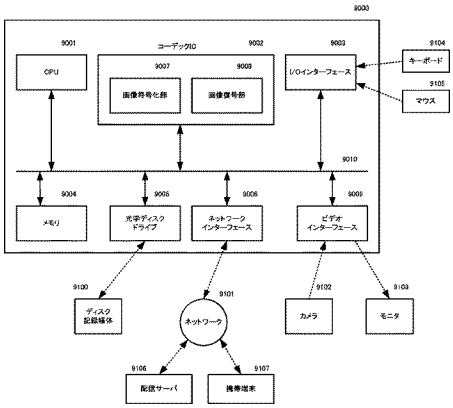
【図１２】

```
pic_parameter_set_rbsp( ) {
  ...
  single_tile_in_pic_flag
  if( !single_tile_in_pic_flag ) {
    ...
    single_brick_per_slice_flag
    if( !single_brick_per_slice_flag )
      rect_slice_flag
    if( rect_slice_flag  &&  !single_brick_per_slice_flag ) {
      num_slices_in_pic_minus1
      bottom_right_brick_idx_length_minus1
      for( i = 0; i < num_slices_in_pic_minus1; i++ ) {
        bottom_right_brick_idx_delta[ i ]
        brick_idx_delta_sign_flag[ i ]
      }
    }
    loop_filter_across_bricks_enabled_flag
    if( loop_filter_across_bricks_enabled_flag )
      loop_filter_across_slices_enabled_flag
  }
  if( rect_slice_flag ) {
    signalled_slice_id_flag
    if( signalled_slice_id_flag ) {
      signalled_slice_id_length_minus1
      for( i = 0; i  <=  num_slices_in_pic_minus1; i++ )
        slice_id[ i ]
    }
  }
  entropy_coding_sync_enabled_flag
  if( !single_brick_per_slice_flag  ||  entropy_coding_sync_enabled_flag )
    entry_point_offsets_present_flag
  cabac_init_present_flag
  ...
```

【図１３】


1001
1002
1003
記憶部
画像符号化部
送信部
画像符号化装置
1000
3001
3002
3003
受信部
画像復号部
画像解析部
画像解析装置
3000

【図１４】


9000
9001
CPU
コーデック IC
9002
9007
画像符号化部
9008
画像復号部
9003
I/Oインターフェース
9104
キーボード
9105
マウス
9010
9004
メモリ
9005
光学ディスクドライブ
9006
ネットワークインターフェース
9009
ビデオインターフェース
9100
ディスク記録媒体
9101
ネットワーク
9102
カメラ
9103
モニタ
9106
配信サーバ
9107
携帯端末

フロントページの続き

(72)発明者　福島　茂
神奈川県横浜市神奈川区守屋町３丁目１２番地
(72)発明者　熊倉　徹
神奈川県横浜市神奈川区守屋町３丁目１２番地
(72)発明者　倉重　宏之
神奈川県横浜市神奈川区守屋町３丁目１２番地

Ｆターム(参考) 5C159 KK13　LC00　LC08　LC09　RC11　TA11　TB06　TC26　TC33　UA02
UA05　UA16