

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5888278号
(P5888278)

(45) 発行日 平成28年3月16日(2016.3.16)

(24) 登録日 平成28年2月26日(2016.2.26)

(51) Int.Cl. F I
HO 4 N 19/117 (2014.01) HO 4 N 19/117
HO 4 N 19/157 (2014.01) HO 4 N 19/157
HO 4 N 19/176 (2014.01) HO 4 N 19/176
HO 4 N 19/70 (2014.01) HO 4 N 19/70
HO 4 N 19/82 (2014.01) HO 4 N 19/82

請求項の数 6 (全 43 頁)

(21) 出願番号	特願2013-79864 (P2013-79864)	(73) 特許権者	308036402
(22) 出願日	平成25年4月5日(2013.4.5)		株式会社 J V C ケンウッド
(65) 公開番号	特開2013-232887 (P2013-232887A)		神奈川県横浜市神奈川区守屋町3丁目12番地
(43) 公開日	平成25年11月14日(2013.11.14)	(74) 代理人	100105924
審査請求日	平成27年10月29日(2015.10.29)		弁理士 森下 賢樹
(31) 優先権主張番号	特願2012-87580 (P2012-87580)	(72) 発明者	竹原 英樹
(32) 優先日	平成24年4月6日(2012.4.6)		神奈川県横浜市神奈川区守屋町3丁目12番地
(33) 優先権主張国	日本国(JP)	(72) 発明者	坂爪 智
早期審査対象出願			神奈川県横浜市神奈川区守屋町3丁目12番地
		(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

最終頁に続く

(54) 【発明の名称】 動画像復号装置、動画像復号方法、動画像復号プログラム、受信装置、受信方法及び受信プログラム

(57) 【特許請求の範囲】

【請求項1】

ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームを復号して得られたピクチャをフィルタリングする動画像復号装置であって、

前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記符号列ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得部と、

復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記符号列ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記符号列ストリームから取得するフィルタパラメータ取得部と、

前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号部と、

前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリング部とを備え、

前記フィルタパラメータ取得部は、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記符号列ストリームから取得することを特徴とする動画像復号装置。

【請求項 2】

ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームを復号して得られたピクチャをフィルタリングする動画像復号方法であって、

前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記符号列ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得ステップと、

復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記符号列ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記符号列ストリームから取得するフィルタパラメータ取得ステップと、

前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号ステップと、

前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリングステップとを備え、

前記フィルタパラメータ取得ステップは、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記符号列ストリームから取得することを特徴とする動画像復号方法。

【請求項 3】

ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームを復号して得られたピクチャをフィルタリングする動画像復号プログラムであって、

前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記符号列ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得ステップと、

復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記符号列ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記符号列ストリームから取得するフィルタパラメータ取得ステップと、

前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号ステップと、

前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリングステップとをコンピュータに実行させ、

前記フィルタパラメータ取得ステップは、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記符号列ストリームから取得することを特徴とする動画像復号プログラム。

【請求項 4】

動画像が符号化された符号列ストリームを受信して復号する受信装置であって、

ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームがパケット化された符号化データを受信する受信部と、

受信された前記符号化データをパケット処理して元の符号列ストリームを復元する復元部と、

前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記元の符号化ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得部と、

復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記元の符号化ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記

10

20

30

40

50

隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記元の符号化ストリームから取得するフィルタパラメータ取得部と、

前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号部と、

前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリング部とを備え、

前記フィルタパラメータ取得部は、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記元の符号化ストリームから取得することを特徴とする受信装置。

10

【請求項 5】

動画画が符号化された符号列ストリームを受信して復号する受信方法であって、

ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームがパケット化された符号化データを受信する受信ステップと、

受信された前記符号化データをパケット処理して元の符号列ストリームを復元する復元ステップと、

前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記元の符号化ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得ステップと、

復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記元の符号化ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記元の符号化ストリームから取得するフィルタパラメータ取得ステップと、

20

前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号ステップと、

前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリングステップとを備え、

前記フィルタパラメータ取得ステップは、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記元の符号化ストリームから取得することを特徴とする受信方法。

30

【請求項 6】

動画画が符号化された符号列ストリームを受信して復号する受信プログラムであって、

ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームがパケット化された符号化データを受信する受信ステップと、

受信された前記符号化データをパケット処理して元の符号列ストリームを復元する復元ステップと、

前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記元の符号化ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得ステップと、

40

復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記元の符号化ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記元の符号化ストリームから取得するフィルタパラメータ取得ステップと、

前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号ステップと、

前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタ

50

リングステップとをコンピュータに実行させ、

前記フィルタパラメータ取得ステップは、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記元の符号化ストリームから取得することを特徴とする受信プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、動き補償予測を用いた動画像復号技術に関し、特にループフィルタとして機能するフィルタ情報を復号する動画像復号技術に関する。

【背景技術】

【0002】

一般的な動画像圧縮符号化であるMPEG-4AVC/H.264（以下、AVC）ではループフィルタとしてデブロックフィルタが利用される。AVCのデブロックフィルタは、対象画像を細かいブロックに分割し、ブロック毎に隣接するブロックの境界付近の画素を補正する技術である。

【0003】

デブロックフィルタは、ブロック間の量子化幅や予測方法の相違などによって生じるブロックの境界付近の誤差を画素単位で補正し、補正したピクチャを参照ピクチャとして動き補償予測で利用することで予測効率を向上させている。

【0004】

特許文献1には、AVCのデブロックフィルタを制御する方法が開示されている。

【先行技術文献】

【特許文献】

【0005】

【特許文献1】WO2009/093472号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

AVCのデブロックフィルタはブロックの境界付近の画素しか補正しないため、ブロック境界ではないブロック内部の画素を補正することはできない。そのため、ブロック境界ではないブロック内部の画素に誤差が生じた場合に、予測効率が良くなる場合があった。

【0007】

このような状況下、本発明者らは、ループフィルタを使用する動画像符号化方式において、ブロック境界ではないブロック内部の画素に誤差が生じた場合でも予測効率を向上させる必要性を認識するに至った。

【0008】

本発明はこうした状況に鑑みてなされたものであり、その目的は、ブロック境界ではないブロック内部の画素に誤差が生じた場合にもブロック境界ではないブロック内部の画素を補正することで予測効率の向上を図って符号化効率を向上させる動画像復号技術を提供することにある。

【課題を解決するための手段】

【0009】

上記課題を解決するために、本発明のある態様の動画像復号装置は、ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームを復号して得られたピクチャをフィルタリングする動画像復号装置であって、前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記符号列ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得部（201）と、復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記符号列ストリームか

10

20

30

40

50

ら取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記符号列ストリームから取得するフィルタパラメータ取得部（204）と、前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号部（202）と、前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリング部（203）とを備える。前記フィルタパラメータ取得部（204）は、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記符号列ストリームから取得する。

【0011】

本発明のさらに別の態様は、動画画像復号方法である。この方法は、ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームを復号して得られたピクチャをフィルタリングする動画画像復号方法であって、前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記符号列ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得ステップと、復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記符号列ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記符号列ストリームから取得するフィルタパラメータ取得ステップと、前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号ステップと、前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリングステップとを備える。前記フィルタパラメータ取得ステップは、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記符号列ストリームから取得する。

【0013】

本発明のさらに別の態様は、受信装置である。この装置は、動画画像が符号化された符号列ストリームを受信して復号する受信装置であって、ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームがパケット化された符号化データを受信する受信部と、受信された前記符号化データをパケット処理して元の符号列ストリームを復元する復元部と、前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記元の符号化ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得部（201）と、復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記元の符号化ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記元の符号化ストリームから取得するフィルタパラメータ取得部（204）と、前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号部（202）と、前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリング部（203）とを備える。前記フィルタパラメータ取得部（204）は、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記元の符号化ストリームから取得する。

【0014】

本発明のさらに別の態様は、受信方法である。この方法は、動画画像が符号化された符号列ストリームを受信して復号する受信方法であって、ピクチャを構成するブロック単位で符号化されたブロック符号列を含む符号列ストリームがパケット化された符号化データを受信する受信ステップと、受信された前記符号化データをパケット処理して元の符号列ス

10

20

30

40

50

トリームを復元する復元ステップと、前記ピクチャが、前記ブロック以上の大きさの複数のタイルに分割された分割領域の情報を、前記元の符号化ストリームのピクチャ特性を示すピクチャパラメータセットから取得する領域情報取得ステップと、復号対象となるブロックのフィルタパラメータを前記復号対象となるブロックに隣接する復号済みのブロックから取得するか否かを示す取得フラグを、前記元の符号化ストリームから取得し、前記取得フラグが取得する旨を示す場合、前記フィルタパラメータを前記隣接する復号済みのブロックから取得し、前記取得フラグが取得しない旨を示す場合、前記フィルタパラメータに関する情報を前記元の符号化ストリームから取得するフィルタパラメータ取得ステップと、前記ブロック符号列を復号して前記復号対象となるブロックの復号画像データを取得する画像データ復号ステップと、前記復号画像データを前記フィルタパラメータに基づいてフィルタリングするフィルタリングステップとを備える。前記フィルタパラメータ取得ステップは、前記復号対象となるブロックがタイル境界に接するか否かを検査し、前記復号対象となるブロックがタイル境界に接する場合、前記取得フラグを前記元の符号化ストリームから取得する。

10

【0015】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、記録媒体、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【0016】

20

本発明によれば、ブロック境界ではないブロック内部の画素に誤差が生じた場合にもブロック境界ではないブロック内部の画素を補正することで予測効率の向上を実現することができる。

【図面の簡単な説明】

【0017】

【図1】CTBとCUの関係を説明する図である。

【図2】タイルについて説明する図である。

【図3】第1の実施の形態に係る動画像符号化装置の構成を説明する図である。

【図4】第1の実施の形態に係る動画像符号化装置の動作を説明するフローチャートである。

30

【図5】CTBアドレスとCTB符号順序を説明する図である。

【図6】図3のフィルタパラメータ決定部の構成を説明する図である。

【図7】図3のフィルタパラメータ決定部の動作を説明するフローチャートである。

【図8】図6のエッジタイプフィルタ部の動作を説明するフローチャートである。

【図9】エッジアングルに応じた隣接画素Aと隣接画素Bを説明する図である。

【図10】右辺や下辺がタイル境界に接しているCTBの画素Aと画素Xを説明する図である。

【図11】図6のバンドタイプフィルタ部の動作を説明するフローチャートである。

【図12】符号化対象CTBと符号化対象CTBに隣接するCTBを説明する図である。

【図13】図3のフィルタパラメータ符号化部の動作を説明するフローチャートである。

40

【図14】隣接CTBの有効性の導出を説明するフローチャートである。

【図15】シンタックスを説明する図である。

【図16】シンタックスを説明する図である。

【図17】第1の実施の形態による符号化ストリームの構成の一例を説明する図である。

【図18】第1の実施の形態に係る動画像復号装置の構成を説明する図である。

【図19】第1の実施の形態に係る動画像復号装置の動作を説明するフローチャートである。

【図20】図18のフィルタパラメータ復号部の動作を説明するフローチャートである。

【図21】第2の実施の形態に係る動画像符号化装置の動作を説明するフローチャートである。

50

【図 2 2】第 2 の実施の形態による符号化ストリームの構成の一例を説明する図である。

【図 2 3】第 2 の実施の形態に係る動画像復号装置の動作を説明するフローチャートである。

【図 2 4】第 3 の実施の形態に係る P P S のシンタックスを説明する図である。

【図 2 5】第 3 の実施の形態に係る動画像符号化装置の動作を説明するフローチャートである。

【図 2 6】第 3 の実施の形態に係る動画像復号装置の動作を説明するフローチャートである。

【図 2 7】第 3 の実施の形態のエッジタイプフィルタ部の動作を説明するフローチャートである。

10

【図 2 8】第 3 の実施の形態における隣接 C T B の有効性の導出を説明するフローチャートである。

【図 2 9】第 4 の実施の形態に係る P P S のシンタックスを説明する図である。

【図 3 0】第 4 の実施の形態における隣接 C T B の有効性の導出を説明するフローチャートである。

【図 3 1】第 5 の実施の形態における C T B の位置に応じて利用可能なフィルタタイプを説明するための図である。

【図 3 2】第 5 の実施の形態におけるエッジタイプフィルタ部の動作を説明するフローチャートである。

【図 3 3】第 6 の実施の形態に係る `loop filter_type_idx` の符号列を説明する図である。

20

【図 3 4】第 6 の実施の形態における隣接 C T B の有効性の導出を説明するフローチャートである。

【図 3 5】第 7 の実施の形態に係るループフィルタユニットのシンタックスを説明する図である。

【図 3 6】第 7 の実施の形態に係る `loop filter_type_idx` の符号列を説明する図である

【図 3 7】第 4 の実施の形態に係る動画像復号装置 2 0 0 の動作を説明するフローチャートである。

【発明を実施するための形態】

30

【0 0 1 8】

[第 1 の実施の形態]

まず、本発明の実施の形態の前提となる技術を説明する。

【0 0 1 9】

2 0 0 3 年に、国際標準化機構 (I S O) と国際電気標準会議 (I E C) のジョイント技術委員会 (I S O / I E C) と、国際電気通信連合電気通信標準化部門 (I T U - T) の共同作業によって A V C と呼ばれる動画像符号化方式 (I S O / I E C では 1 4 4 9 6 - 1 0 、 I T U - T では H . 2 6 4 の規格番号がつけられている) が国際標準として制定された。A V C ではループフィルタとしてデブロックフィルタが利用される。

【0 0 2 0】

40

最初に、デブロックフィルタについて説明する。デブロックフィルタは、量子化パラメータやイントラ予測 (画面内予測) であるかインター予測 (動き補償予測) であるかなどの情報によって、デブロックフィルタの強度をブロック単位で適応的に変え、ブロックの境界付近の画素間の画素値の差によってデブロックフィルタの O N / O F F を画素単位で制御しながら、ブロック間の量子化幅や予測方法の相違などによって生じるブロックの境界付近の誤差を画素単位で補正して誤差を低減する技術である。デブロックフィルタは、視覚的に目立ちやすいブロック境界の誤差を低減して主観画質を向上させ、さらに、ブロック境界の誤差を低減した画像を参照画像として用いることにより、インター予測の効率を向上させる効果がある。

【0 0 2 1】

50

デブロックフィルタの制御情報として、A V Cではデブロックフィルタをスライス単位でO N / O F Fするためのフラグ、デブロックフィルタの適用度合い（画素単位のO N / O F F）を調整するための情報がスライスヘッダにある。

【 0 0 2 2 】

近年のハードウェアやC P Uは並列処理を前提とした構成を取ることが一般的となりつつあり、動画像符号化も並列処理に対応することが重要な課題となっている。画像のサイズが大きくなるほどこの課題は大きくなる。そのため、一般的な動画像符号化方式であるA V Cなどでは、動画像ピクチャを複数の領域に分割し、各領域を並列に符号化または復号するためにスライスが用いられる。スライスを並列に処理する場合、各スライス境界のデブロックフィルタの処理が課題となる。A V Cのデブロックフィルタでは、画像の左上のブロックからラスタースキャン順に各ブロックについて垂直方向のフィルタと水平方向のフィルタが順に処理されるため、領域境界のデブロックフィルタを事前に処理することができない。そのため、A V Cではスライス単位でデブロックフィルタをO N / O F Fするためのフラグが導入されており、ループフィルタを含めた並列処理を容易に実現する場合などに利用されている。

10

【 0 0 2 3 】

しかし、A V Cのデブロックフィルタはブロックの境界付近の画素しか補正できないため、ブロック境界ではないブロック内部の画素に誤差が生じた場合に、予測効率が良くならない可能性があった。そこで、ブロック境界のみならず、ブロックの内部の画素を補正するフィルタとしてS A Oフィルタ（S a m p l e A d a p t a t i v e O f f s e t F i l t e r）を導入する。

20

【 0 0 2 4 】

ここで、S A Oフィルタの概要について説明する。S A Oフィルタはブロックの境界付近の画素のみならず、ブロックの内部の全ての画素に対して所定のオフセット値を加え、画素を補正するフィルタである。ブロックの内部の全ての画素に対して処理を行うため、デブロックフィルタのようにブロック間の相違などの情報を利用して詳細な適応処理をすることは困難で、デブロックフィルタよりもより細かい領域単位でオフセット値を調整して性能の向上を図る必要がある。細かい領域単位でオフセット値を調整することで誤差の補正の精度は向上するが、細かい領域単位でオフセット値を符号化するための符号量が増加するため、これを抑制する必要がある。S A Oフィルタの詳細については後述する。

30

【 0 0 2 5 】

以下、S A Oフィルタを効果的に動作させることのできる動画像符号化装置及び動画像復号装置を実現する。

【 0 0 2 6 】

（符号化ツリーブロック）

本実施の形態では、入力された画像信号（ピクチャ）は符号化ツリーブロック単位に分割されて符号化と復号の処理がなされる。符号化ツリーブロック（C T B）は階層的に4分割されて、符号化ユニット（C U）となる。C T B内の各C Uはそれぞれイントラ予測またはインター予測を用いて予測符号化される。

【 0 0 2 7 】

本実施の形態では、C T Bを6 4（画素）× 6 4（画素）、最大の分割回数を3回とする。そのため、C Uの大きさとして、8 × 8、1 6 × 1 6、3 2 × 3 2、6 4 × 6 4が存在する。図1はC T BとC Uの関係を説明する図である。

40

【 0 0 2 8 】

本実施の形態では、C T Bを6 4（画素）× 6 4（画素）、C T Bの最大の分割回数を3回とするが、この組み合わせに限定されない。また、C Uはさらに1つから4つの予測ブロックに分割されて、それぞれの予測ブロックについてイントラ予測またはインター予測を用いて予測符号化されてもよい。

【 0 0 2 9 】

（ピクチャ）

50

ピクチャは符号化と復号の単位となる 1 枚の画像である。ピクチャには、イントラ予測のみを用いた I ピクチャ、イントラ予測と単予測のインター予測を用いた P ピクチャ、イントラ予測と、単予測と双予測のインター予測を用いた B ピクチャがある。

【 0 0 3 0 】

(タイル)

本実施の形態では、ピクチャを垂直方向と水平方向に分割された領域として定義することのできるタイルを導入する。

【 0 0 3 1 】

次に図を用いてタイルについて説明する。図 2 はタイルについて説明する図である。図 2 (a) はピクチャが垂直方向に 2 つに分割されて、ピクチャがタイル 0 とタイル 1 で構成される例である。図 2 (b) はピクチャが垂直方向と水平方向にそれぞれ 2 つに分割されて、ピクチャがタイル 0 からタイル 3 で構成される例である。ピクチャ内のタイルはラスタースキャン順に処理される。また、タイル内の C T B はラスタースキャン順に処理される。ここでは、説明を容易にするためにタイルがピクチャ内で同一の大きさとなるようにしているが、タイルは同一の大きさである必要はない。図 2 (a) と図 2 (b) で示されるように、タイルは矩形領域として定義される。

【 0 0 3 2 】

(スライス)

本実施の形態では、スライスはタイル内の C T B をラスタースキャン順に包含する。ここでは、ピクチャ内のスライスは 1 つであるとして説明するが、スライスは 1 つのタイル内の 1 以上の C T B から構成することができる。すなわち、タイルは 1 以上のスライスに分割されることもできる。また、スライスは複数のタイルを包含することもできる。すなわち、複数のタイルを 1 つのスライスにまとめることもできる。

【 0 0 3 3 】

本実施の形態では、タイルを導入することで、スライスでは実現できなかった垂直方向に領域を分割することができる。垂直方向に領域を分割することで、各タイルの水平画素数を減ずることができ、各タイルを独立処理 (並列処理) する場合に、ラインメモリ量を削減することができる。一方、タイルに分割することで垂直方向の境界が長くなるため、垂直方向の境界の処理が符号化効率や視覚的な違和感などとして認識されて課題となる。これは大画面になるほど影響が大きくなる。また、タイル内の C T B はラスタースキャン順に処理されてスライスはタイル内の C T B をラスタースキャン順に包含する構成とするため、タイルとスライスを組み合わせることで、スライスの垂直方向の境界についても同様の課題が生じる場合がある。なお、スライスのみで領域を構成する場合、垂直方向の境界はスライスの開始 C T B にのみ発生するため、垂直方向の境界に対する課題は小さい。

【 0 0 3 4 】

以下、図面とともに本発明の好適な実施の形態に係る動画像符号化装置、動画像符号化方法および動画像符号化プログラム、ならびに動画像復号装置、動画像復号方法および動画像復号プログラムの詳細について説明する。なお、図面の説明において同一要素には同一符号を付与して重複する説明を省略する。

【 0 0 3 5 】

(動画像符号化装置 1 0 0 の構成)

図 3 は、第 1 の実施の形態に係る動画像符号化装置 1 0 0 の構成を説明する図である。動画像符号化装置 1 0 0 は、入力される動画像を、1 ピクチャ単位で符号化する装置である。

【 0 0 3 6 】

動画像符号化装置 1 0 0 は、CPU (Central Processing Unit)、フレームメモリ、ハードディスクなどを備える情報処理装置などのハードウェアにより実現される。動画像符号化装置 1 0 0 は、上記の構成要素が動作することにより、以下に説明する機能的な構成要素を実現する。

【 0 0 3 7 】

本実施の形態の動画像符号化装置 100 は、画像データ取得部 101、CTB 符号化部 102、フィルタパラメータ決定部 103、フィルタパラメータ符号化部 105、ループフィルタ 104、フレームメモリ 106、符号列多重部 107、及び符号化制御部 110 を含む。符号化制御部 110 は、タイル設定部 111、ループフィルタ設定部 112、及び領域情報設定部 113 を含む。

【0038】

(動画像符号化装置 100 の機能と動作)

図 4 は、第 1 の実施の形態に係る動画像符号化装置 100 の動作を説明するフローチャートである。以下、図 3 と図 4 を用いて、各部の機能と動作について説明する。ここでは、入力されるピクチャのサイズは 768×384 、ビット深度は 8 ビットとする。この場合、CTB は水平方向に 12 個、垂直方向に 6 個存在し、ピクチャ内で 72 個の CTB が存在する。以降、ピクチャの水平方向の CTB 数を `NumCtbInPicWidth` とする。ビット深度が 8 ビットであるため、画素値は 0 から 255 となる。入力されたピクチャは、GOP (Group of Picture) として一般的な $M = 3$ 、 $N = 15$ で符号化され、I ピクチャは I スライスだけ、P ピクチャは P スライスだけ、B ピクチャは B スライスだけで符号化されるものとする。ここでは、ピクチャのサイズを 768×384 として CTB の倍数に設定したがこれに限定されない。また、GOP を $M = 3$ 、 $N = 15$ としたが、I ピクチャだけとなる $N = 1$ などとしてもよく、これらに限定されない。

【0039】

最初に、タイル設定部 111 は、タイルを設定する (S100)。これ以降、特に断らない限り、ピクチャは図 2 (b) のように 4 つのタイルから構成されるものとする。

【0040】

次に、ループフィルタ設定部 112 は、ループフィルタ情報を設定する (S101)。ここで、ループフィルタ情報について説明する。ループフィルタ情報とは、ループフィルタインターリーブフラグ (`loopfilter_interleave_flag`) である。ループフィルタインターリーブフラグは、フィルタパラメータの符号化単位が CTB 単位で多重化するかピクチャ単位で多重化するかを示すフラグである。ループフィルタインターリーブフラグが 1 であれば、フィルタパラメータが CTB 単位で多重化されることを示し、ループフィルタインターリーブフラグが 0 であれば、フィルタパラメータがピクチャ単位で多重化されることを示す。以降、フラグは 0 または 1 の値の 1 ビットの情報であるとする。

【0041】

本実施の形態では、`loopfilter_interleave_flag` は 1 に設定される。つまり、フィルタパラメータは CTB 単位で多重化される。

【0042】

まず、領域情報設定部 113 は、CTB に対して CTB アドレスと CTB 符号順序を設定する (S102)。ここで、CTB アドレスと CTB 符号順序について説明する。図 5 は CTB アドレスと CTB 符号順序を説明する図である。

【0043】

図 5 (a) はピクチャ内の各 CTB に設定される CTB アドレスを示す。CTB アドレスは図 5 (a) のように、ピクチャ内の各 CTB に対してラスタースキャン順に 1 ずつインクリメントして設定される。図 5 (b) は図 2 (b) のようにピクチャが 4 つのタイルから構成される場合のピクチャ内の各 CTB に設定される CTB 符号順序を示す。第 1 のタイルは CTB 符号順序が 0 から 17 であり、第 2 のタイルは CTB 符号順序が 18 から 35 であり、第 3 のタイルは CTB 符号順序が 36 から 53 であり、第 4 のタイルは CTB 符号順序が 54 から 71 である。CTB 符号順序は図 5 (b) のように、タイル内の各 CTB に対してラスタースキャン順に 1 ずつインクリメントして設定される。図 5 で示されるように、タイルが垂直方向に分割される場合、CTB アドレス (CTB のアドレス順序) と、CTB が符号化される順序とが同一とはならないブロックが発生する。また、図 5 で示されるように、CTB アドレスはピクチャに応じたラスタースキャン順で定まり、

C T B 符号順序はタイルに応じたラスターキャン順で定まる。

【 0 0 4 4 】

図 5 で示されるように、C T B アドレスはタイルに依存せずピクチャのサイズによって設定され、C T B 符号順序はピクチャのサイズとタイルのサイズによって設定される。つまり、C T B アドレスはピクチャ内の位置を示し、C T B アドレスによって C T B のピクチャ内での位置関係を得ることができる。C T B がいずれのタイルに属するかを識別するのに、C T B アドレスと C T B 符号順序のそれぞれまたは両方を用いてもよい。

【 0 0 4 5 】

また、符号化制御部 1 1 0 は、S P S、P P S、スライスヘッダなどピクチャを符号化するために必要となる情報を必要に応じて設定して管理する。S P S、P P S、スライスヘッダについては後述する。そして、画像データ取得部 1 0 1、C T B 符号化部 1 0 2、フィルタパラメータ決定部 1 0 3、フィルタパラメータ符号化部 1 0 5、ループフィルタ 1 0 4、フレームメモリ 1 0 6、及び符号列多重部 1 0 7 を制御して符号化ストリームを生成する。

10

【 0 0 4 6 】

なお、タイル、ループフィルタ情報、S P S、P P S、スライスヘッダなどピクチャを符号化するために必要となる情報は動画像符号化装置 1 0 0 内で共有されているものとし、これらデータのフローの説明は特に断らない限り省略する。

【 0 0 4 7 】

次に、C T B 符号順序に従って、ピクチャ内の全ての C T B について以下の処理が繰り返される (S 1 0 3 から S 1 1 1)。

20

【 0 0 4 8 】

次に、画像データ取得部 1 0 1 は、符号化対象となる C T B である符号化対象 C T B のピクチャ内の位置に基づいて、端子 1 0 より入力される画像データから符号化対象 C T B に該当する C T B の原画像データを取得する (S 1 0 4)。そして、画像データ取得部 1 0 1 は C T B の原画像データを C T B 符号化部 1 0 2 に供給する。また、画像データ取得部 1 0 1 は C T B の原画像データをフィルタパラメータ決定部 1 0 3 に供給する。ここで、符号化対象 C T B が属するタイルを符号化対象タイル、符号化対象 C T B が属するスライスを符号化対象スライスとする。

【 0 0 4 9 】

30

次に、エントロピー符号部の設定が行われる (S 1 0 5)。ここで、エントロピー符号部の設定について説明する。符号化対象 C T B がスライスまたはタイルの最初の C T B であれば、C T B 符号化部 1 0 2 は C T B 符号化部 1 0 2 の内部のエントロピー符号部を初期化し、フィルタパラメータ符号化部 1 0 5 はフィルタパラメータ符号化部 1 0 5 の内部のエントロピー符号部を初期化する。なお、C T B 符号化部 1 0 2 の内部のエントロピー符号部とフィルタパラメータ符号化部 1 0 5 の内部のエントロピー符号部は、A V C で利用されている算術符号化の C A B A C であるとする。

【 0 0 5 0 】

次に、C T B 符号化部 1 0 2 は、画像データ取得部 1 0 1 より供給される C T B の原画像データをスライスタイプに基づいて、画面内予測 (イントラ予測) や動き補償予測 (インター予測) を行って誤差信号を算出し、誤差信号について直交変換や量子化処理を用いて符号化し、C T B 符号列を生成するとともに、局部復号を行って復号画像データを生成する (S 1 0 6)。なお、C T B 符号化部 1 0 2 で行われる符号化の処理は、C T B を階層的に符号化して量子化などによって符号化歪が生じる可能性のある方法であればよく、ここでは詳細な説明は省略する。また、局部復号についても一般的な動画像符号化で行われる手法であるためここでは詳細な説明は省略するが、復号画像データは、動画像符号化装置 1 0 0 の出力する符号化ストリームを復号した際に得られる画像データである。C T B 符号化部 1 0 2 は、生成した C T B 符号列を符号列多重部 1 0 7 に供給する。また、C T B 符号化部 1 0 2 は、復号画像データをフィルタパラメータ決定部 1 0 3 とループフィルタ 1 0 4 に供給する。

40

50

【 0 0 5 1 】

次に、フィルタパラメータ決定部 1 0 3 は、画像データ取得部 1 0 1 より供給される C T B の原画像データと C T B 符号化部 1 0 2 より供給される復号画像データに基づいてフィルタパラメータを決定する (S 1 0 7)。フィルタパラメータ決定部 1 0 3 は、フィルタパラメータをフィルタパラメータ符号化部 1 0 5 とループフィルタ 1 0 4 に供給する。フィルタパラメータ決定部 1 0 3 の詳細については後述する。

【 0 0 5 2 】

ループフィルタ 1 0 4 は、フィルタパラメータ決定部 1 0 3 より供給されるフィルタパラメータに基づいて C T B 符号化部 1 0 2 より供給される復号画像データに対してループフィルタを実行する (S 1 0 8)。ループフィルタを実行して得た新たな復号画像データをフレームメモリ 1 0 6 に供給する。

10

【 0 0 5 3 】

次に、フィルタパラメータ符号化部 1 0 5 は、フィルタパラメータ決定部 1 0 3 より供給されるフィルタパラメータをループフィルタユニットに符号化し (S 1 0 9)、ループフィルタユニットを符号列多重部 1 0 7 に供給する。フィルタパラメータ符号化部 1 0 5 とループフィルタユニットについては後述する。

【 0 0 5 4 】

次に、符号列多重部 1 0 7 は、C T B 符号化部 1 0 2 より供給される C T B 符号列とフィルタパラメータ符号化部 1 0 5 より供給されるループフィルタユニットを、必要に応じて S P S、P P S、スライスヘッダなどとともに符号化ストリームとして多重化し (S 1 1 0)、符号化ストリームを端子 1 1 より出力する。符号列多重部 1 0 7 の詳細については後述する。

20

【 0 0 5 5 】

ピクチャ内の全ての C T B について処理が完了すれば、処理を終了する。

【 0 0 5 6 】

(フィルタパラメータ決定部 1 0 3)

続いて、フィルタパラメータ決定部 1 0 3 の詳細な構成について説明する。図 6 はフィルタパラメータ決定部 1 0 3 の構成を説明する図である。図 6 で示されるように、フィルタパラメータ決定部 1 0 3 は、フィルタパラメータ設定部 1 2 0、ループフィルタ実行部 1 2 1、誤差計測部 1 2 2、及びフィルタパラメータ確定部 1 2 3 で構成される。端子 1 2 は C T B 符号化部 1 0 2 に接続され、端子 1 3 は画像データ取得部 1 0 1 に接続され、端子 1 4 はフィルタパラメータ符号化部 1 0 5 とループフィルタ 1 0 4 に接続されている。ループフィルタ実行部 1 2 1 は、フィルタタイプ判定部 1 3 0、エッジタイプフィルタ 1 3 1、バンドタイプフィルタ 1 3 2 を含む。

30

【 0 0 5 7 】

引き続いて、フィルタパラメータ決定部 1 0 3 の動作について説明する。図 7 はフィルタパラメータ決定部 1 0 3 の動作を説明するフローチャートである。以下、図 6 と図 7 を用いて、フィルタパラメータ決定部 1 0 3 の動作について説明する。

【 0 0 5 8 】

最初に、フィルタパラメータ設定部 1 2 0 は、端子 1 2 より供給される復号画像データと端子 1 3 より供給される原画像データに基づいて、フィルタパラメータ候補リストを生成し (S 1 2 0)、生成したフィルタパラメータ候補リストをループフィルタ実行部 1 2 1 に供給する。フィルタパラメータ設定部 1 2 0 の詳細については後述する。

40

【 0 0 5 9 】

次に、フィルタパラメータ候補リストに登録されている全てのフィルタパラメータについて以下の処理が繰り返される (S 1 2 1 から S 1 2 6)。

【 0 0 6 0 】

まず、ループフィルタ実行部 1 2 1 は、端子 1 2 より供給される復号画像データに対してフィルタパラメータに基づいてループフィルタを適用して新たな復号画像データを生成し (S 1 2 2)、当該復号画像データを誤差計測部 1 2 2 に供給する。ループフィルタ実

50

行部 1 2 1 の詳細については後述する。

【 0 0 6 1 】

誤差計測部 1 2 2 は、ループフィルタ実行部 1 2 1 より供給される復号画像データと端子 1 3 より供給される原画像データの誤差評価値を算出する (S 1 2 3)。ここでは、誤差評価値はブロック内の復号画像データと原画像データの絶対差分和 (S A D) とする。

【 0 0 6 2 】

次に、誤差計測部 1 2 2 は、誤差評価値が最小であるか検査する (S 1 2 4)。誤差評価値が最小であれば (S 1 2 4 の Y)、フィルタパラメータをフィルタパラメータ確定部 1 2 3 に供給し、フィルタパラメータ確定部 1 2 3 は当該フィルタパラメータを保持する (S 1 2 5)。

10

【 0 0 6 3 】

誤差評価値が最小でなければ (S 1 2 4 の N)、次のフィルタパラメータが処理される (S 1 2 6)。フィルタパラメータ候補リストに登録されている全てのフィルタパラメータの処理が完了すると、フィルタパラメータ確定部 1 2 3 が保持している誤差評価値が最小であるフィルタパラメータをフィルタパラメータとして確定して (S 1 2 7)、処理を終了する。

【 0 0 6 4 】

ここでは、誤差評価値として S A D を用いたが、フィルタパラメータが算出できればよく、これに限定されず S S D などを用いても良い。また、復号画像データと原画像データの誤差だけでなく、フィルタパラメータを伝送するのに必要な符号量を加味して評価しても良い。

20

【 0 0 6 5 】

(フィルタパラメータ設定部 1 2 0)

続いて、フィルタパラメータ設定部 1 2 0 の詳細について説明する。

【 0 0 6 6 】

最初に、フィルタパラメータについて説明する。フィルタパラメータは、フィルタタイプ、バンド幅、バンド位置、オフセット 0、オフセット 1、オフセット 2、オフセット 3、オフセット 4 で構成される。

【 0 0 6 7 】

フィルタタイプには、フィルタタイプ 0、フィルタタイプ 1、フィルタタイプ 2、フィルタタイプ 3、フィルタタイプ 4、及びフィルタタイプ 5 の 6 つがある。

30

【 0 0 6 8 】

フィルタタイプ 0 はループフィルタを処理しないことを示し、フィルタタイプ 1 はループフィルタがバンドタイプフィルタであることを示し、フィルタタイプ 2 からフィルタタイプ 5 はループフィルタがエッジタイプフィルタであることを示す。

【 0 0 6 9 】

フィルタパラメータ設定部 1 2 0 は、復号画像データと原画像データの誤差評価値が閾値以下であれば、フィルタタイプをフィルタタイプ 0 だけで構成されるフィルタパラメータ候補リストを生成し、誤差評価値が閾値以下でなければ、フィルタパラメータが取り得る全ての組み合わせのフィルタパラメータを含むフィルタパラメータ候補リストを生成する。

40

【 0 0 7 0 】

(ループフィルタ実行部 1 2 1)

続いて、ループフィルタ実行部 1 2 1 の詳細について説明する。ループフィルタ実行部 1 2 1 は、フィルタタイプ判定部 1 3 0 とエッジタイプフィルタ 1 3 1、バンドタイプフィルタ 1 3 2 で構成される。

【 0 0 7 1 】

引き続き、ループフィルタ実行部 1 2 1 の動作について説明する。

最初に、フィルタタイプ判定部 1 3 0 は、フィルタタイプの判定をする。フィルタタイプが 0 であれば、端子 1 2 より供給される復号画像データをフィルタ適用後の復号画像デ

50

ータとして誤差計測部 1 2 2 に出力する。フィルタタイプが 1 であれば、バンドタイプフィルタ 1 3 2 は、端子 1 2 より供給される復号画像データにフィルタパラメータに基づいてバンドタイプフィルタを適用し、フィルタ適用後の復号画像データを誤差計測部 1 2 2 に出力する。フィルタタイプが 2 から 5 であれば、エッジタイプフィルタ 1 3 1 は、端子 1 2 より供給される復号画像データにフィルタパラメータに基づいてエッジタイプフィルタを適用し、フィルタ適用後の復号画像データを誤差計測部 1 2 2 に出力する。

【 0 0 7 2 】

(エッジタイプフィルタ 1 3 1)

続いて、エッジタイプフィルタ 1 3 1 の詳細について説明する。エッジタイプフィルタ 1 3 1 は、エッジアングル設定部 1 4 1、隣接画素設定部 1 4 2、エッジタイプ判定部 1 4 3、エッジオフセット値設定部 1 4 4、およびエッジオフセット値加算部 1 4 5 を含む。

10

【 0 0 7 3 】

図 8 はエッジタイプフィルタ 1 3 1 の動作を説明するフローチャートである。以下、図 8 を用いて、エッジタイプフィルタ 1 3 1 の動作を説明する。

【 0 0 7 4 】

最初に、エッジアングル設定部 1 4 1 は、エッジアングルの判定を行う (S 1 3 0)。エッジアングル n はフィルタタイプ n とする。n は 2 , 3 , 4 , 5 である。次に、符号化ツリーブロック内の 6 4 個の画素 X に対してラスタースキャン順に以下の処理を繰り返して行う (S 1 3 1 から S 1 3 9)。X は 0、1、2、・・・、6 3 である。

20

【 0 0 7 5 】

まず、隣接画素設定部 1 4 2 は、エッジアングルに応じた隣接画素 A と隣接画素 B を設定する (S 1 3 2)。エッジアングルに応じた隣接画素 A と隣接画素 B については後述する。次に、隣接画素設定部 1 4 2 は隣接画素 A または隣接画素 B がピクチャ外であるか検査する (S 1 3 3)。隣接画素 A または隣接画素 B がピクチャ外であれば (S 1 3 3 の Y)、エッジオフセット値設定部 1 4 4 はオフセット値を 0 に設定して (S 1 3 7)、エッジオフセット値加算部 1 4 5 は画素 X にオフセット値を加算して処理画素 X' を算出する (S 1 3 8)。隣接画素 A または隣接画素 B がピクチャ外でなければ (S 1 3 3 の N)、`loopfilter__interleave__flag` が 1 であるか検査する (S 1 4 0)。`loopfilter__interleave__flag` が 1 であれば (S 1 4 0 の Y)、隣接画素設定部 1 4 2 は隣接画素 A または隣接画素 B がタイル外であるか検査する (S 1 3 4)。隣接画素 A または隣接画素 B がタイル外であるとは、隣接画素 A または隣接画素 B が画素 X とは異なるタイルに属する C T B に含まれることである。`loopfilter__interleave__flag` が 1 でなければ (S 1 4 0 の N)、エッジタイプ判定部 1 4 3 は、エッジタイプの判定をする (S 1 3 5)。

30

【 0 0 7 6 】

隣接画素 A または隣接画素 B がタイル外であれば (S 1 3 4 の Y)、エッジオフセット値設定部 1 4 4 はオフセット値を 0 に設定して (S 1 3 7)、エッジオフセット値加算部 1 4 5 は画素 X にオフセット値を加算して処理画素 X' を算出する (S 1 3 8)。隣接画素 A または隣接画素 B がタイル外でなければ (S 1 3 4 の N)、エッジタイプ判定部 1 4 3 は、エッジタイプの判定をする (S 1 3 5)。エッジタイプの判定については後述する。次に、エッジオフセット値設定部 1 4 4 は、オフセット値の設定をする (S 1 3 6)。次に、エッジオフセット値加算部 1 4 5 は、画素 X にオフセット値を加算して処理画素 X' を算出する (S 1 3 8)。処理画素 X' の算出については後述する。

40

【 0 0 7 7 】

(エッジアングルに応じた隣接画素 A と隣接画素 B)

図 9 はエッジアングルに応じた隣接画素 A と隣接画素 B を説明する図である。エッジアングル 1 の場合、隣接画素 A は画素 X の左の画素、隣接画素 B は画素 X の右の画素となる (図 9 (a))。エッジアングル 2 の場合、隣接画素 A は画素 X の上の画素、隣接画素 B は画素 X の下の画素となる (図 9 (b))。エッジアングル 3 の場合、隣接画素 A は画素

50

Xの右上の画素、隣接画素Bは画素Xの左下の画素となる(図9(c))。エッジアングル4の場合、隣接画素Aは画素Xの左上の画素、隣接画素Bは画素Xの右下の画素となる(図9(d))。このように、水平方向、垂直方向、右斜め方向、及び左斜め方向に対してエッジタイプフィルタを作用させることで、各方向に生じた誤差を補正することが可能となる。ここで、エッジアングル1、エッジアングル2、エッジアングル3、及びエッジアングル4はそれぞれ、フィルタタイプ2、フィルタタイプ3、フィルタタイプ4、及びフィルタタイプ5に割り当てられる。

【0078】

以上のように、エッジタイプフィルタ131において、フィルタパラメータがCTB単位で多重化されている(loopfilter_interleave_flagが1)場合、隣接画素Aまたは隣接画素Bがタイル外であれば、オフセット値を0に設定してループフィルタを作用させないようにすることで、ループフィルタの動作をCTB単位で完結させることができる。そのため、図10のようにCTB内の画素で右辺や下辺がタイル境界に接しているCTB内の画素Xにループフィルタを作用させるために必要となるフィルタ適用前の画素Aと画素Xは、タイル境界を跨いで保持される必要がなくなる。

【0079】

(エッジタイプの判定)

エッジタイプは、画素Xと隣接画素Aの差分の符号と、画素Xと隣接画素Bの差分の符号の和を加算した値である符号和SSで決定する。画素mの画素値をP(m)とすると、符号和SSは式(1)で算出される。

$$SS = \text{Sign}(P(X) - P(A)) + \text{Sign}(P(X) - P(B)); \quad \text{式(1)}$$

ここで、Sign(i)は入力値iが0未満であれば-1を、入力値iが0以上であれば1を返す関数である。

【0080】

SSが0、-2、-1、1、2である場合のエッジタイプをそれぞれ0、1、2、3、4とする。エッジタイプの0、1、2、3、4のオフセット値としてオフセット0、オフセット1、オフセット2、オフセット3、オフセット4がそれぞれ設定される。ここで、SSが0である場合は画素X、隣接画素A、及び隣接画素Bが直線的に並んでいるため、誤差はないとしてオフセット0は0に設定する。SSが0でない場合は歪みを補正するためオフセット0からオフセット4のそれぞれを適切な値に設定する。

【0081】

(エッジタイプフィルタの処理画素X'の算出)

画素Xのエッジタイプフィルタ適用後の処理画素X'は式(2)で算出される。

$$P(X') = P(X) + \text{OFFSET}[SS]; \quad \text{式(2)}$$

ここで、OFFSET[m]はオフセットmである(m=0, 1, ..., 4)。

【0082】

以上のように、エッジタイプフィルタでは、水平方向、垂直方向、右斜め方向、及び左斜め方向に対して、隣接画素との差分の符号によって、画素単位で適応的に誤差を補正することが可能となる。

【0083】

(バンドタイプフィルタ132)

続いて、バンドタイプフィルタ132の詳細について説明する。バンドタイプフィルタ132は、バンド幅設定部151、バンド位置設定部152、バンドオフセット値設定部153、およびバンドオフセット値加算部154を含む。

【0084】

図11はバンドタイプフィルタ132の動作を説明するフローチャートである。以下、図11を用いて、バンドタイプフィルタの動作を説明する。

【0085】

最初に、バンド幅設定部151は、バンド幅を設定する(S150)。

10

20

30

40

50

【 0 0 8 6 】

(バンド幅)

ここで、バンド幅について説明する。バンド幅は、入力画像の取り得る画素値の範囲を複数のバンドに分割する際のバンド幅である。本実施の形態では、バンド幅は32とする。本実施の形態では入力画像は8ビットであるため0から255までの画素値の範囲を有するため、0から31(バンド0)、32から63(バンド1)、64から95(バンド2)、96から127(バンド3)、128から159(バンド4)、160から191(バンド5)、192から223(バンド6)、224から255(バンド7)の8つのバンドに分割される。

【 0 0 8 7 】

10

このようにバンド幅を設定して、バンド幅を大きくすることで、広範囲の画素値を一括して補正することができ、バンド幅を小さくすることで、狭範囲の画素値を一括して補正することができる。例えば、あるピクチャ内の画素の取り得る画素値のレンジが広い場合にはバンド幅が大きくなるようにし、あるピクチャ内の画素の取り得る画素値のレンジが狭い場合には、バンド幅が小さくなるようにして、ピクチャの特性に応じて、復号画像を適切に補正することができる。本実施の形態ではバンド幅は固定値として説明しているが、入力画像の取り得る画素値の範囲を1以上のバンドに分割できればよく、16や128でもよい。また、バンド幅をPPSなどに多重化して伝送し、ピクチャ内の画素の取り得る画素値のレンジなどに合わせて適応的に変更することで、効果的にバンドタイプフィルタを作用させることができる。

20

【 0 0 8 8 】

ステップS150に続いて、3つのバンド位置について以下の処理が繰り返される(S151からS158)。ここで、3つのバンド位置について説明する。1番目のバンド位置はフィルタパラメータに含まれるバンド位置である。2番目のバンド位置は1番目のバンド位置を1つだけずらしたバンド位置となる。例えば、1番目のバンド位置がバンド2であったとすると、2番目のバンド位置はバンド3となる。同様に、3番目のバンド位置は2番目のバンド位置を1つだけずらしたバンド位置となる。

【 0 0 8 9 】

バンド位置設定部152は、b番目のバンド位置を設定する(S152)。ここで、bは0と1である。バンド位置は、バンド幅によって分割されたバンドのいずれかを示す値であって、ここでは、0から7までの整数となる。例えば、バンド位置の0はバンド0を示し、バンド位置の2はバンド3を示す。

30

【 0 0 9 0 】

ステップS152に続いて、バンドオフセット値設定部153は、バンド毎のオフセット値を設定する(S153)。バンド位置で示されたバンドのオフセット値はオフセット0に設定されて、バンド位置で示されたバンド以外のバンドのオフセット値は0に設定される。

【 0 0 9 1 】

次に、画素値の符号化ツリーブロック内の64個の画素Xに対してラスタースキャン順に以下の処理が行なわれる(S154からS157)。Xは0、1、・・・、63である。

40

【 0 0 9 2 】

まず、バンドオフセット値加算部154は、画素Xの属するバンドであるx bを決定する(S155)。次に、バンドオフセット値加算部154は、画素Xに画素Xが属するバンドのオフセット値Ob[x b]を加算して処理画素X'を算出する(S156)。

【 0 0 9 3 】

(バンドタイプフィルタの処理画素X'の算出)

画素Xのバンドタイプフィルタ適用後の処理画素X'は式(3)で算出される。

$$P(X') = P(X) + Ob[xb]; \text{式(3)}$$

【 0 0 9 4 】

50

以上のように、バンドタイプフィルタは特定範囲内の画素値の画素だけを補正することができるため、平坦領域に発生し易いノイズなどを低減する効果がある。特に、4:2:0フォーマットなど輝度信号より解像度を減らした色差信号については、輝度信号よりもDCに誤差が発生し易くエッジが少ないため、バンドタイプフィルタの効果が大きい。また、隣接画素を参照しないためエッジタイプのオフセットよりもフィルタリングの処理量やメモリ量を少なく実現することができる。

【0095】

(ループフィルタ104)

続いて、ループフィルタ104について説明する。ループフィルタ104は、ループフィルタ実行部121と同一の機能と動作を有するため、ここでの説明は省略する。

10

【0096】

(フィルタパラメータ符号化部105)

続いて、フィルタパラメータ符号化部105の詳細について説明する。図12は符号化対象CTBと符号化対象CTBに隣接するCTBを説明する図である。図13はフィルタパラメータ符号化部105の動作を説明するフローチャートである。以下、図12と図13を用いて、フィルタパラメータ符号化部105の動作を説明する。

【0097】

最初に、隣接CTBの有効性である`available_left_ctb_flag`と`available_above_ctb_flag`を導出する(S160)。次に、`available_left_ctb_flag`が1であるか検査する(S161)。`available_left_ctb_flag`が1であれば(S161のY)、符号化対象CTBと符号化対象CTBの左に位置するCTBのフィルタパラメータが同一であるか検査する(S162)。ここで、符号化対象CTBの左に位置するCTBは図12のCTBのAである。符号化対象CTBと符号化対象CTBの左に位置するCTBのフィルタパラメータが同一であれば(S162のY)、`loopfilter_merge_left_flag`を1として`loopfilter_merge_left_flag`を符号化し(S163)、処理を終了する。符号化対象CTBと符号化対象CTBの左に位置するCTBのフィルタパラメータが同一でなければ(S162のN)、`loopfilter_merge_left_flag`を0として`loopfilter_merge_left_flag`を符号化する(S164)。

20

30

【0098】

`available_left_ctb_flag`が1でない場合と(S161のN)、ステップS164に続いて、`available_above_ctb_flag`が1であるか検査する(S165)。`available_above_ctb_flag`が1であれば(S165のY)、符号化対象CTBと符号化対象CTBの上に位置するCTBのフィルタパラメータが同一であるか検査する(S166)。ここで、符号化対象CTBの上に位置するCTBは図12のCTBのBである。符号化対象CTBと符号化対象CTBの上に位置するCTBのフィルタパラメータが同一であれば(S166のY)、`loopfilter_merge_above_flag`を1として`loopfilter_merge_above_flag`を符号化し(S167)、処理を終了する。符号化対象CTBと符号化対象CTBの上に位置するCTBのフィルタパラメータが同一でなければ(S166のN)、`loopfilter_merge_above_flag`を0として`loopfilter_merge_above_flag`を符号化する(S168)。ステップS168に続いて、ループフィルタ符号化パラメータである`loopfilter_cod_param()`を符号化する。ここで、ループフィルタ符号化パラメータはフィルタパラメータの全ての要素を符号化した符号列である。

40

【0099】

なお、フィルタパラメータ符号化部105は、符号化対象CTBの左と上に位置するCTBのフィルタパラメータを参照するため、ピクチャ内の全てのCTBのフィルタパラメータを保持しているものとする。

50

【 0 1 0 0 】

ここで、C T B 単位でフィルタパラメータを符号化する場合、全てのフィルタパラメータをループフィルタ符号化パラメータとして符号化すると、フィルタパラメータの符号化に膨大な符号量が必要となって、画像データを符号化したC T B 符号列に十分に符号量を割り当てることができなくなり、結果的に符号化効率が低下する可能性がある。また、同一オブジェクトに属するC T B に対して異なるフィルタパラメータを用いた場合、同一オブジェクト内でのループフィルタの特性の相違がブロック歪みとして視覚的に認識される可能性がある。そのため、一般的には同一オブジェクト内ではフィルタパラメータは同一になるようにフィルタパラメータが設定される可能性が高い。よって、以上のように、符号化対象C T B のフィルタパラメータが符号化対象C T B に隣接する符号化済みのC T B のフィルタパラメータと同一である場合には、符号化対象C T B のフィルタパラメータとして符号化対象C T B に隣接する符号化済みのC T B のフィルタパラメータを利用することを示す1ビットのフラグを符号化することで、視覚的な劣化を伴うことなく、ループフィルタ符号化パラメータに要する符号量を1ビットに削減して、符号化効率を向上させることができる。なお、ループフィルタ符号化パラメータの詳細については後述するが、ループフィルタ符号化パラメータの符号化に要する符号量は1ビットよりも十分に大きな符号量となる。一方、符号化対象C T B のフィルタパラメータが符号化対象C T B に隣接する符号化済みのC T B のフィルタパラメータと同一でない場合には、符号化対象C T B のフィルタパラメータとして符号化対象C T B に隣接する符号化済みのC T B のフィルタパラメータを利用しないことを示す1ビットのフラグを符号化し、ループフィルタ符号化パラメータを符号化することで、C T B 毎にフィルタパラメータを変更できる。

【 0 1 0 1 】

ここで、隣接C T B の有効性である `available__left__ctb__flag` と `available__above__ctb__flag` の導出について説明する。図14は隣接C T B の有効性の導出を説明するフローチャートである。最初に、`available__left__ctb__flag` と `available__above__ctb__flag` は1に設定される(S180)。次に、符号化対象C T B の左辺がピクチャ境界に接しているか検査する(S181)。符号化対象C T B の左辺がピクチャ境界に接していれば(S181のY)、`available__left__ctb__flag` を0とする(S184)。符号化対象C T B の左辺がピクチャ境界に接していなければ(S181のN)、符号化対象C T B の左辺がタイル境界に接しているか検査する(S182)。ここで、符号化対象C T B の左辺がタイル境界に接しているとは、符号化対象C T B と符号化対象C T B の左に位置するC T B が別のタイルに属することである。一方、符号化対象C T B の左辺がタイル境界に接していないとは、符号化対象C T B と、符号化対象C T B の左に位置するC T B とが同一のタイルに属することである。符号化対象C T B の左辺がタイル境界に接していれば(S182のY)、`loopfilter__interleave__flag` が1であるか検査する(S183)。`loopfilter__interleave__flag` が1であれば(S183のY)、`available__left__ctb__flag` を0とする(S184)。`loopfilter__interleave__flag` が1でない場合(S183のN)は、ステップS184はスキップされる。符号化対象C T B の左辺がタイル境界に接していない場合(S182のN)は、ステップS183とステップS184はスキップされる。

【 0 1 0 2 】

次に、符号化対象C T B の上辺がピクチャ境界に接しているか検査する(S185)。符号化対象C T B の上辺がピクチャ境界に接していれば(S185のY)、`available__above__ctb__flag` を0とする(S188)。符号化対象C T B の上辺がピクチャ境界に接していなければ(S185のN)、符号化対象C T B の上辺がタイル境界に接しているか検査する(S186)。ここで、符号化対象C T B の上辺がタイル境界に接しているとは、符号化対象C T B と符号化対象C T B の上に位置するC T B が別のタイルに属することである。一方、符号化対象C T B の上辺がタイル境界に接していな

いは、符号化対象CTBと、符号化対象CTBの上に位置するCTBとが同一のタイルに属することである。符号化対象CTBの上辺がタイル境界に接していれば(S186のY)、`loopfilter_interleave_flag`が1であるか検査する(S187)。`loopfilter_interleave_flag`が1であれば(S187のY)、`available_above_ctb_flag`を0とする(S188)。`loopfilter_interleave_flag`が1でない場合(S187のN)は、ステップS188はスキップされる。符号化対象CTBの上辺がタイル境界に接していない場合(S186のN)は、ステップS187とステップS188はスキップされる。

【0103】

ここで、フィルタパラメータがCTB単位で多重化され(`loopfilter_interleave_flag`が1)、符号化処理がタイル単位で独立(並列)に行なわれるような場合に、符号化対象CTBの左辺がタイル境界に接している符号化対象CTBについて、左に位置するCTBからフィルタパラメータを取得することを許可すると、左に位置するCTBのフィルタパラメータが確定されるまで、符号化対象CTBとフィルタパラメータと左に位置するCTBのフィルタパラメータとが同一であるか検査することができないため、待ち時間が生じる可能性がある。そこで、CTBの左辺がタイル境界に接しているような符号化対象CTBについては、左に位置するCTBのフィルタパラメータの利用を禁止する。さらに、`available_left_ctb_flag`を0として`loopfilter_merge_left_flag`を符号化しないようにすることで、`loopfilter_merge_left_flag`の符号量を削減し、符号化効率の低下を抑制する。また、`loopfilter_merge_left_flag`を符号化しないようにすることで、明示的に左に位置するCTBのフィルタパラメータの利用を禁止することができる。同様に、符号化対象CTBの上辺がタイル境界に接している場合に、`available_above_ctb_flag`を0として`loopfilter_merge_above_flag`を符号化しないようにすることで符号化効率の低下を抑制し、明示的に上に位置するCTBのフィルタパラメータの利用を禁止することができる。

【0104】

(シンタックス)

続いて、本実施の形態に係るシンタックスについて説明する。符号化においてはシンタックスに含まれる各要素が符号化ストリーム中に符号化される。一方、復号においては、シンタックスに含まれる各要素が符号化ストリームから復号されて、符号化において符号化された要素と同じ値を取得することができる。これ以降、特に断らない限り、シンタックスを構成する各要素は固定長ビットで符号列となる。図15と図16はシンタックスを説明する図である。以下、図15と図16を用いて、シンタックスについて説明する。ここでは、説明を容易にするためにシンタックスを構成する各要素は固定長ビットで符号列となとしたが、可変長ビットとしてもよい。

【0105】

最初に、SPS(Sequence Parameter Set)について説明する。SPSはシーケンス(符号化ストリーム)の特性を決定するためのパラメータ群を定義するパラメータセットである。ピクチャのサイズ、ビット深度、CTBのサイズ、CTBの分割回数などが定義される。

【0106】

続いて、PPS(Picture Parameter Set)について説明する。図15(a)はPPSのシンタックスの一例を説明する図である。PPSはピクチャの特性を決定するためのパラメータ群を定義するパラメータセットである。PPSには、`loopfilter_interleave_flag`、`pic_loopfilter_param()`、`tiles_coding_flag`、`num_tile_columns_minus1`、`num_tile_rows_minus1`、`uniform_s`

10

20

30

40

50

`padding_flag`、`column_width`、`column_height`がシンタックスに従って設置され、動画像符号化において符号化され、動画像復号において復号される。

【0107】

PPSのシンタックス要素である`tiles_coding_flag`、`num_tile_columns_minus1`、`num_tile_rows_minus1`、`uniform_spacing_flag`、`column_width`、`column_height`はタイル情報である。`tiles_coding_flag`は、1であればピクチャは複数のタイルで構成され、0であればピクチャはタイルとして構成されないことを示す。`num_tile_columns_minus1`と`num_tile_rows_minus1`はそれぞれタイルを垂直方向と水平方向に分割する数を示す。`uniform_spacing_flag`はピクチャを予め定められた方法で分割するか否かを決定する情報である。ここでは、予め定められた方法とは、ピクチャを同じ大きさのタイルに分割することを示す。`column_width[i]`と`column_height[i]`はそれぞれ*i*番目のタイルの幅と高さを示す。図2(b)の場合は、`num_tile_columns_minus1`と`num_tile_rows_minus1`は1、`uniform_spacing_flag`は1として符号化される。なお、シーケンス単位でタイルの構成が変更されない場合、これらタイル情報のシンタックス要素をSPSに設置することもできる。

【0108】

(スライスヘッダ)

続いて、スライスヘッダについて説明する。図15(b)はスライスヘッダのシンタックスの一例を説明する図である。スライスヘッダはスライスの特性を決定するためのパラメータ群を定義するヘッダ情報である。スライスヘッダには、`slice_address`、`slice_type`、`num_of_tiles`、`tile_position`がシンタックスに従って設置され、動画像符号化において符号化され、動画像復号において復号される。

【0109】

`slice_address`(スライスアドレス)は、スライスに含まれる最初のCTBのCTBアドレスを示す。`slice_type`(スライスタイプ)は、イントラ予測のみを用いたIスライス、イントラ予測と単予測のインター予測(動き補償予測)を用いたPスライス、イントラ予測と、単予測と双予測のインター予測を用いたBスライスを示す値である。`num_of_tiles`はスライスに含まれるタイル数を示す。`tile_position`はタイルの先頭バイトのスライスヘッダの先頭バイトからのバイト数を示す。

【0110】

(ループフィルタに関するシンタックス)

図16(a)はPPS内のピクチャループフィルタパラメータ(`pic_loop_filter_param()`)のシンタックスの一例を説明する図である。ピクチャループフィルタパラメータには、`loop_filter_unit()`がシンタックスに従って設置され、動画像符号化において符号化され、動画像復号において復号される。`num_of_ctbs`はピクチャ内の総CTB数であって、PPS内にはピクチャ内の全てのCTBの`loop_filter_unit()`が格納される。

【0111】

図16(b)はループフィルタ符号化パラメータ(`loop_filter_cod_param()`)のシンタックスの一例を説明する図である。ループフィルタ符号化パラメータには、`loop_filter_type_idx`、`loop_filter_band_position`、`loop_filter_offset`がシンタックスに従って設置され、動画像符号化において符号化され、動画像復号において復号される。

【0112】

10

20

30

40

50

`loopfilter_type_idx`はフィルタパラメータのフィルタタイプを示すインデックスであって、0から5の整数である。`loopfilter_type_idx`は3ビットの符号列である。`loopfilter_band_position`はフィルタパラメータのバンド位置であって、0から7までの整数を示す3ビットの符号列である。`loopfilter_offset`はフィルタパラメータのオフセットであって、`loopfilter_type_idx`が1の場合は-32から31までの整数であって6ビットの符号列となり、`loopfilter_type_idx`が2から5の場合は0から31までの整数であって5ビットの符号列となる。

【0113】

ここで、エッジタイプフィルタに要する符号量は`loopfilter_type_idx`の3ビットと`loopfilter_offset`の5ビットが4個であり、23ビットとなる。バンドタイプフィルタに要する符号量は`loopfilter_type_idx`の3ビットと`loopfilter_band_position`の3ビットと`loopfilter_offset`の6ビットが3個であり、24ビットとなる。

【0114】

ここで、本実施の形態ではエッジタイプフィルタとバンドタイプフィルタのオフセット数をそれぞれ4個と3個に設定している。これは、バンドタイプフィルタのフィルタパラメータはエッジタイプフィルタのフィルタパラメータよりも1個あたりの符号量が大きいいため、エッジタイプフィルタとバンドタイプフィルタのそれぞれに要する符号量を均衡させるためである。エッジタイプフィルタとバンドタイプフィルタのそれぞれに要する符号量を均衡させることで、エッジタイプフィルタとバンドタイプフィルタのそれぞれのループフィルタ符号化パラメータの符号化時間と復号時間を均衡させることができ、ハードウェアの回路設計やソフトウェアの設計を容易にすることができる。ここでは、エッジタイプフィルタとバンドタイプフィルタのオフセット数をそれぞれ4個と3個に設定したが、例えば、エッジタイプフィルタとバンドタイプフィルタのオフセット数をそれぞれ2個と1個などのように設定してもよい。

【0115】

図16(c)はループフィルタユニット(`loopfilter_unit()`)のシンタックスの一例を説明する図である。ループフィルタユニットには、`loopfilter_merge_left_flag`、`loopfilter_merge_above_flag`、`loopfilter_cod_param()`がシンタックスに従って設置され、動画像符号化において符号化され、動画像復号において復号される。

【0116】

(符号列多重部107の機能と動作)

以下、符号列多重部107によって多重化される符号化ストリームの構成について説明する。図17は第1の実施の形態による符号化ストリームの構成の一例を説明する図である。図17は`loopfilter_interleave_flag`が1である場合の符号化ストリームの構成を示す。SPSは符号列多重部107によって符号化されて、符号化ストリームの先頭に多重化される。PPSは符号列多重部107によって符号化されて、ピクチャの先頭に多重化される。スライスヘッダは符号列多重部107によって符号化されて、スライスの先頭に多重化される。スライスヘッダに続いて、そのスライスに属するCTBについてCTB符号順序に以下の処理が繰り返し行われる。まず、CTBのループフィルタユニットが多重化されて、その後にCTB符号列が多重化される。

【0117】

ここで、SPS、PPS、スライスヘッダには一般的な同期コードが付与されて、SPS、PPS、スライスヘッダは符号化ストリーム中から分離できるようになる。また、タイルにはヘッダが付与されないが、スライスヘッダの`tile_position`によって符号化ストリーム中から分離できる。

【0118】

以上のように、`loopfilter_interleave_flag`を1に設定し

てCTB符号順序にCTBのフィルタパラメータをCTB単位で多重化することで、フィルタパラメータの伝送遅延を抑制した動画像符号化装置を実現することができる。

【0119】

(動画像復号装置200の構成)

図18は、第1の実施の形態に係る動画像復号装置200の構成を説明する図である。動画像復号装置200は、動画像符号化装置100によって生成された符号化ストリームが入力されて、ピクチャ単位で復号ピクチャを再生する装置である。

【0120】

動画像復号装置200は、CPU(Central Processing Unit)、フレームメモリ、ハードディスクなどを備える情報処理装置などのハードウェアにより実現される。動画像復号装置200は、上記の構成要素が動作することにより、以下に説明する機能的な構成要素を実現する。

【0121】

本実施の形態の動画像復号装置200は、符号列分離部201、CTB復号部202、ループフィルタ203、フィルタパラメータ復号部204、フレームメモリ205、及び復号制御部210を含む。

【0122】

(動画像復号装置200の機能と動作)

復号制御部210は、領域情報、ループフィルタ情報、SPS、PPS、スライスヘッダなどピクチャを復号するために必要となる情報を管理して、符号列分離部201、CTB復号部202、ループフィルタ203、フィルタパラメータ復号部204、フレームメモリ205を制御して符号化ストリームを復号する。なお、領域情報、ループフィルタ情報、SPS、PPS、スライスヘッダなどピクチャを復号するために必要となる情報は動画像復号装置200内で共有されているものとし、これらデータのフローの説明は省略する。

【0123】

図19は、第1の実施の形態に係る動画像復号装置200の動作を説明するフローチャートである。図18はピクチャ単位の復号動作を示す。以下、図18と図19を用いて、各部の機能と動作について説明する。

【0124】

本実施の形態では、loopfilter__interleave__flagは1に設定された符号化ストリームが入力されるとする。

【0125】

最初に、符号列分離部201は、端子20より入力される符号化ストリームからピクチャストリームを取得する(S500)。ここで、ピクチャストリームとはピクチャ単位の符号化ストリームであり、ピクチャストリームの最初のCTBの符号列の前にSPSが存在する場合にはSPSも含まれる。

【0126】

次に、符号列分離部201は、タイルストリームにSPSが含まれていればSPSを復号し、ピクチャのサイズを取得する。また、図15のシンタックスに従ってPPSを復号し、ループフィルタ情報とタイル情報を取得する(S501)。

【0127】

復号制御部210は、ピクチャのサイズとタイル情報から領域情報を導出する。ここで、領域情報は動画像符号化装置100で設定された領域情報と同じ情報が導出されるものとする。なお、ピクチャのサイズとタイル情報によって設定されるCTBアドレス(CTBのアドレス順序)と、CTBが符号化される順序との関係も動画像符号化装置100で設定された動画像符号化装置100で設定された関係と同一となる。

【0128】

次に、ピクチャストリームに含まれる全てのCTBについて、以下の処理が繰り返し行われる(S502からS508)。

【0129】

まず、スライスヘッダがあれば、図15のシンタックスに従ってスライスヘッダを復号してスライスアドレスを取得し、スライスアドレスをCTBアドレスとし、領域情報からCTBアドレスに該当するCTB符号順序を得る。スライスヘッダがなければCTB符号順序を1だけインクリメントし、領域情報からCTB符号順序に該当するCTBアドレスを得る(S503)。ここで、CTBアドレスは、CTB符号順序から図5に示すCTBアドレスとCTB符号順序の関係に基づいて導出される。例えば、CTB符号順序が8のCTBのCTBアドレスは14となる。次のCTBではCTB符号順序が1だけインクリメントされて9となり、CTBアドレスは15となる。以上のように、CTB符号順序からCTBアドレスとCTB符号順序の関係に基づいてCTBアドレスを得る。

10

【0130】

次に、CTB復号部202は、CTB復号部202内部のエントロピー復号部の設定をする(S504)。ここで、CTB復号部202の内部のエントロピー復号部は、動画像符号化装置100によって生成された符号列を復号可能なCABA Cであって、スライスまたはタイルの開始において初期化される。

【0131】

次に、符号列分離部201は復号対象CTBのループフィルタユニットをフィルタパラメータ復号部204に供給し、フィルタパラメータ復号部204はループフィルタユニットをフィルタパラメータに復号して(S505)、当該フィルタパラメータをループフィルタ203に供給する。フィルタパラメータ復号部204の詳細については後述する。次に、符号列分離部201は復号対象CTBのCTB符号列をCTB復号部202に供給し、CTB復号部202は復号対象CTBのCTB符号列を復号して、復号対象CTBの復号画像データを取得して(S506)、当該復号画像データをループフィルタ203に供給する。

20

【0132】

次に、ループフィルタ203は、CTB復号部202より供給される復号画像データに対して、符号列分離部201より供給されるフィルタパラメータに基づいてループフィルタを実行し(S507)、新たな復号画像データを生成して、当該復号画像データをフレームメモリ205に供給する。ループフィルタ203は動画像符号化装置100のループフィルタ104と同一の機能を有するため、説明は省略する。

30

【0133】

(フィルタパラメータ復号部204の機能と動作)

続いて、フィルタパラメータ復号部204の機能と動作について説明する。

図20はフィルタパラメータ復号部204の動作を説明するフローチャートである。以下、図20を用いて、フィルタパラメータ復号部204の機能と動作について説明する。

【0134】

最初に、フィルタパラメータ復号部204は、`available_left_ctb_flag`と`available_above_ctb_flag`を導出する(S510)。ここで、`available_left_ctb_flag`と`available_above_ctb_flag`の導出方法は図14で説明した処理において「符号化対象CTB」を「復号対象CTB」に置き換えたものと同一である。

40

【0135】

次に、`available_left_ctb_flag`が1であるか検査する(S511)。`available_left_ctb_flag`が1であれば(S511のY)、`loopfilter_merge_left_flag`を復号する(S512)。次に、`loopfilter_merge_left_flag`が1であるか検査する(S513)。`loopfilter_merge_left_flag`が1であれば(S513のY)、復号対象CTBの左に位置するCTBのフィルタパラメータを復号対象CTBのフィルタパラメータとして(S514)、処理を終了する。

【0136】

50

available_left_ctb_flag が 1 でないか (S511 の N)、loopfilter_merge_left_flag が 1 でなければ (S513 の N)、available_above_ctb_flag が 1 であるか検査する (S515)。available_above_ctb_flag が 1 であれば (S515 の Y)、loopfilter_merge_above_flag を復号する (S516)。次に、loopfilter_merge_above_flag が 1 であるか検査する (S517)。loopfilter_merge_above_flag が 1 であれば (S517 の Y)、復号対象 CTB の上に位置する CTB のフィルタパラメータを復号対象 CTB のフィルタパラメータとして (S518)、処理を終了する。available_above_ctb_flag が 1 でないか (S515 の N)、loopfilter_merge_above_flag が 1 でなければ (S517 の N)、loopfilter_cod_param() を復号して (S519)、処理を終了する。

10

【0137】

以上のように、loopfilter_interleave_flag を 1 に設定して CTB 符号順序に CTB のフィルタパラメータを CTB 単位で分離して復号し、CTB 単位でループフィルタを実行することで、ピクチャ内の全ての CTB の復号処理の完了直後に復号画像データを出力することのできる低遅延な動画像復号装置を実現することができる。なお、フィルタパラメータ復号部 204 は、復号対象 CTB の左と上に位置する CTB のフィルタパラメータを参照するため、ピクチャ内の全ての CTB のフィルタパラメータを保持しているものとする。

20

【0138】

また、復号対象 CTB の左辺がタイル境界に接している場合、復号対象 CTB のフィルタパラメータとして復号対象 CTB の左に位置する復号済みの CTB のフィルタパラメータを利用すると、復号対象 CTB の左辺が接している CTB の処理が終了するまで復号対象 CTB のフィルタパラメータが確定できなくなる。そこで、復号対象 CTB の左辺がタイル境界に接している場合、復号対象 CTB のフィルタパラメータとして復号対象 CTB の左に位置する復号済みの CTB のフィルタパラメータを利用するか否かを示すフラグである loopfilter_merge_left_flag を復号しないことで、復号対象 CTB の左に位置する復号済みの CTB のフィルタパラメータを利用不能とする。このようにすることで、復号対象 CTB の左辺がタイル境界に接している場合でも、復号対象 CTB のループフィルタユニットを復号した直後にフィルタパラメータを得ることを可能とする。そして、復号対象 CTB の左辺が接している CTB の処理の終了を待つことなく、ループフィルタ 203 の処理を CTB 単位で動作させて、ループフィルタ適用後の復号画像データの出力遅延を最小限に低減することができる。

30

【0139】

同様に、復号対象 CTB の上辺がタイル境界に接している場合、復号対象 CTB のフィルタパラメータとして復号対象 CTB の上に位置する復号済みの CTB のフィルタパラメータを利用するか否かを示すフラグである loopfilter_merge_above_flag を復号しないことで、復号対象 CTB の上に位置する復号済みの CTB のフィルタパラメータを利用不能とする。このようにすることで、復号対象 CTB の上辺がタイル境界に接している場合でも、復号対象 CTB のループフィルタユニットを復号した直後にフィルタパラメータを得ることを可能とする。そして、復号対象 CTB の上辺が接している CTB の処理の終了を待つことなく、ループフィルタ 203 の処理を CTB 単位で動作させて、ループフィルタ適用後の復号画像データの出力遅延を最小限に低減することができる。

40

【0140】

[第 2 の実施の形態]

次に、第 2 の実施の形態について説明する。第 2 の実施の形態は第 1 の実施の形態とは loopfilter_interleave_flag が 0 に設定されることが異なる。つまり、フィルタパラメータはピクチャ単位で符号化される。

50

以下、第１の実施の形態とは異なる点について説明する。

【０１４１】

最初に、本実施の形態に係る動画像符号化装置１００の動作を説明する。図２１は、第２の実施の形態に係る動画像符号化装置１００の動作を説明するフローチャートである。図２１に示すように、ステップＳ１０７からステップＳ１１０がピクチャ内の全てのＣＴＢの符号化処理がなされた後に実行される。ＣＴＢアドレスの順序に従って、ピクチャ内の全てのＣＴＢについてステップＳ１０７、ステップＳ１０８、ステップＳ１０９の処理が繰り返し行われる（Ｓ１１２からＳ１１３）。

【０１４２】

次に、ピクチャの全てのＣＴＢについて、符号列多重部１０７はＣＴＢ符号化部１０２より供給されるＣＴＢ符号列とフィルタパラメータ符号化部１０５より供給されるループフィルタユニットを、必要に応じてＳＰＳ、ＰＰＳ、スライスヘッダなどとともに符号化ストリームとして多重化する（Ｓ１１０）。ここで、符号列多重部１０７によって多重化される符号化ストリームの構成について説明する。図２２は第２の実施の形態による符号化ストリームの構成の一例を説明する図である。図２２は`loopfilter__interleave__flag`が０である場合の符号化ストリームの構成を示す。ＳＰＳは符号化ストリームの先頭に多重化される。ＰＰＳはピクチャの先頭に多重化される。このとき、ＰＰＳにはループフィルタユニットをピクチャ単位でまとめたピクチャループフィルタパラメータが多重化される。スライスの先頭にはスライスヘッダが多重化される。スライスヘッダに続いて、そのスライスに属するＣＴＢについてＣＴＢ符号順序にＣＴＢ符号列が多重化される。

【０１４３】

また、ステップＳ１０５のエントロピー符号部の設定の動作が異なる。本実施の形態では、符号化対象ＣＴＢがスライスまたはタイルの最初のＣＴＢである場合、フィルタパラメータ符号化部１０５はフィルタパラメータ符号化部１０５の内部のエントロピー符号部を初期化しない。

【０１４４】

ここで、フィルタパラメータの符号化について、第１の実施の形態では`loopfilter__interleave__flag`が１に設定されていたため、符号化対象ＣＴＢの左辺がタイル境界に接している場合、`available__left__ctb__flag`が０に設定されたが、第２の実施の形態では`loopfilter__interleave__flag`が０に設定されるため、符号化対象ＣＴＢの左辺がタイル境界に接している場合でも、`available__left__ctb__flag`が１に設定される。そのため、符号化対象ＣＴＢの左辺がタイル境界に接している場合でも`loopfilter__merge__left__flag`を符号化することが可能となっており、符号化効率を向上させることができる。同様に符号化対象ＣＴＢの上辺がタイル境界に接している場合でも`loopfilter__merge__left__flag`を符号化することが可能となっており、符号化効率を向上させることができる。

【０１４５】

また、`loopfilter__interleave__flag`を０に設定してフィルタパラメータをピクチャ単位でまとめて多重化することで、フィルタパラメータの符号量を抑制した符号化装置を実現することができる。

【０１４６】

続いて、本実施の形態に係る動画像復号装置２００の動作を説明する。図２３は第２の実施の形態に係る動画像復号装置２００の動作を説明するフローチャートである。図２３に示すように、ステップＳ５０５とステップＳ５０７がピクチャ内の全てのＣＴＢの復号処理がなされた後に実行される。ＣＴＢアドレスの順序に従って、ピクチャ内の全てのＣＴＢについてステップＳ５０５、ステップＳ５０７の処理が繰り返し行われる（Ｓ５１０からＳ５１１）。

【０１４７】

10

20

30

40

50

ここで、本実施の形態におけるステップ S 5 0 5 について説明する。符号列分離部 2 0 1 はピクチャループフィルタパラメータ中の復号対象 C T B のループフィルタユニットをフィルタパラメータ復号部 2 0 4 に供給し、フィルタパラメータ復号部 2 0 4 はループフィルタユニットをフィルタパラメータに復号して (S 5 0 5)、当該フィルタパラメータをループフィルタ 2 0 3 に供給する。

【 0 1 4 8 】

また、ループフィルタ 2 0 3 の動作が異なる。本実施の形態では、ループフィルタはピクチャ内の全ての C T B の復号処理がなされた後に実行されるため、ループフィルタ 2 0 3 はピクチャ内の全ての C T B の復号画像データを保持している。

【 0 1 4 9 】

また、ステップ S 5 0 4 のエントロピー復号部の設定の動作が異なる。本実施の形態では、復号対象 C T B がスライスまたはタイルの最初の C T B であっても、フィルタパラメータ復号部 2 0 4 はフィルタパラメータ復号部 2 0 4 の内部のエントロピー復号部を初期化しないことが第 1 の実施の形態とは異なる。

【 0 1 5 0 】

以上のように、本実施の形態に係る動画像復号装置 2 0 0 は、本実施の形態に係る動画像符号化装置 1 0 0 によって生成された符号化ストリームを復号し、復号画像データを出力することができる。

【 0 1 5 1 】

[第 3 の実施の形態]

次に、第 3 の実施の形態について説明する。本実施の形態は第 1 の実施の形態とはループフィルタを適用する画素が異なる。第 1 の実施の形態では C T B で完結するようにループフィルタを動作させたが、本実施の形態では C T B を跨ってループフィルタを実行させることを可能とする。

以下、第 1 の実施の形態とは異なる点について説明する。

【 0 1 5 2 】

最初に、P P S のシンタックスが異なる。図 2 4 は、第 3 の実施の形態に係る P P S のシンタックスを説明する図である。P P S に `loopfilter__across__tiles__flag` が追加されることが第 1 の実施の形態とは異なる。`loopfilter__across__tiles__flag` はループフィルタをタイル境界に跨って処理することを許可するか否かを示す 1 ビットのフラグである。`loopfilter__across__tiles__flag` が 1 であれば、ループフィルタをタイル境界に跨って処理することを許可し、`loopfilter__across__tiles__flag` が 0 であれば、ループフィルタをタイル境界に跨って処理することを許可しない。本実施の形態では、`loopfilter__across__tiles__flag` は 1 であるとする。

【 0 1 5 3 】

図 2 5 は、第 3 の実施の形態に係る動画像符号化装置 1 0 0 の動作を説明するフローチャートである。図 2 5 を用いて第 1 の実施の形態とは異なる本実施の形態に係る動画像符号化装置 1 0 0 の動作を説明する。ステップ S 1 0 2 に引き続いて、ピクチャの全ての C T B のフィルタパラメータが決定される (S 1 1 2)。ここで、ピクチャの全ての C T B のフィルタパラメータが決定される方法については、例えば、一般的な 2 パスのエンコードや前のピクチャのデータを参照にして生成するなどの手法で実現できる。

【 0 1 5 4 】

また、ステップ S 1 0 7 の代わりにステップ S 1 1 5 が実施される。ステップ S 1 1 5 では、ステップ S 1 1 2 で決定されたフィルタパラメータについて、符号化対象の C T B のフィルタパラメータが取得される。

【 0 1 5 5 】

また、ステップ S 1 1 1 に引き続いて、タイル境界を跨る画素についてループフィルタを実行する (S 1 1 6)。タイル境界を跨る画素とは、垂直方向のタイル境界であれば、タイル境界の左に位置する画素とタイル境界の右に位置する画素であり、水平方向のタイ

10

20

30

40

50

ル境界であれば、タイル境界の上に位置する画素とタイル境界の下に位置する画素である。ステップS 1 1 6は、ステップS 1 0 3からステップS 1 1 1をタイル単位で独立処理（並列処理）させる場合に必要な処理である。

【0156】

以上のように、ピクチャ内の全てのCTBのフィルタパラメータをCTB単位の符号化処理の前に決定し、ピクチャ内の全てのCTBについてCTB単位の符号化処理が終了した後にループフィルタを実行することで、符号化ストリームの出力は、CTB単位の符号化処理で出力することができるため、低遅延の動画像符号化装置を実現することができる。

【0157】

図26は、第3の実施の形態に係る動画像復号装置200の動作を説明するフローチャートである。図26を用いて第1の実施の形態とは異なる本実施の形態に係る動画像復号装置200の動作を説明する。ステップS 5 0 6の次にステップS 5 0 7が実施されない。ステップS 5 0 8に引き続いて、ピクチャ内の全てのCTBについて、CTBアドレスの順序でステップS 5 1 2とステップS 5 0 7の処理が繰り返し行われる（S 5 1 0からS 5 1 1）。以下、ステップS 5 1 2とステップS 5 0 7について説明する。

【0158】

処理対象CTBの`loopfilter_merge_left_flag`が1であれば、処理対象CTBの左に位置するCTBからフィルタパラメータを取得する。処理対象CTBの`loopfilter_merge_left_flag`が0で、処理対象CTBの`loopfilter_merge_above_flag`が1であれば、処理対象CTBの上に位置するCTBからフィルタパラメータを取得する。処理対象CTBの`loopfilter_merge_left_flag`が0で、処理対象CTBの`loopfilter_merge_above_flag`が0であれば、処理対象CTBのファイルパラメータを取得する（S 5 1 2）。

【0159】

ループフィルタ104は、処理対象CTBの復号画像データに対して、処理対象CTBのフィルタパラメータに基づいてループフィルタを実行し、新たな復号画像データを生成して、当該復号画像データをフレームメモリ205に供給する（S 5 0 7）。ここでは、ループフィルタ104はピクチャ内のすべてのCTBの復号画像データを保持していると
するが、CTB復号部202より供給される復号画像データをフレームメモリ205に供給して記憶させ、ステップS 5 0 7ではフレームメモリ205から処理対象CTBの復号画像データを取得するようにしてもよい。

【0160】

続いて、エッジタイプフィルタ131の動作について説明する。図27は第3の実施の形態のエッジタイプフィルタ131の動作を説明するフローチャートである。ステップS 1 4 0が第1の実施の形態とは異なる。ステップS 1 4 0では、`loopfilter_across_tiles_flag`が1であれば、ステップS 1 3 4の画素Aまたは画素Bがタイル外であるかの検査結果をスキップして、エッジタイプフィルタを実行するステップであるステップS 1 3 5以降の動作を行う。

【0161】

このように、`loopfilter_interleave_flag`が1としてフィルタパラメータの伝送遅延を小さくする場合でも、`loopfilter_across_tiles_flag`が1であれば、エッジタイプフィルタにおいてタイル外の画素Aと画素Bを参照することで、タイル境界の左右または上下に接する2画素にループフィルタを適用することが可能となつて、タイル境界の歪を低減することができる。また動き補償予測などで利用される参照ピクチャの画質が改善されるため、符号化効率を向上させることができる。

【0162】

続いて、フィルタパラメータ符号化部105とフィルタパラメータ復号部204の動作

10

20

30

40

50

における隣接CTBの有効性である`available_left_ctb_flag`と`available_above_ctb_flag`の導出について説明する。図28は第3の実施の形態における隣接CTBの有効性の導出を説明するフローチャートである。ステップS190とステップS191が追加されている点が第1の実施の形態とは異なる。ステップS190とステップS191は`loopfilter_across_tiles_flag`が1であるか検査する。以下、ステップS190とステップS191について説明する。

【0163】

`loopfilter_across_tiles_flag`が1であれば(S190のY)、ステップ182、ステップ183、及びステップ184がスキップされて、符号化対象CTBの左辺がタイル境界に接しているか否かに関わらず`available_left_ctb_flag`が1に設定される。`loopfilter_across_tiles_flag`が1でなければ(S190のN)、ステップS182が実行される。

10

`loopfilter_across_tiles_flag`が1であれば(S190のY)、ステップ186、ステップ187、及びステップ188がスキップされて、符号化対象CTBの上辺がタイル境界に接しているか否かに関わらず`available_above_ctb_flag`が1に設定される。`loopfilter_across_tiles_flag`が1でなければ(S191のN)、ステップS186が実行される。

【0164】

20

以上のように、`loopfilter_across_tiles_flag`が1に設定される場合、フィルタパラメータの符号化では、タイル境界を跨いでフィルタパラメータを取得することを示すフラグである`loopfilter_merge_left_flag`と`loopfilter_merge_above_flag`の符号化を許可し、タイル境界を跨いでフィルタパラメータが同一となるような場合、`loopfilter_merge_left_flag`や`loopfilter_merge_above_flag`を1に設定してフィルタパラメータに要する符号量を1ビットにすることで、符号化効率を向上させることができる。

【0165】

また、`loopfilter_across_tiles_flag`が1に設定される場合、ファイルユニットの復号では、`loopfilter_merge_left_flag`、`loopfilter_merge_above_flag`、及び`loopfilter_coded_param()`を復号しておき、全てのCTBの復号処理が終了した後で、ループフィルタを実行する前に`loopfilter_merge_left_flag`が1であるCTBについては当該CTBの左に位置するCTBからフィルタパラメータを取得し、`loopfilter_merge_above_flag`が1であるCTBについては当該CTBの上に位置するCTBからフィルタパラメータを取得することで、フィルタパラメータの復号処理を停止させることなく、ピクチャの全てのCTBの復号処理が終了した後にループフィルタを正確に実行させることができる。

30

【0166】

40

ここで、ファイルユニットのシンタックス要素において最も複雑な構造を有する`loopfilter_coded_param()`を事前に復号しておくことで、フィルタパラメータを取得する遅延を最小限に抑制することができる。

【0167】

また、ループフィルタの実行遅延を抑制する場合には、ステップS505において、タイル境界の外のフィルタパラメータを利用しないCTBについてはループフィルタを実行しておき、全てのCTBの復号処理が終了した後で、タイル境界の外のフィルタパラメータを利用するCTBについてループフィルタを実行するようにしてもよい。

【0168】

[第4の実施の形態]

50

次に、第4の実施の形態について説明する。第4の実施の形態は第3の実施の形態とはタイル境界におけるフィルタリングとフィルタパラメータの符号化の制御が異なる。

以下、第3の実施の形態とは異なる点について説明する。

【0169】

最初に、PPSのシンタックスが異なる。図29は、第4の実施の形態に係るPPSのシンタックスを説明する図である。PPSに`loopfilter__interleave__across__tiles__flag`が追加されることが第3の実施の形態とは異なる。`loopfilter__interleave__across__tiles__flag`はタイル境界を跨ってフィルタパラメータを取得することを許可するか否かを示す1ビットのフラグである。`loopfilter__interleave__across__tiles__flag`が1であれば、タイル境界を跨ってフィルタパラメータを取得することを許可し、`loopfilter__interleave__across__tiles__flag`が0であれば、タイル境界を跨ってフィルタパラメータを取得することを許可しない。本実施の形態では、`loopfilter__interleave__across__tiles__flag`は0であるとする。

10

【0170】

続いて、フィルタパラメータ符号化部105とフィルタパラメータ復号部204の動作における隣接CTBの有効性である`available__left__ctb__flag`と`available__above__ctb__flag`の導出について説明する。図30は第4の実施の形態における隣接CTBの有効性の導出を説明するフローチャートである。ステップS190とステップS191が第3の実施の形態とは異なる。ステップS190とステップS191は`loopfilter__interleave__across__tiles__flag`が1であるか検査する。

20

【0171】

続いて、動画像復号装置200の動作を説明する。図37は、第4の実施の形態に係る動画像復号装置200の動作を説明するフローチャートである。図37を用いて第3の実施の形態とは異なる本実施の形態に係る動画像復号装置200の動作を説明する。

ステップS506に続いて、タイル境界に接する画素以外の画素についてループフィルタが実行される(S520)。また、ステップS508に続いて、タイル境界に接する画素についてループフィルタが実行される(S521)。

30

【0172】

以上のように、`loopfilter__interleave__flag`を1、`loopfilter__across__tiles__flag`を1、`loopfilter__interleave__across__tiles__flag`を0に設定することで、全てのCTBの復号処理が終了した後のループフィルタに係る処理を、タイル境界に接する画素だけにすることで、タイル境界の歪を低減しながらループフィルタの実行遅延を最小限にすることができる。また、ピクチャ内のすべてのCTBの復号処理が完了した後にループフィルタを実行させるために記憶する必要のある復号画素を、タイル境界に接する画素に対する隣接画素Aと隣接画素Bだけにすることができ、復号画像データの記憶容量を削減することができる。また、`loopfilter__interleave__flag`を1、`loopfilter__across__tiles__flag`を1、`loopfilter__interleave__across__tiles__flag`を1に設定することで、第3の実施の形態と同じ効果を得ることができる。

40

【0173】

[第5の実施の形態]

次に、第5の実施の形態について説明する。第5の実施の形態は第1の実施の形態とは利用可能なフィルタタイプが異なる。第1の実施の形態ではタイル境界に位置するCTBとそれ以外のCTBにおいて利用可能なフィルタタイプは同一であったが、本実施の形態ではタイル境界に位置するCTBとそれ以外のCTBにおいて利用可能なフィルタタイプが異なる。

50

以下、第1の実施の形態とは異なる点について説明する。

【0174】

図31は第5の実施の形態におけるCTBの位置に応じて利用可能なフィルタタイプを説明するための図である。図31を用いてCTBの位置に応じて利用可能なフィルタタイプについて説明する。available__left__ctb__flagとavailable__above__ctb__flagが共に0で、CTBの左辺と上辺が共にピクチャ境界とタイル境界に接しないCTBについては、第1の実施の形態と同様にフィルタタイプ0からフィルタタイプ5まで全てのフィルタタイプの利用を許可して、フィルタタイプの昇順にloopfilter__type__idxを0から割り当てる。CTBの左辺がピクチャ境界またはタイル境界に接するCTBについては、フィルタタイプ1とフィルタタイプ2だけを利用を許可し、フィルタタイプ1とフィルタタイプ2にloopfilter__type__idxとしてそれぞれ0と1を割り当てる。CTBの上辺がピクチャ境界またはタイル境界に接するCTBについては、フィルタタイプ1とフィルタタイプ3だけを利用を許可し、フィルタタイプ1とフィルタタイプ3にloopfilter__type__idxとしてそれぞれ0と1を割り当てる。

10

【0175】

次に、エッジタイプフィルタ131の動作について説明する。図32は第5の実施の形態におけるエッジタイプフィルタ131の動作を説明するフローチャートである。画素Aまたは画素Bがピクチャ境界であるか検査するステップS133、画素Aまたは画素Bがタイル境界であるか検査するステップS134、オフセット値を0に設定するステップS137が省略されている点が異なる。

20

【0176】

以上のように、loopfilter__interleave__flagが1である場合、CTBの左辺がピクチャ境界またはタイル境界に接するCTBについては、ピクチャ境界またはタイル境界の外の画素を参照するフィルタタイプであるフィルタタイプ2、フィルタタイプ4、及びフィルタタイプ5を無効とすることで、ピクチャ境界またはタイル境界の外の画素を参照する必要がなくなるため、エッジタイプフィルタ131のステップS133、ステップS134、及びステップS137が不要となつて、エッジタイプフィルタ131の処理量を削減することができる。同様に、CTBの上辺がピクチャ境界またはタイル境界に接するCTBについては、ピクチャ境界またはタイル境界の外の画素を参照するフィルタタイプであるフィルタタイプ1、フィルタタイプ4、及びフィルタタイプ5を無効とすることで、ピクチャ境界またはタイル境界の外の画素を参照する必要がなくなるため、エッジタイプフィルタ131のステップS133、ステップS134、及びステップS137が不要となつて、エッジタイプフィルタ131の処理量を削減することができる。

30

【0177】

また、有効なフィルタタイプのみloopfilter__type__idxを割り当てることで、CTBの左辺または上辺がピクチャ境界またはタイル境界に接するCTBについては、loopfilter__type__idxの符号量を1ビットのフラグに削減することで、符号化効率を向上させることができる。ここで、フィルタタイプ0、つまりループフィルタを処理しないフィルタタイプについては、フィルタタイプ1またはフィルタタイプ2でオフセットを0として設定することで対応できる。

40

【0178】

さらに、第3の実施の形態や第4の実施の形態における動画像符号化装置において、本実施の形態のようにタイル境界におけるフィルタタイプに制限を設けることで、常にCTBで完結した低遅延の動画像符号化装置と動画像復号装置を実現することが可能となる。

【0179】

[第6の実施の形態]

次に、第6の実施の形態について説明する。第6の実施の形態は第4の実施の形態とはloopfilter__type__idxの符号列とavailable__left__c

50

`tb_flag`と`available_above_ctb_flag`の導出が異なる。

以下、第4の実施の形態とは異なる点について説明する。

【0180】

最初に、`loopfilter_type_idx`の符号列が異なる。図33は、第6の実施の形態に係る`loopfilter_type_idx`の符号列を説明する図である。`loopfilter_type_idx`の0はフィルタタイプ0を示し、符号列(`bin`)は'0'である。`loopfilter_type_idx`の1はフィルタタイプ1を示し、符号列(`bin`)は'10'となる。`loopfilter_type_idx`の2はフィルタタイプ2を示し、符号列(`bin`)は'1100'となる。`loopfilter_type_idx`の3はフィルタタイプ3を示し、符号列(`bin`)は'1101'となる。`loopfilter_type_idx`の4はフィルタタイプ4を示し、符号列(`bin`)は'1110'となる。`loopfilter_type_idx`の5はフィルタタイプ5を示し、符号列(`bin`)は'1111'となる。

【0181】

ここで、バンドタイプフィルタのフィルタパラメータはエッジタイプフィルタのフィルタパラメータよりも1個あたりの符号量が大きいため、バンドタイプフィルタの`loopfilter_type_idx`にエッジタイプフィルタ`loopfilter_type_idx`よりも短い符号量を割り当てることで、エッジタイプフィルタとバンドタイプフィルタのそれぞれに要する符号量を均衡させることができる。

【0182】

エッジタイプフィルタとバンドタイプフィルタのそれぞれに要する符号量を均衡させることで、エッジタイプフィルタとバンドタイプフィルタのそれぞれのループフィルタ符号化パラメータの符号化時間と復号時間を均衡させることができ、ハードウェアの回路設計やソフトウェアの設計を容易にすることができる。ここでは、バンドタイプフィルタの`loopfilter_type_idx`を2ビット、エッジタイプフィルタ`loopfilter_type_idx`を4ビットに設定したが、バンドタイプフィルタの`loopfilter_type_idx`がエッジタイプフィルタ`loopfilter_type_idx`よりも短い符号量であれば、これに限定されない。

【0183】

次に、`available_left_ctb_flag`と`available_above_ctb_flag`の導出が異なる。続いて、フィルタパラメータ符号化部105とフィルタパラメータ復号部204の動作における隣接CTBの有効性である`available_left_ctb_flag`と`available_above_ctb_flag`の導出について説明する。図34は第6の実施の形態における隣接CTBの有効性の導出を説明するフローチャートである。ステップS196とステップS197が追加されている点が第4の実施の形態とは異なる。以下、ステップS196とステップS197について説明する。

【0184】

符号化対象CTBの左に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0(フィルタOFF)であるか検査する(S196)。符号化対象CTBの左に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0であれば(S196のY)、`available_left_ctb_flag`を0とする(S184)。符号化対象CTBの左に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0でなければ(S196のN)、S182を実行する。

【0185】

符号化対象CTBの上に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0(フィルタOFF)であるか検査する(S197)。符号化対象CTBの上に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0であれば(S197のY)、`available_above_ctb_flag`を0とする(S188)。符号化対象CTBの上に位置するCTBのフィルタパラメータのフィルタタイプ

がフィルタタイプ0でなければ(S197のN)、S186を実行する。

【0186】

ここで、フィルタタイプがフィルタタイプ0である場合、`loopfilter_type_idx`は1ビットであるため、`loopfilter_merge_left_flag`や`loopfilter_merge_above_flag`を用いて符号化するよりも少ない符号量で済む可能性が高い。これは、左に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0であれば、`loopfilter_merge_left_flag`の1ビットで済むが、左に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0でなく、上に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0であれば、`loopfilter_merge_left_flag`と`loopfilter_merge_above_flag`の2ビットとなり、左に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0でなく、上に位置するCTBのフィルタパラメータのフィルタタイプがフィルタタイプ0でなければ、`loopfilter_merge_left_flag`、`loopfilter_merge_above_flag`、`loopfilter_type_idx`の3ビットとなるためである。

10

【0187】

したがって、フィルタタイプがフィルタタイプ0である場合、`loopfilter_merge_left_flag`と`loopfilter_merge_above_flag`を符号化せず、ループフィルタ符号化パラメータだけを符号化するようにして符号化効率を向上させることができる。

20

【0188】

[第7の実施の形態]

次に、第7の実施の形態について説明する。第7の実施の形態は第6の実施の形態とはループフィルタユニットのシンタックスと`loopfilter_type_idx`の符号列が異なる。

以下、第6の実施の形態とは異なる点について説明する。

【0189】

最初に、ループフィルタユニットのシンタックスが異なる。図35は第7の実施の形態に係るループフィルタユニットのシンタックスを説明する図である。ループフィルタ許可フラグである`enable_loopfilter_flag`が追加されている。`enable_loopfilter_flag`はCTBのループフィルタのONとOFFを制御するフラグであって、`enable_loopfilter_flag`が1であるCTBについてはループフィルタを実行し、`enable_loopfilter_flag`が0であるCTBについてはループフィルタを実行しない。

30

【0190】

次に、`loopfilter_type_idx`の符号列が異なる。図36は、第7の実施の形態に係る`loopfilter_type_idx`の符号列を説明する図である。`loopfilter_type_idx`の0はフィルタタイプ1を示し、符号列(bin)は'0'である。`loopfilter_type_idx`の1はフィルタタイプ2を示し、符号列(bin)は'100'となる。`loopfilter_type_idx`の2はフィルタタイプ3を示し、符号列(bin)は'101'となる。`loopfilter_type_idx`の3はフィルタタイプ4を示し、符号列(bin)は'110'となる。`loopfilter_type_idx`の4はフィルタタイプ5を示し、符号列(bin)は'111'となる。

40

【0191】

次に、フィルタパラメータ符号化部105とフィルタパラメータ復号部204の動作における隣接CTBの有効性である`available_left_ctb_flag`と`available_above_ctb_flag`の導出が異なる。第6の実施の形態とは、ステップS196とステップS197が異なり、ステップS196とステップS19

50

7について説明する。

【0192】

符号化対象CTBの左に位置するCTBのフィルタパラメータのループフィルタ許可フラグが0であるか検査する(S196)。符号化対象CTBの左に位置するCTBのフィルタパラメータのループフィルタ許可フラグが0であれば(S196のY)、available_left_ctb_flagを0とする(S184)。符号化対象CTBの左に位置するCTBのフィルタパラメータのループフィルタ許可フラグが0でなければ(S196のN)、S182を実行する。

【0193】

符号化対象CTBの上に位置するCTBのフィルタパラメータのループフィルタ許可フラグが0であるか検査する(S197)。符号化対象CTBの上に位置するCTBのフィルタパラメータのループフィルタ許可フラグが0であれば(S197のY)、available_above_ctb_flagを0とする(S188)。符号化対象CTBの上に位置するCTBのフィルタパラメータのループフィルタ許可フラグが0でなければ(S197のN)、S186を実行する。

【0194】

以上述べた実施の形態の動画像符号化装置と動画像復号装置では、領域の境界としてタイル境界を説明したが、タイル内のCTBはラスタスキャン順に処理されてスライス
はタイル内のCTBをラスタスキャン順に包含するようにタイルとスライスを構成した場合に、タイルとスライスを組み合わせることで、スライス境界についてもタイル境界をス
ライス境界に置き換えることで同様に適用することができる。

【0195】

以上述べた実施の形態の動画像符号化装置と動画像復号装置では、ループフィルタとしてSAOフィルタを用いたが、例えば、デブロックフィルタとSAOフィルタを組み合わせることもできる。

【0196】

以上述べた実施の形態の動画像符号化装置が出力する動画像の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おり、動画像符号化装置に対応する動画像復号装置が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。

【0197】

動画像符号化装置と動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、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に変換して伝送してもよい。その場合、動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動画像送信装置と、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動画像復号装置に供給する動画像受信装置とが設けられる。

【0198】

動画像送信装置は、動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、パケット化された符号化データをネットワークを介して送信する送信部とを含む。動画像受信装置は、パケット化された符号化データをネットワークを介して受信する受信部と、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、動画像復号装置に提供するパケット処理部とを含む。

【0199】

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することができるのは勿論のこと、ROM(リード・オンリ・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も実現することができる。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することも、有線あるいは無線

10

20

30

40

50

のネットワークを通してサーバから提供すること、地上波あるいは衛星デジタル放送のデータ放送として提供することも可能である。

【 0 2 0 0 】

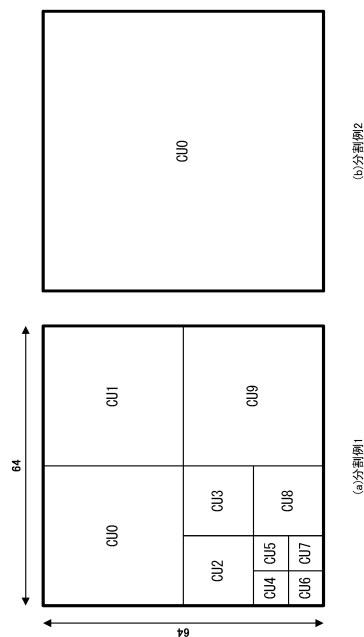
以上、本発明を実施の形態をもとに説明した。実施の形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

【符号の説明】

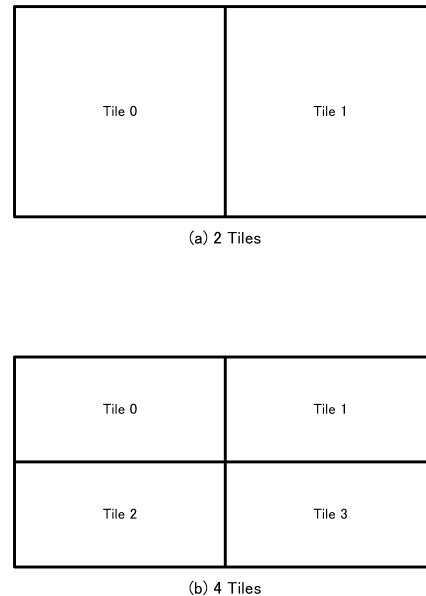
【 0 2 0 1 】

1 0 0 動画像符号化装置、 1 0 1 画像データ取得部、 1 0 2 C T B 符号化部、 1 0 3 フィルタパラメータ決定部、 1 0 4 ループフィルタ、 1 0 5 フィルタパラメータ符号化部、 1 0 6 フレームメモリ、 1 0 7 符号列多重部、 1 1 0 符号化制御部、 1 1 1 タイル設定部、 1 1 2 ループフィルタ設定部、 1 1 3 領域情報設定部、 1 2 0 フィルタパラメータ設定部、 1 2 1 ループフィルタ実行部、 1 2 2 誤差計測部、 1 2 3 フィルタパラメータ確定部、 1 3 0 フィルタタイプ判定部、 1 3 1 エッジタイプフィルタ、 1 3 2 バンドタイプフィルタ、 1 4 1 エッジアングル設定部、 1 4 2 隣接画素設定部、 1 4 3 エッジタイプ判定部、 1 4 4 エッジオフセット値設定部、 1 4 5 エッジオフセット値加算部、 1 5 1 ハンド幅設定部、 1 5 2 バンド位置設定部、 1 5 3 バンドオフセット値設定部、 1 5 4 バンドオフセット値加算部、 2 0 0 動画像復号装置、 2 0 1 符号列分離部、 2 0 2 C T B 復号部、 2 0 3 ループフィルタ、 2 0 4 フィルタパラメータ復号部、 2 0 5 フレームメモリ、 2 1 0 復号制御部。

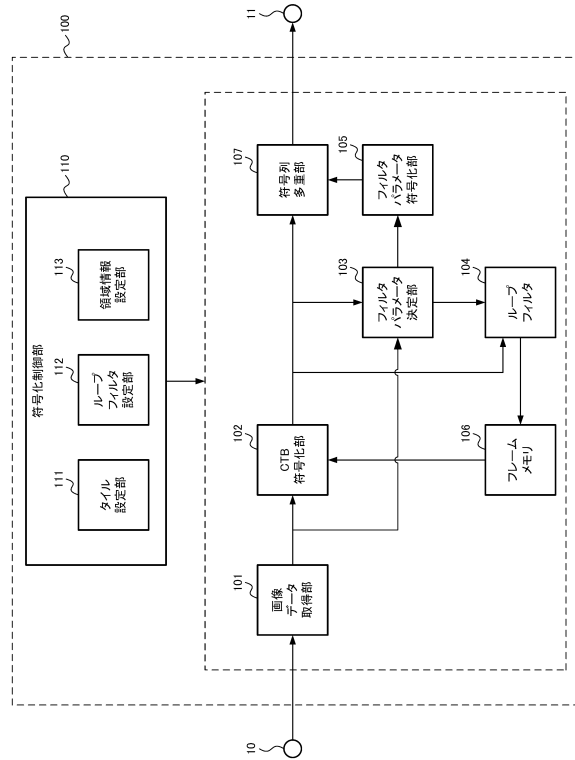
【 図 1 】



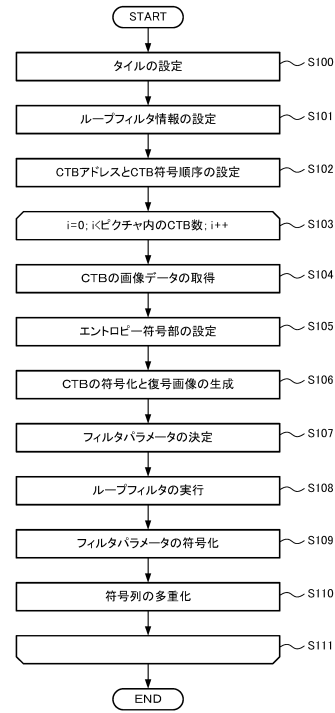
【 図 2 】



【図 3】



【図 4】



【図 5】

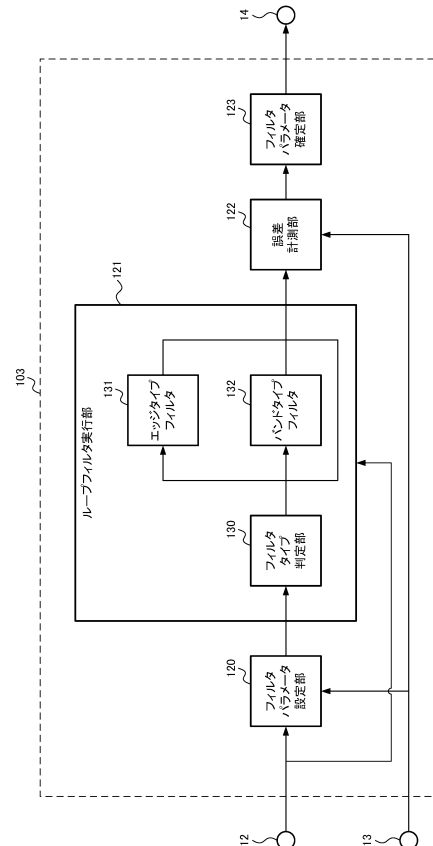
0	1	2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31	32	33	34	35
36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71

(a) Address Order

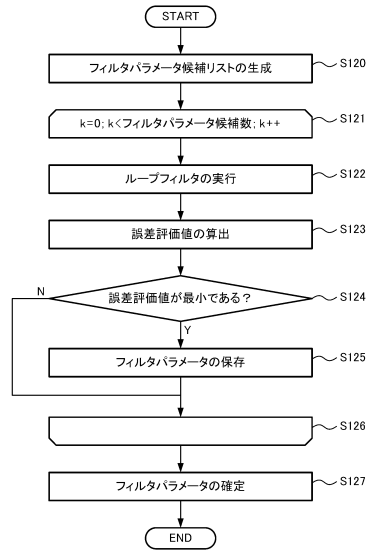
0	1	2	3	4	5	18	19	20	21	22	23
6	7	8	9	10	11	24	25	26	27	28	29
12	13	14	15	16	17	30	31	32	33	34	35
36	37	38	39	40	41	54	55	56	57	58	59
42	43	44	45	46	47	60	61	62	63	64	65
48	49	50	51	52	53	66	67	68	69	70	71

(b) Coding Order

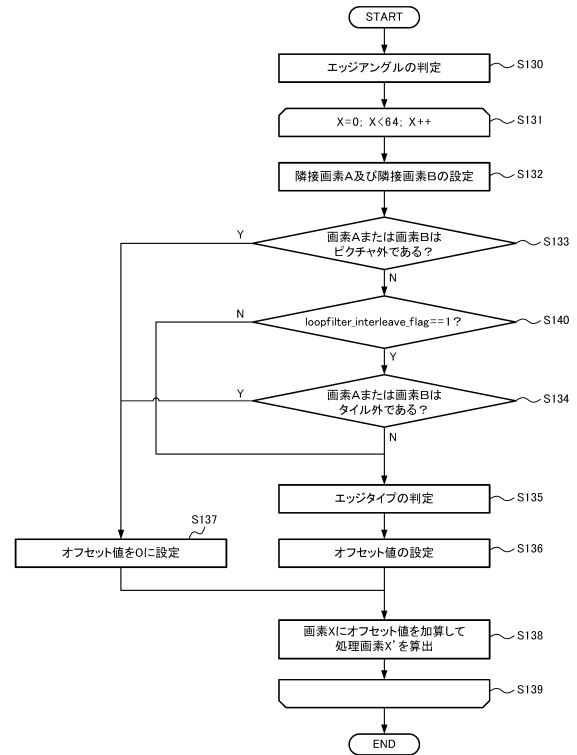
【図 6】



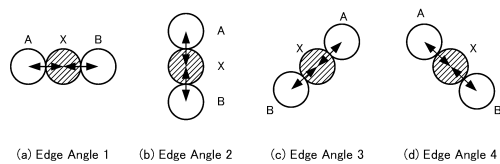
【図 7】



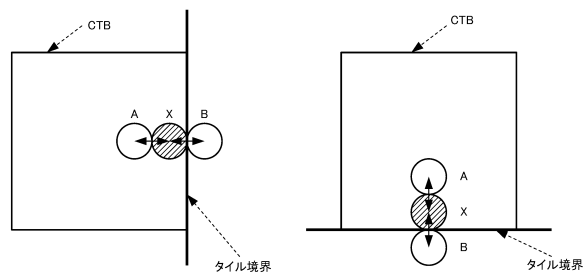
【図 8】



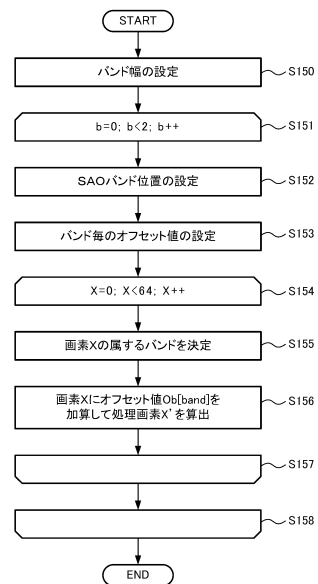
【図 9】



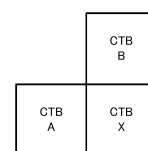
【図 10】



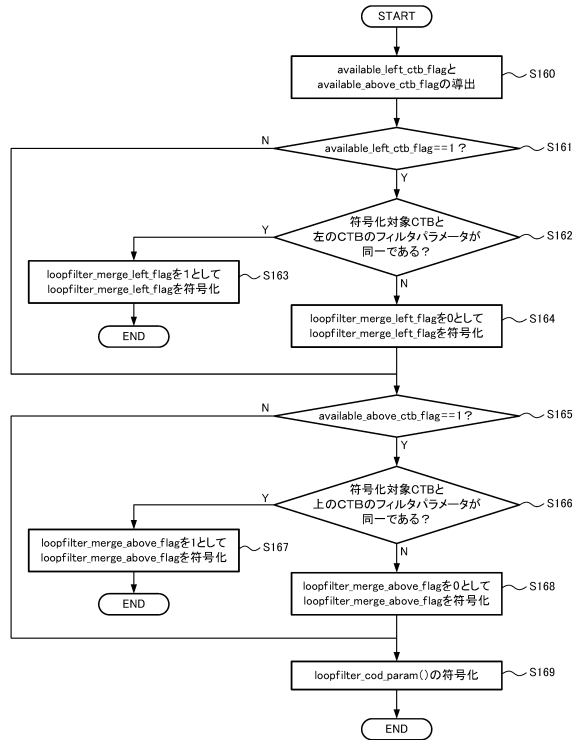
【図 11】



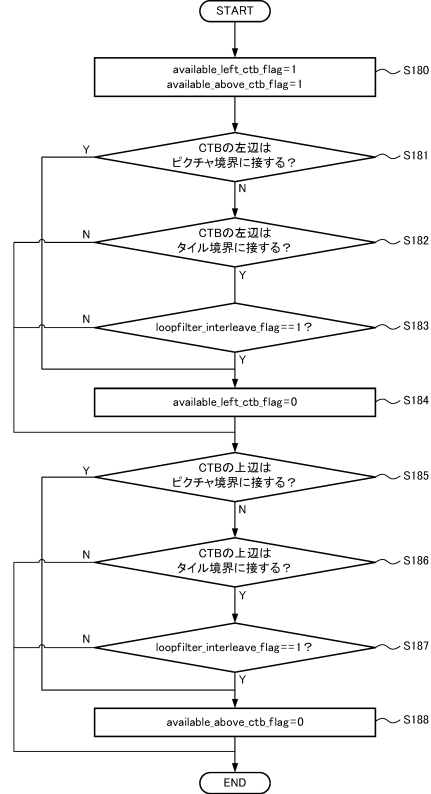
【図 12】



【図 13】



【図 14】



【図 15】

(a) PPS

```

PictureParameterSet()
{
    ...
    loopfilter_interleave_flag
    if( !loopfilter_interleave_flag ) {
        pic_loopfilter_param()
    }
    ...
    tiles_coding_flag
    if( tiles_coding_flag ) {
        num_tile_columns_minus1
        num_tile_rows_minus1
        uniform_spacing_flag
        if( !uniform_spacing_flag ) {
            for( i = 0; i < num_tile_columns_minus1; i++ )
                column_width[ i ]
            for( i = 0; i < num_tile_rows_minus1; i++ )
                column_height[ i ]
        }
    }
}
  
```

(b) slice header

```

slice_header()
{
    slice_address
    slice_type
    ...
    if( tiles_coding_flag > 0 ) {
        num_of_tiles
        for( i = 1; i <= num_of_tiles; i++ )
            tile_position[ i ]
    }
}
  
```

【図 16】

(a) picture loopfilter parameter

```

pic_loopfilter_param() {
    ...
    for( i = 0; i < num_of_ctbs; i++ )
        loopfilter_unit()
    }
}
  
```

(b) loopfilter coding parameter

```

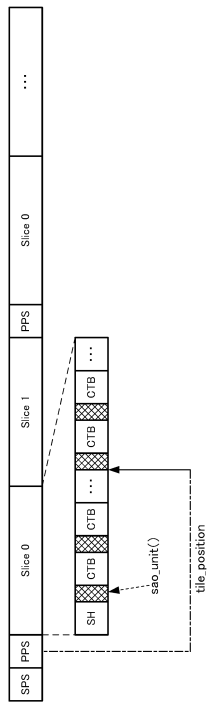
loopfilter_cod_param() {
    loopfilter_type_idx
    if( loopfilter_type_idx == 5 ) {
        loopfilter_band_position
        loopfilter_offset
    }
    if( loopfilter_type_idx == 1 )
        for( i = 0; i < 2; i++ )
            loopfilter_offset[ i ]
    else if( loopfilter_type_idx != 0 )
        for( i = 0; i < 4; i++ )
            loopfilter_offset[ i ]
}
  
```

(c) loopfilter unit

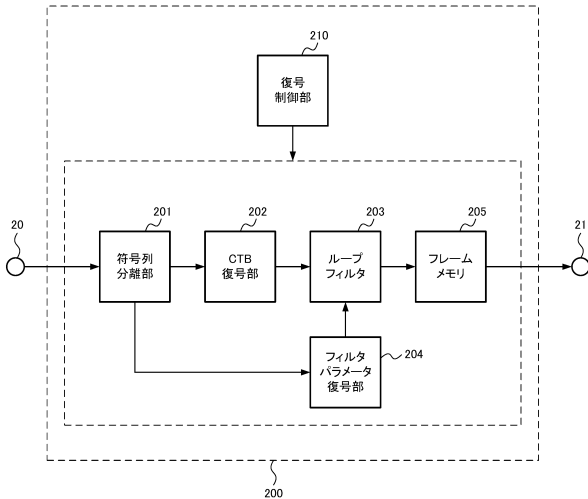
```

loopfilter_unit() {
    if( available_left_ctb_flag )
        loopfilter_merge_left_flag
    if( !loopfilter_merge_left_flag && available_above_ctb_flag )
        loopfilter_merge_above_flag
    if( !loopfilter_merge_left_flag && !loopfilter_merge_above_flag )
        loopfilter_cod_param()
}
  
```

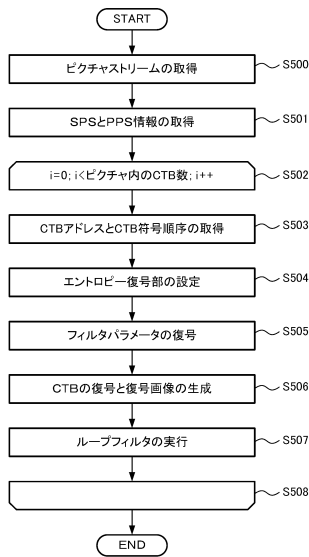
【図 17】



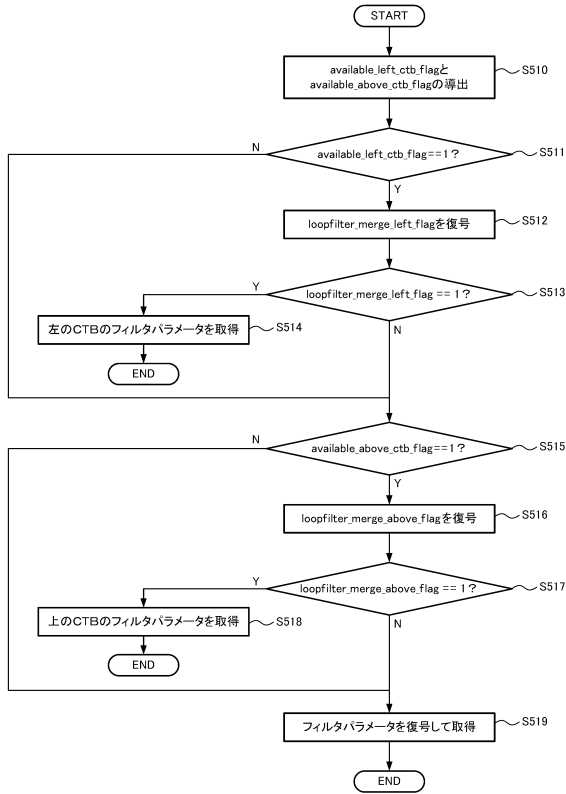
【図 18】



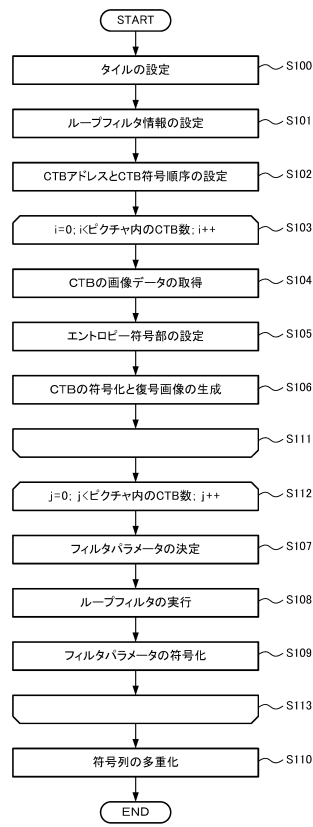
【図 19】



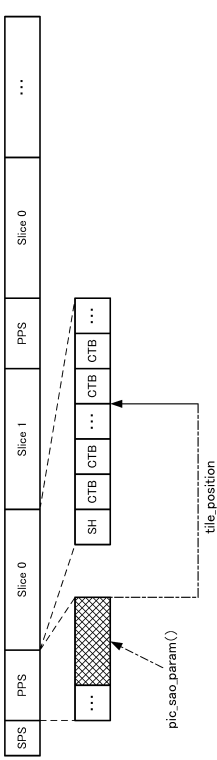
【図 20】



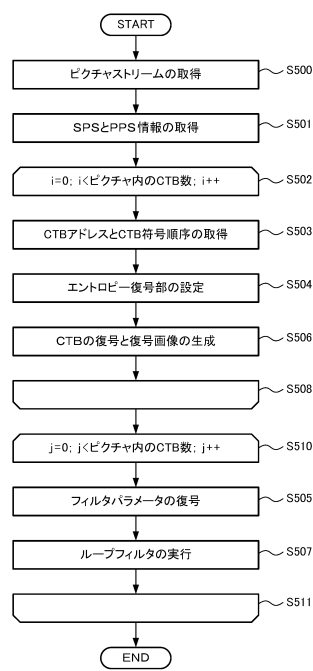
【図 2 1】



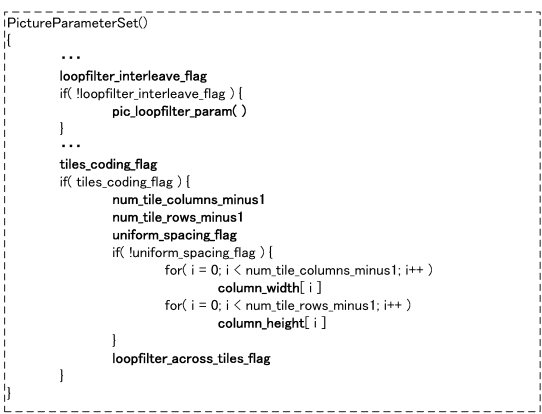
【図 2 2】



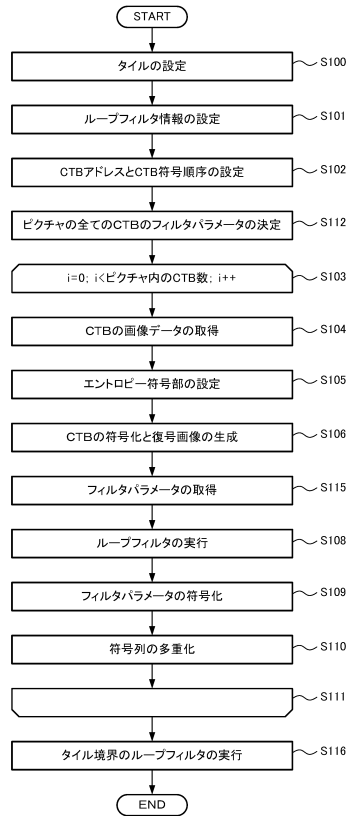
【図 2 3】



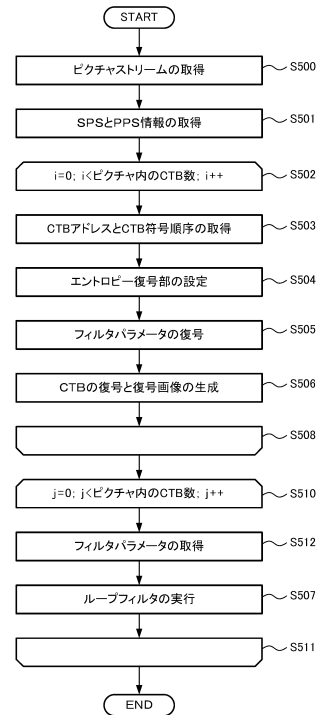
【図 2 4】



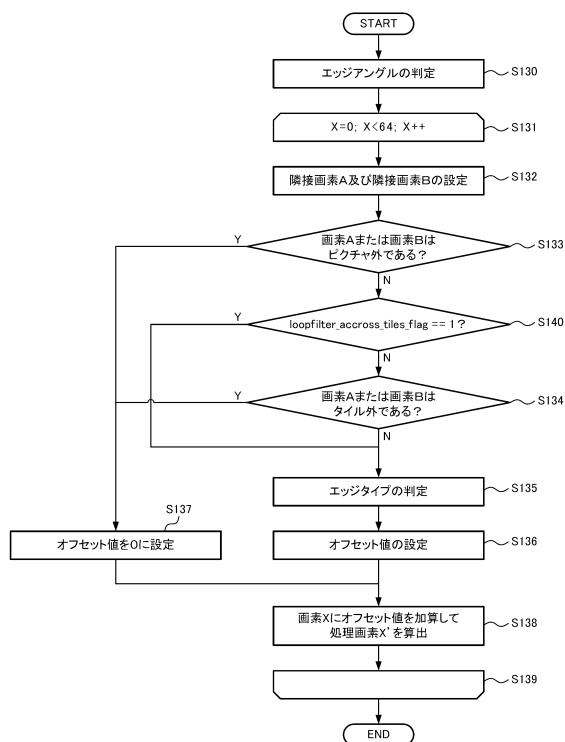
【図 25】



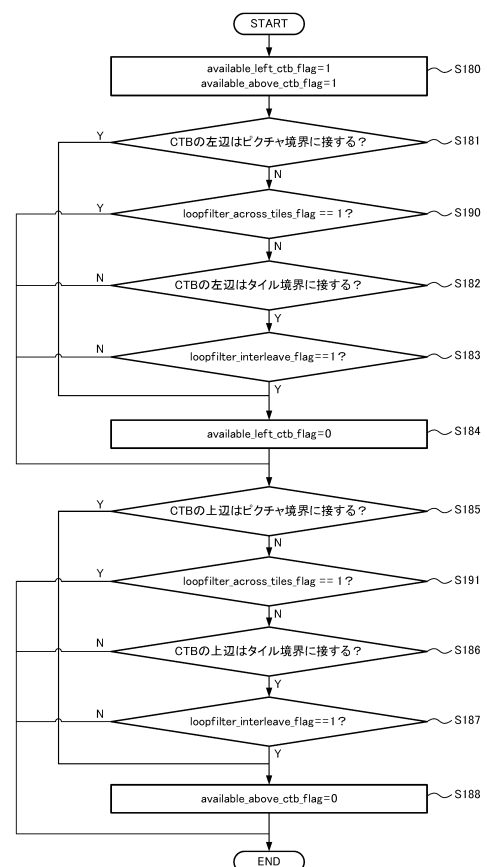
【図 26】



【図 27】



【図 28】



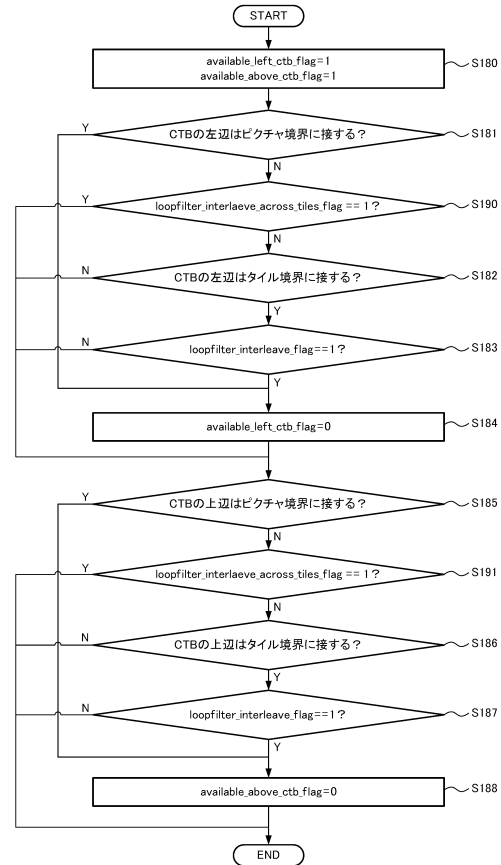
【図 29】

```

PictureParameterSet()
{
    ...
    loopfilter_interleave_flag
    if( !loopfilter_interleave_flag ) {
        pic_loopfilter_param()
    }
    ...
    tiles_coding_flag
    if( tiles_coding_flag ) {
        num_tile_columns_minus1
        num_tile_rows_minus1
        uniform_spacing_flag
        if( !uniform_spacing_flag ) {
            for( i = 0; i < num_tile_columns_minus1; i++ )
                column_width[ i ]
            for( i = 0; i < num_tile_rows_minus1; i++ )
                column_height[ i ]
        }
        loopfilter_across_tiles_flag
        if( loopfilter_interleave_flag ) {
            loopfilter_interleave_across_tiles_flag
        }
    }
}

```

【図 30】



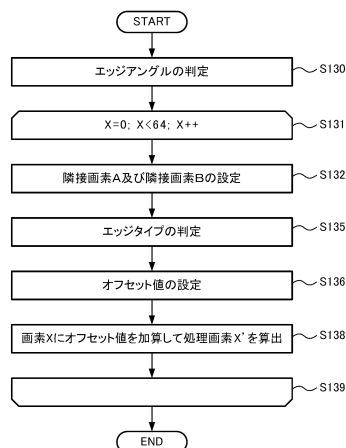
【図 31】

loopfilter_type_idx	タイル境界に接しない CTB	左辺がタイル境界に接する CTB	上辺がタイル境界に接する CTB
0	フィルタタイプ0	フィルタタイプ1	フィルタタイプ1
1	フィルタタイプ1	フィルタタイプ2	フィルタタイプ3
2	フィルタタイプ2	—	—
3	フィルタタイプ3	—	—
4	フィルタタイプ4	—	—
5	フィルタタイプ5	—	—

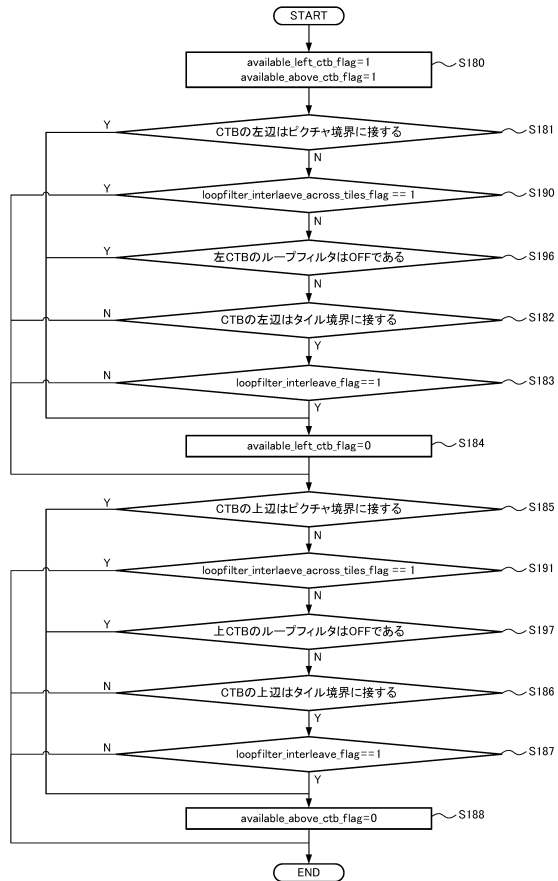
【図 33】

loopfilter_type_idx	フィルタタイプ	bin
0	フィルタタイプ0	0
1	フィルタタイプ1	10
2	フィルタタイプ2	1100
3	フィルタタイプ3	1101
4	フィルタタイプ4	1110
5	フィルタタイプ5	1111

【図 32】



【図 3 4】



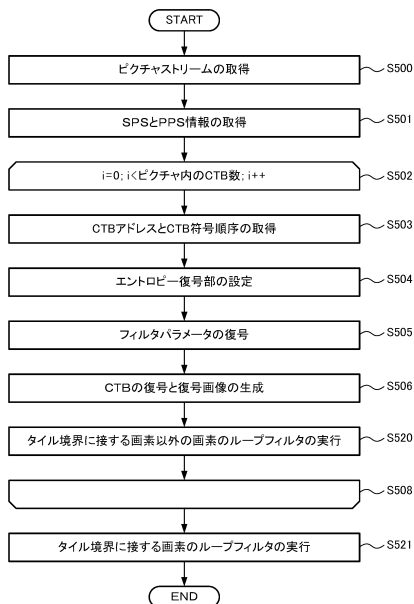
【図 3 5】

```
Loopfilter_unit( ) {
    enable_loopfilter_flag
    if( available_left_ctb_flag ) {
        if( available_left_ctb_flag )
            loopfilter_merge_left_flag
        if( !loopfilter_merge_left_flag && available_above_ctb_flag )
            loopfilter_merge_above_flag
        if( !loopfilter_merge_left_flag && !loopfilter_merge_above_flag )
            loopfilter_cod_Param( )
    }
}
```

【図 3 6】

loopfilter_type_idx	フィルタタイプ	bin
0	フィルタタイプ1	0
1	フィルタタイプ2	100
2	フィルタタイプ3	101
3	フィルタタイプ4	110
4	フィルタタイプ5	111

【図 3 7】



フロントページの続き

審査官 岩井 健二

(56)参考文献 国際公開第 2 0 1 3 / 1 1 1 5 9 3 (WO , A 1)

Chih-Ming Fu et al. , Sample Adaptive Offset with LCU-based Syntax , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 6th Meeting: Torino, IT , 2 0 1 1 年 7 月 , JCTVC-F056_r3 , pp.1-6

Chia-Yang Tsai et al. , AHG4: Non-cross-tiles loop filtering for independent tiles , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 7th Meeting: Geneva , 2 0 1 1 年 1 1 月 , JCTVC-G194_r5 , pp.1-6

Ye-Kui Wang et al. , Dependency and loop filtering control over tile boundaries , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 7th Meeting: Geneva, CH , 2 0 1 1 年 1 1 月 , JCTVC-G317 , pp.1-7

Ching-Yeh Chen et al. , Non-CE8: Sample Adaptive Offset with LCU-based Syntax , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 7th Meeting: Geneva, CH , 2 0 1 1 年 1 1 月 , JCTVC-G831-r1 , pp.1-4

Ye-Kui Wang et al. , Flexible tile dependency and loop filtering control , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 8th Meeting: San Jose, CA, USA , 2 0 1 2 年 1 月 , JCTVC-H0521r1 , pp.1-7

Benjamin Bross et al. , High efficiency video coding (HEVC) text specification draft 6 , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 7th Meeting: Geneva, CH , 2 0 1 2 年 2 月 , JCTVC-H1003_dJ , pp.i-x,1-19,27-46,70-77,83,154-156

Koohyar Minoo et al. , Non-CE1: Coding of SAO merge left and merge up , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 9th Meeting: Geneva, CH , 2 0 1 2 年 4 月 , JCTVC-I0507 , pp.1-7

Koohyar Minoo and David Baylon , Coding of SAO merge left and merge up flags , Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 , 10th Meeting: Stockholm, SE , 2 0 1 2 年 7 月 , JCTVC-J0355-r2 , pp.1-7

(58)調査した分野(Int.Cl. , D B 名)

H 0 4 N 1 9 / 0 0 - 1 9 / 9 8