

(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2010-16910

(P2010-16910A)

(43) 公開日 平成22年1月21日(2010.1.21)

(51) Int.Cl. F I テーマコード (参考)
H04N 7/32 (2006.01) H04N 7/137 Z 5C159

審査請求 有 請求項の数 6 O L (全 50 頁)

(21) 出願番号	特願2009-242688 (P2009-242688)	(71) 出願人	000005049
(22) 出願日	平成21年10月21日 (2009.10.21)		シャープ株式会社
(62) 分割の表示	特願2008-501902 (P2008-501902) の分割		大阪府大阪市阿倍野区長池町2番2号
原出願日	平成18年2月28日 (2006.2.28)	(74) 代理人	100153110
(31) 優先権主張番号	60/663,161		弁理士 岡田 宏之
(32) 優先日	平成17年3月18日 (2005.3.18)	(74) 代理人	100079843
(33) 優先権主張国	米国 (US)		弁理士 高野 明近
(31) 優先権主張番号	60/683,060	(72) 発明者	サン, シジュン
(32) 優先日	平成17年5月20日 (2005.5.20)		アメリカ合衆国 98052 ワシントン
(33) 優先権主張国	米国 (US)		州, レドモント, 12015, エヌイー
(31) 優先権主張番号	60/686,676		174番 コート
(32) 優先日	平成17年6月1日 (2005.6.1)	Fターム(参考)	5C159 KK03 MA00 MA05 MA17 MA21
(33) 優先権主張国	米国 (US)		MA32 MC11 MC38 ME01 PP16
			SS03 TA69 TB08 TC24 TC42

(54) 【発明の名称】 空間的にスケーラブルなビデオ符号化のために、低い複雑さでブロッキング歪を低減するための方法およびシステム

(57) 【要約】

【課題】画像処理によって生じるブロッキング歪を低減する。

【解決手段】本発明の実施例は、空間スケーラブルなビデオ符号化におけるフィルタ演算のためのブロック境界の特徴を定めるためのシステムおよび方法を含む。本発明の一部の実施例は、H.264/MPEG-4 AVCのスケーラブルなビデオ符号化拡張と共に使用されるようになっている方法およびシステムを含む。

【選択図】なし

【特許請求の範囲】**【請求項 1】**

レイヤ間テクスチャ予測を使って隣接するブロックのうちの少なくとも 1 つを符号化する、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロック間のブロック境界の特徴を定めるための方法であって、

前記レイヤ間テクスチャ予測以外のイントラ予測モードを使って前記隣接するブロックのうちの 1 つを符号化するとき、第 1 の境界強度インジケータにより前記ブロック境界の特徴を定めるステップと、

前記隣接するブロックのうちの 1 つをインター予測モードを使って符号化するとき、第 2 の境界強度インジケータを用いて前記ブロック境界の特徴を定めるステップとを含み、

前記ブロック境界に対してデブロッキングフィルタを適用するときに、

前記第 1 の境界強度インジケータは前記第 2 の境界強度インジケータよりも強いデブロッキングフィルタを適用させることを特徴とする方法。

【請求項 2】

レイヤ間テクスチャ予測を使って隣接するブロックのうちの少なくとも 1 つを符号化する、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロック間のブロック境界の特徴を定めるための方法であって、

前記レイヤ間テクスチャ予測以外のイントラ予測モードを使って前記隣接するブロックのうちの 1 つを符号化するとき、第 1 の境界強度インジケータにより前記ブロック境界の特徴を定めるステップと、

前記レイヤ間テクスチャ予測を使って前記隣接するブロックの双方を符号化して、

前記隣接するブロックのいずれかが非ゼロ変換係数を有するときに、

第 3 の境界強度インジケータを用いて前記ブロック境界の特徴を定めるステップを含み、

前記ブロック境界に対してデブロッキングフィルタを適用するときに、

前記第 1 の境界強度インジケータは前記第 3 の境界強度インジケータよりも強いデブロッキングフィルタを適用させることを特徴とする方法。

【請求項 3】

レイヤ間テクスチャ予測を使って隣接するブロックのうちの少なくとも 1 つを符号化する、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロック間のブロック境界の特徴を定めるための方法であって、

前記レイヤ間テクスチャ予測以外のイントラ予測モードを使って前記隣接するブロックのうちの 1 つを符号化するとき、第 1 の境界強度インジケータにより前記ブロック境界の特徴を定めるステップと、

前記レイヤ間テクスチャ予測を使ってもとの通り前記隣接するブロックの双方を符号化して、

前記隣接するブロックのいずれかが非ゼロ変換係数を有するときに、

第 3 の境界強度インジケータを用いて前記ブロック境界の特徴を定めるステップと、

前記レイヤ間テクスチャ予測を使って前記隣接するブロックの双方を符号化して、

前記隣接するブロックのいずれも非ゼロ変換係数を有しないときに、

第 4 の境界強度インジケータを用いて前記ブロック境界の特徴を定めるステップを含み、

前記ブロック境界に対してデブロッキングフィルタを適用するときに、

前記第 1 の境界強度インジケータは前記第 3 の境界強度インジケータよりも強いデブロッキングフィルタを適用させ、

前記第 3 の境界強度インジケータは前記第 4 の境界強度インジケータよりも強いデブロッキングフィルタを適用させることを特徴とする方法。

【請求項 4】

レイヤ間テクスチャ予測を使って隣接するブロックのうちの少なくとも 1 つを符号化する

る、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロック間のブロック境界の特徴を定めるシステムであって、

前記レイヤ間テクスチャ予測以外のイントラ予測モードを使って前記隣接するブロックのうちの1つを符号化するとき、第1の境界強度インジケータを用いて前記ブロック境界の特徴を定めるための予測モードアナライザを備え、

前記隣接するブロックのうちの1つをインター予測モードを使って符号化するとき、

第2の境界強度インジケータを用いて前記ブロック境界の特徴を定め、

前記ブロック境界に対してデブロッキングフィルタを適用するときに、

前記第1の境界強度インジケータは前記第2の境界強度インジケータよりも強いデブロッキングフィルタを適用させることを特徴とするシステム。

10

【請求項5】

レイヤ間テクスチャ予測を使って隣接するブロックのうちの少なくとも1つを符号化する、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロック間のブロック境界の特徴を定めるシステムであって、

前記レイヤ間テクスチャ予測以外のイントラ予測モードを使って前記隣接するブロックのうちの1つを符号化するとき、第1の境界強度インジケータを用いて前記ブロック境界の特徴を定めるための予測モードアナライザと、

前記レイヤ間テクスチャ予測を使って前記隣接するブロックの双方を符号化して、

前記隣接するブロックのいずれかが非ゼロ変換係数を有するときに、

第3の境界強度インジケータを用いて前記ブロック境界の特徴を定めるための係数アナライザを備え、

20

前記ブロック境界に対してデブロッキングフィルタを適用するときに、

前記第1の境界強度インジケータは前記第3の境界強度インジケータよりも強いデブロッキングフィルタを適用させることを特徴とするシステム。

【請求項6】

レイヤ間テクスチャ予測を使って隣接するブロックのうちの少なくとも1つを符号化する、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロック間のブロック境界の特徴を定めるシステムであって、

前記レイヤ間テクスチャ予測以外のイントラ予測モードを使って前記隣接するブロックのうちの1つを符号化するとき、第1の境界強度インジケータを用いて前記ブロック境界の特徴を定めるための予測モードアナライザと、

30

前記レイヤ間テクスチャ予測を使って前記隣接するブロックの双方を符号化して、

前記隣接するブロックのいずれかが非ゼロ変換係数を有するときに、

第3の境界強度インジケータを用いて前記ブロック境界の特徴を定めるための係数アナライザを備え、

前記レイヤ間テクスチャ予測を使って前記隣接するブロックの双方を符号化して、

前記隣接するブロックのいずれも非ゼロ変換係数を有しないときに、

第4の境界強度インジケータを用いて前記ブロック境界の特徴を定め、

前記ブロック境界に対してデブロッキングフィルタを適用するときに、

前記第1の境界強度インジケータは前記第3の境界強度インジケータよりも強いデブロッキングフィルタを適用させ、

40

前記第3の境界強度インジケータは前記第4の境界強度インジケータよりも強いデブロッキングフィルタを適用させることを特徴とするシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明の実施例は、画像ブロック境界フィルタリング制御のための方法およびシステムに関する。本発明の一部の実施例は、逆ブロッキングフィルタ演算を制御するための、空間的スケーラビリティエンハンスメントレイヤ内の隣接ブロック間のブロック境界の特徴を定めるための方法およびシステムを含む。

50

【背景技術】

【0002】

H.264/MPEG-4 AVC [ITU-T VCEGおよびISO/IEC MPEGのJoint Video Team、「高度ビデオ符号化(AVC) - 第4版」、ITU-T 勧告H.264およびISO/IEC14496-10 (MPEG-4-Part10)、2005年1月] (この規格を本明細書で参考例として援用する)は、ビデオコーデック仕様であり、この仕様は、マクロブロック予測を使用しており、マクロブロック予測の次に残差符号化が行われ、圧縮効率のためにビデオシーケンス内の時間的および空間的冗長性を低減するようになっている。空間的スケーラビリティとは、サポートされた任意の空間解像度でレート歪性能を維持しながら、ビットストリームの一部を除去できる機能を意味する。単一レイヤのH.264/MPEG-4 AVCは、空間的スケーラビリティをサポートしていない。

空間的スケーラビリティは、H.264/MPEG-4 AVCのスケーラブルなビデオ符号化(SVC)拡張によってサポートされている。

10

【0003】

H.264/MPEG-4 AVC [ジョイントスケーラブルビデオモデル(JSVM)のための作業ドキュメント1.0 (WD-1.0) (MPEGドキュメントN6901)のSVC拡張(これを本明細書で参考例として援用する)は、レイヤ状のビデオコーデックであり、このビデオコーデックではレイヤ間予測機構により空間レイヤ間の冗長性を利用している。H.264/MPEG-4 AVCのSVC拡張のデザイン内には、3つのレイヤ間予測技術、すなわちレイヤ間動き予測、レイヤ間残差予測およびレイヤ間イントラテクスチャ予測が含まれている。

【0004】

ブロックをベースとする動き補償ビデオ符号化は、多くのビデオ圧縮規格、例えばH.261、H.263、H.264、MPEG-1、MPEG-2およびMPEG-4で使用されている。損失の大きい圧縮プロセスは、復号された画像内に、画像歪と称される視覚的な歪を発生することがある。ブロッキング歪は、画像内のブロック境界に沿って生じ、変換係数の粗い量子化が原因となって生じる。

20

【0005】

再構成された画像内の歪を低減するのに、画像フィルタリング技術を使用できる。再構成された画像は、逆変換され、復号された後に生成される画像であり、これら技術における経験則は、画像の残りの部分をスムーズにする一方、画像のエッジを保存しなければならないことである。特定のピクセルまたは画像エッジを囲むピクセルの組の特性に基づき、ローパスフィルタを注意深く選択する。

30

【発明の概要】

【発明が解決しようとする課題】

【0006】

画像ブロック境界を横断するように延びる相関性のない画像ピクセルは、ブロッキング歪を低減するために特にフィルタリングされる。しかしながら、このフィルタリングは、画像内にボケた歪を生じさせる。隣接するブロック間にブロッキング歪がほとんどないか、または皆無である場合、ローパスフィルタリングは不必要に画像内にボケを生じさせ、同時に処理リソースを無駄にする。

【0007】

これまでは、SVCによりディアディック空間的スケーラビリティしか解決されていなかった。ディアディック空間的スケーラビリティとは、2つの連続する空間レイヤの間のピクチャーディメンジョンの比が2の累乗となる構成を意味する。連続する空間レイヤの間のピクチャーディメンジョンの比が2の累乗でなく、高レベルのピクチャーがより低いレベルの対応するピクチャー内に存在しない領域を含むことがある構成を管理する新しいツール(クロッピングウィンドウ(cropping window)による非ディアディックスケーリングと称される)がこれまで提案されている。

40

【0008】

レイヤ間予測方法のいずれも、ピクチャーアップサンプリングを含む。ピクチャーアップサンプリングとは、より低い解像度の画像からより高い解像度の画像を生成するプロセ

50

スのことである。一部のピクチャーアップサンプリングプロセスは、サンプル補間法を含む。SVCデザインで使用されるこれまでのアップサンプリングプロセスは、インター予測のためのH.264で指定されているルミナンスの4分の1サンプル補間方法に基づいていた。従来の方法を空間的にスケーラブルな符号化に使用するとき、この従来の方法には次の2つの欠点がある。すなわち補間解像度が4分の1のサンプルに限定されるので、非ディアドリックスケーリングをサポートできず、4分の1サンプル位置を得るにはハーフサンプル補間が必要であるので、この方法は計算上取り扱いにくい。これら問題を克服するピクチャーアップサンプリング方法が望まれている。

【課題を解決するための手段】

【0009】

10

本発明の実施例は、画像符号化および復号のための方法およびシステムを含む。本発明の一部の実施例は、空間的スケーラビリティエンハンスメントレイヤー内の隣接するブロック間のブロック境界の特性を定めるための方法およびシステムを含む。一部の実施例では、レイヤ間テクスチャー予測を利用して、隣接するブロックのうちの少なくとも1つを符号化する。前記隣接するブロックのうちの1つが指定された基準を満たすと、境界強度インジケータによりブロック境界の特性を定めることができる。

【0010】

添付図面を参照しながら本発明の詳細な説明を検討すれば、本発明の上記およびそれ以外の目的、特徴および利点について、より容易に理解できよう。

【図面の簡単な説明】

20

【0011】

【図1】隣接する画像ブロック間の類似性に従って、デブロックフィルタリングをどのように選択的にスキップするかを示す図である。

【図2】類似する動きベクトルを有する2つの隣接する画像ブロックを示す図である。

【図3】画像ブロックのうちの1つに対し、変換係数をどのように識別するかを示す図である。

【図4】2つの隣接する画像ブロックの間で残差変換係数をどのように比較するかを示す図である。

【図5】ビデオ画像をどのように符号化し、復号するかを示すブロック図である。

【図6】コーデックにおいて、デブロックフィルタリングをどのように選択的にスキップするかを示すブロック図である。

30

【図7】現在のブロックに基づく画像フィルタリング技法を示す図である。

【図8】フィルタ処理をする境界および使用するそれぞれのフィルタの強度を決定するための技法を示すブロック図である。

【図9】本発明の別の実施例を説明するための図である。

【図10】本発明の更に別の実施例を説明するための図である。

【図11】本発明の更に別の実施例を説明するための図である。

【図12】本発明の更に別の実施例を説明するための図である。

【図13】隣接するブロック間のデブロックフィルタリングが隣接するブロック内の符号化パラメータの類似性に依存する本発明の一実施例の処理ステップを説明するフローチャートである。

40

【図14】隣接するブロック間のデブロックフィルタリングが類似する動きベクトルを有する隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

【図15】隣接するブロック間のデブロックフィルタリングが同一の基準フレームを指示する類似する動きベクトルを有する隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

【図16】隣接するブロック間のデブロックフィルタリングが単一基準フレーム内の隣接する基準ブロックを指示する類似する動きベクトルを有する隣接するブロックに依存する、本発明の一実施例のステップを説明するフローチャートである。

50

【図 17】隣接するブロック間のデブロックフィルタリングが、類似する直流変換係数を含むパラメータを有する隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

【図 18】隣接するブロック間のデブロックフィルタリングが、類似する交流変換係数を含むパラメータを有する隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

【図 19】隣接するブロック間のデブロックフィルタリングが、基準フレーム内の類似する動きベクトルおよび類似する動きベクトルターゲットを含むパラメータを有するルミナンス画像内の隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

10

【図 20】隣接するブロック間のデブロックフィルタリングが、基準フレーム内の類似する動きベクトル、類似する動きベクトルターゲットと、類似する変換係数とを含むパラメータを有するルミナンス画像内の隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

【図 21】1つの画像が別個のルミナンスチャンネルとクロミナンスチャンネルとに分割されており、各ルミナンス画像またはクロミナンス画像内の隣接するブロック間のデブロックフィルタリングが、類似する動きベクトルを含むパラメータを有するルミナンス画像内の隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

【図 22】1つの画像が別個のルミナンスチャンネルとクロミナンスチャンネルとに分割されており、各ルミナンス画像またはクロミナンス画像内の隣接するブロック間のデブロックフィルタリングが、基準フレーム内の類似する動きベクトル、類似する動きベクトルターゲット並びに類似する変換係数を含むパラメータを有するルミナンス画像内の隣接するブロックに依存する本発明の一実施例の処理ステップを説明するフローチャートである。

20

【図 23】本発明の一部の実施例におけるベース空間レイヤとエンハンスメント空間レイヤとの間の幾何学的関係を示す図である。

【図 24】本発明の一部の実施例のアップサンプリングされたベースレイヤのピクチャとエンハンスメントレイヤのピクチャとの間の幾何学的関係を示す図である。

【図 25】4×4ブロックのピクセルを示す図である。

30

【図 26】8×8ブロック内の4×4ブロックを示す図である。

【図 27】予測マクロブロックの8×8ブロックを示す図である。

【図 28】隣接するブロック属性に基づき、ブロック境界の特性を定めるための方法の一例を示すフローチャートである。

【図 29 A】隣接するブロック属性に基づき、ブロック境界の特性を定めるための方法の別の例を示すフローチャートである。

【図 29 B】隣接するブロック属性に基づき、ブロック境界の特性を定めるための方法の別の例を示すフローチャートである。

【図 30 A】隣接するブロック属性に基づき、ブロック境界の特性を定めるための方法の更に別の例を示すフローチャートである。

40

【図 30 B】隣接するブロック属性に基づき、ブロック境界の特性を定めるための方法の更に別の例を示すフローチャートである。

【発明を実施するための形態】

【0012】

図面を参照することにより、本発明の実施例が最も良く理解されよう。図全体にわたり、同様な部品は同様な番号で示す。図面はこの詳細な説明の一部として援用する。

【0013】

本明細書で全般的に説明し、図に示すように、本発明の構成要素は、広範な種々の異なる構成で配置し、設計できることが容易に理解できよう。従って、本発明の方法およびシステムの実施例の次のより詳細な説明は、本発明の範囲を制限するものでなく、単に現時

50

点で好ましい本発明の実施例を示すにすぎない。

【0014】

本発明の実施例の要素は、ハードウェア、ファームウェアおよび/またはソフトウェアとして具現化できる。本明細書に開示する実施例は、単にこれら形態のうちの1つを示すにすぎないが、当業者であれば本発明の範囲内でこれら要素をこれら形態のうちのいずれかで実施することができると理解すべきである。

【0015】

従来のフィルタリングプロセスは、一度に1つの再構成された画像フレームを検討する。ブロックに基づく符号化技術は、ピクセルのブロックの動きを推定するのに、動きベクトルを使用する。この動きベクトル情報は、符号化と復号の双方で利用できるが、従来の
10 フィルタリングプロセスでは使用されない。例えば、2つの隣接するブロックが同じ基準画像フレームに対して同じ動きベクトルを共用する場合、(多数の基準フレームシステムでは)各ブロックの残差画像間には大きな差がない可能性が高いので、これらブロックをフィルタリングしてはならない。本質的には、画像の隣接する部分は同じ基準フレームに
20 対して同じ動きを有するので、残差画像間に大きな差がないと予測される。多くのケースでは、これら2つの隣接ブロックのブロック境界を基準フレーム内でフィルタリングすることがあり、従って、現フレームに対して再びフィルタリングしてはならない。動きベクトル情報を検討することなく、逆ブロックフィルタを使用する場合、従来のフィルタリングプロセスはフレーム毎に同じ境界を何回もフィルタリングする可能性があった。このよ
うな不必要なフィルタリングは、不必要なボケを生じさせるだけでなく、結果としてフィ
ルタリング計算を更に行うことにもなる。

【0016】

図1は、画像ブロック間の類似性に従ってブロッキング歪を選択的にフィルタリングする画像12を示す。この画像は、同様に、非矩形ブロックまたはピクセルの他の任意の組も使用できると理解すべきである。ブロック14の一部の間の境は、ブロッキング歪18
を含む。一般に、ブロッキング歪は、符号化および/または復号プロセスから生じ得るブ
ロック14の間の画像の不連続性である。隣接する画像ブロックの境に存在するブロッキ
ング歪を低減するのに、ローパスフィルタまたは他のフィルタを使用できる。

【0017】

例えば、ブロック20と22との間にブロッキング歪24が存在する。ブロッキング歪
24を除くか、あるいはこれを低減するために、ブロック20と22との間の境26でロ
ーパスフィルタを使用できる。このローパスフィルタは、例えば境26の両側からピクセル
のグループ28を選択する。ピクセルのグループ28から平均ピクセル値または他の任
意の統計的尺度を導き、次に、各個々のピクセルと平均ピクセル値とを比較する。次に、
平均ピクセル値の所定のレンジの外にあるグループ28内のピクセルを平均ピクセル値と
置き換える。

【0018】

前に述べたように、隣接するピクセル間にブロッキング歪24がほとんどないか、また
は全くない場合に、ピクセルのグループ28を不必要にフィルタリングし、よって画像内
にボケを生じさせる可能性がある。スキップモードフィルタリング方式は、どれを選択的
にフィルタリングするかに基づき、隣接する画像ブロックに対し、動き推定および/また
は補償情報を使用できる。動き推定情報と補償情報とが十分類似している場合、フィル
タリングをスキップしてもよい。これによって、不要な画像のボケを解消し、フィルタリ
ング演算の必要な回数、または他の任意の適当な値を大幅に低減できる。

【0019】

一例として、符号化プロセス中に隣接する画像ブロック30と32とは、類似する符号
化パラメータを有すると判断できる。従って、隣接するブロック30と32との間の境3
1を横断するピクセルのグループ34に対し、デブロックフィルタリングをスキップでき
る。画像12内の隣接ブロック間の任意の水平線、垂線または他の任意の境界に対し、ス
キップモードのフィルタリングを使用できる。

10

20

30

40

50

【 0 0 2 0 】

図 2 は、基準フレーム 4 2、基準フレーム 4 8 および現在符号化中または復号中である現フレーム 4 0 を示す。2 つの隣接するブロック 4 4 と 4 6 との間でデブロックフィルタリングをスキップすべきかどうかを判断するために、ブロック 4 4 に対する符号化パラメータとブロック 4 6 に対する符号化パラメータとを比較する。比較できる符号化パラメータのうちの 1 つは、ブロック 4 4 および 4 6 に対する動きベクトル (M V) である。

【 0 0 2 1 】

動きベクトル M V 1 は、現在の画像フレーム 4 0 内のブロック 4 4 から基準画像 4 2 内の関連するブロック 4 4 ' を指示する。動きベクトル M V 2 は、現在の画像フレーム 4 0 内のブロック 4 6 から基準フレーム 4 2 内の関連するブロック 4 6 ' を指示する。スキップモードフィルタリングは動きベクトル M V 1 と M V 2 とが同一基準フレーム 4 2 内の隣接するブロックを指示しているかどうかを調べるためにチェックする。これら動きベクトルが同一基準フレーム内の隣接するブロックを指示している場合 (M V 1 = M V 2)、デブロックフィルタリングをスキップしてよい。この動きベクトル情報は、他の符号化情報と共に使用し、2 つの画像ブロック 4 4 と 4 6 との間のデブロックフィルタリングをスキップするかどうかを判断するのに使用される。

【 0 0 2 2 】

符号化および復号プロセス中に 2 つ以上の基準フレームを使用できる。例えば別の基準フレーム 4 8 が存在することがある。隣接するブロック 4 4 および 4 6 は、異なる基準フレームを指示する動きベクトルを有することがある。一例では、デブロックフィルタリングをスキップするかどうかの判断は、2 つの隣接するブロックに対する動きベクトルが同じ基準フレームを指示しているかどうかによって決まる。例えば画像ブロック 4 4 が、基準フレーム 4 8 を指示する動きベクトル 4 9 を有し、画像ブロック 4 6 が基準フレーム 4 2 を指示する動きベクトル M V 2 を有することがある。この例では、動きベクトル 4 9 と M V 2 とは異なる基準フレームを指示するので、デブロックフィルタリングはスキップされない。

【 0 0 2 3 】

図 3 は、デブロックフィルタリングを選択的にスキップするかどうかの判断をするのに使用できる符号化パラメータの別の例を示す。図 2 において、前に示したように、画像フレーム 4 0 からの画像ブロック 4 4 と動きベクトル M V 1 が指し示した基準フレーム 4 2 からの基準ブロック 4 4 ' とを比較する。画像ブロック 4 4 と基準ブロック 4 4 ' と比較することから、残差ブロック 4 4 " が出力される。残差ブロック 4 4 " に対して変換 5 0 を実行し、変換係数をもつ変換されたブロック 4 4 ' ' ' を作成する。一実施例では、変換 5 0 は離散的コサイン変換であり、変換されたブロック 4 4 ' ' ' は直流成分 5 2 と交流成分 5 3 とを含む。

【 0 0 2 4 】

直流成分 5 2 は、残差ブロック 4 4 " 内の最低周波数の変換係数を意味する。例えば残差ブロック 4 4 " 内の平均エネルギーを示す変換係数である。交流成分 5 3 は、残差ブロック 4 4 " 内のより高い周波数の成分を表す変換係数を示す。例えば残差ブロック 4 4 " 内のピクセル間の大きいエネルギー差を表す変換係数のことである。

【 0 0 2 5 】

図 4 は、変換されたブロック 4 4 ' ' ' および 4 6 ' ' ' を示す。これら 2 つの変換ブロック 4 4 ' ' ' および 4 6 ' ' ' からの直流成分 5 2 はプロセッサ 5 4 で比較される。直流成分が同一であるか、または互いにある範囲内にある場合、プロセッサ 5 4 は 2 つの隣接するブロック 4 4 と 4 6 の境間のデブロックフィルタリングをスキップすることをデブロックフィルタ演算 5 6 に通知する。直流成分 5 2 が類似していなければ、スキップの通知を開始せず、ブロック 4 4 と 4 6 との間の境をデブロックフィルタリングする。

【 0 0 2 6 】

一例では、国際通信連合 (I T U - T) の通信セクタが提案した H.26L 符号化方式にスキップモードフィルタリングを組み込んでもよい。この H.26L 方式は 4 × 4 の整数離散コ

10

20

30

40

50

サイン変換 (DCT) ブロックを使用するものである。所望する場合、隣接する 2 つのブロックの直流成分だけをチェックできる。しかしながら、特に画像ブロックがより大きいサイズ、例えば 8×8 ブロックまたは 16×16 ブロックであるときに、ある限定された低周波交流係数を同様にチェックすることができる。例えばブロック 4×4 に対する上部直流成分 5 2 および 3 つの低周波数の交流変換係数 5 3 とブロック 4×6 に対する上部直流成分 5 2 および 3 つの低周波数の交流変換係数 5 3 とを比較することができる。直流変換係数および / または任意の交流変換係数との異なる組み合わせを使用し、2 つの隣接するブロック 4×4 と 4×6 との間の相対的類似性を識別できる。

【0027】

プロセッサ 5 4 は、符号化プロセス中に生成された他の符号化パラメータ 5 5 も受信できる。これら符号化パラメータは、前に述べたように、隣接するブロック 4×4 および 4×6 に対する動きベクトルおよび基準フレーム情報を含む。プロセッサ 5 4 は、これら符号化パラメータのうちの一部または全てを使用し、隣接する画像ブロック 4×4 と 4×6 との間のデブロックフィルタリングをスキップするかどうかを判断できる。画像に対して実行される他の符号化および変換機能は、同じプロセッサ 5 4 または異なる処理回路で実行することができる。同じプロセッサで符号化の全てまたはほとんどを実行する場合、フィルタリングルーチンでスキップパラメータを設定することにより、簡単にスキップモードも可能にすることができる。

【0028】

図 5 は、ブロックに基づく動き補償符号化 / 復号 (コーデック) 6 0 内でどのようにスキップモードのフィルタリングを使用できるかを示す。このコーデック 6 0 は、フレーム間符号化のために使用される。ボックス 6 2 からコンパレータ 6 4 へ、現フレームからの入力ビデオブロックが送られる。フレームバッファリングボックス 8 0 の出力は、推定動きベクトル (および可能な基準フレーム番号) に従って基準ブロック 8 1 を生成する。ボックス 6 6 内で入力ビデオブロックと基準ブロック 8 1 との間の差が変換され、ボックス 6 8 内で量子化される。ボックス 7 0 内の可変長符号化 (VLC) により、量子化された変換ブロックが符号化され、次に送信され、格納等される。

【0029】

コーデック 6 0 の符号化セクションは、ボックス 7 2 内で変換された画像を第 1 逆量子化 (IQ) することにより、変換され、量子化された画像を再構成する。次に、この逆量子化された画像は、ボックス 7 4 内で逆変換され、再構成された残差画像を生成する。次に、ボックス 7 6 内で、この再構成された残差ブロックが基準ブロック 8 1 に追加され、再構成された画像ブロックが生成される。一般に、この再構成された画像はボックス 7 8 内でループフィルタがかけられ、量子化および変換プロセスによって生じたブロック歪が低減される。次に、フィルタにかけられた画像はボックス 8 0 内でバッファ化され、基準フレームが形成される。ボックス 8 0 内のフレームバッファリングは、動き推定および補償を行うために、再構成された基準フレームを使用する。コンパレータ 6 4 内で基準ブロック 8 1 と入力ビデオブロックとを比較する。符号化セクションから、符号化された画像が出力され、次に格納または送信される。

【0030】

コーデック 6 0 の復号部分では、可変長復号 (VLD) がボックス 8 2 内で、符号化された画像を復号する。復号された画像は、ボックス 8 4 内で逆量子化され、ボックス 8 6 内で逆変換される。ボックス 8 6 からの再構成された残差画像は、ブロック歪を低減するためにボックス 9 0 内でループフィルタリング処理される前に、総和ボックス 8 8 内で基準ブロック 9 1 に加えられ、基準フレームとしてボックス 9 2 内に一時蓄積される。受信された動きベクトル情報に従い、ボックス 9 2 から基準ブロック 9 1 が生成される。ボックス 9 0 からのループフィルタ処理された出力を、ボックス 9 6 内でビデオ画像としてディスプレイする前に、画像の歪を更に低減するために、オプションとしてボックス 9 4 内でポストフィルタ処理することができる。ボックス 7 8、9 0 および 9 4 内のフィルタリング機能の任意の組み合わせで、スキップモードフィルタリング方式を実行できる。

10

20

30

40

50

【 0 0 3 1 】

ボックス 7 8、9 0 および / または 9 4 内のデブロックフィルタリングをいつスキップするかを判断するために、ビデオ符号化中に利用できる動き推定および補償情報を使用できる。符号化および復号プロセス中に、これら符号化パラメータは既に生成されているので、特にスキップモードフィルタリングのために生成したり、伝送しなければならない別の符号化パラメータは存在しない。

【 0 0 3 2 】

図 6 は、図 5 内の符号化および復号内のフィルタ 7 8、9 0 および / または 9 4 内でのようにスキップモードフィルタリングを使用できるかを、更に詳細に示している。まず、ボックス 1 0 0 内では、任意の 2 つの隣接するブロック “ j ” と “ k ” との間のブロック間境界が識別される。画像フレーム内では、これら 2 つのブロックは水平方向または垂直方向に隣接することがある。判断ボックス 1 0 2 は、ブロック j のための動きベクトル $m v(j)$ と、ブロック k のための動きベクトル $m v(k)$ とを比較する。まず、2 つの隣接するブロック j と k とが、同じ基準フレームを指示する同じ動きベクトルを有するかどうかを判断する。換言すれば、隣接するブロックのための動きベクトルは同一基準フレーム ($ref(j) = ref(k)$) 内の隣接ブロック ($m v(j) = m v(k)$) を指示する。

【 0 0 3 3 】

次に、隣接する 2 つのブロックに対する残差係数が類似しているかどうかを判断する。隣接するブロックの残差画像間に大きな差がない場合、例えば 2 つのブロック j と k とが同一または近似する直流成分 ($dc(j) = dc(k)$) となっている場合、ボックス 1 0 4 内のデブロックフィルタリングプロセスがスキップされる。次にスキップモードフィルタリングは、ボックス 1 0 6 内の次のブロック間境界へ移動し、判断ボックス 1 0 2 内で次の比較を実行する。スキップモードフィルタリングは水平方向に隣接するブロックおよび垂直方向に隣接するブロックのいずれにも実行できる。

【 0 0 3 4 】

一実施例では、ブロックをスキップすることを判断するために、隣接する画像ブロックに対する基準フレームおよび動きベクトル情報だけを使用する。別の実施例では、ブロックをスキップすることを判断するために、直流および / または交流残差係数だけを使用する。別の実施例では、ブロックをスキップすることを判断するために動きベクトル、基準フレームおよび残差係数の全てを使用する。

【 0 0 3 5 】

空間的にサブサンプリングされたクロミナンスチャンネルに対し、スキップモードのフィルタリング方式を使用できる。例えば 4 : 2 : 0 のカラーフォーマットシーケンスを有するケースでは、ブロック境界に対するスキップモードフィルタリングは、画像のルミナンス成分に対する動きベクトルの質および直流成分だけに依存し得る。動きベクトルと直流成分とが同一である場合、隣接する画像ブロックのルミナンス成分およびクロミナンス成分の双方に対して、デブロックフィルタリングをスキップする。別の実施例では、隣接するブロックの各ルミナンス成分およびクロミナンス成分に対し、動きベクトルおよび直流成分を別々に検討する。この場合、隣接するブロックに対するルミナンス成分またはクロミナンス成分をデブロックフィルタリングできるが、同じ隣接ブロックに対する他のルミナンス成分またはクロミナンス成分をデブロックフィルタリングすることはない。

【 0 0 3 6 】

図 7 を参照すると、一部の公知の技術は、ループフィルタリングプロセスを制御するためにループフィルタのためのブロック強度パラメータを定めている。画像の各ブロックは、そのブロックに関連した強度の値を有し、4 つのブロック境界のうちのすべてで実行されるフィルタリングを制御する。このブロック強度の値は、ビットストリーム内で入手できる動きベクトルおよび変換係数に基づいて導かれる。しかしながら、ブロックのうちの 4 つのすべてのエッジに対し、ブロック強度を使用することを検討した後に、本発明者たちは、このことは、あるエッジにおける一部のブロックング歪を除去できるが、他のエッ

10

20

30

40

50

ジに沿って不必要にボケを生じさせるという結果になることを実現させるに至った。

【 0 0 3 7 】

ブロックごとのフィルタリングと対照的に、本発明者たちは他の情報と共にエッジごとにフィルタリングの判断をしなければならないことが分かった。他の情報として、例えばブロックのブロック内符号化、残差情報を持つブロックの動き推定、残差情報を持たないブロックの動き推定および十分な差を有する残差情報を持たないブロックの動き推定を上げることができる。これら情報特性のうちの1つ、2つ、3つまたは4つを使用し、エッジごとのフィルタリング能力を改善できる。これら特性の異なる組に基づき、所望するようにフィルタリングを変えることができる。

【 0 0 3 8 】

各ブロック境界に対し、制御パラメータ、すなわち境界強度 B_s を定めることが好ましい。図8を参照すると、共通する境界を共有する一对のブロックを j および k と称す。第1ブロック110は、2つのブロックのうちの一方がイントラ符号化されているかどうかを見るようにチェックする。いずれかがイントラ符号化されている場合、ブロック112にて境界強度を3にセットする。ブロック110は、ブロックの双方が動き予測されていないかどうかを判断する。動き予測が使用されていない場合、ブロックはフレーム自身から導き、従って境界上でフィルタリングを実行しなければならない。イントラ符号化されたブロック境界は普通ブロッキング歪を含むので、このことは通常適切である。

【 0 0 3 9 】

ブロック j および k の双方が少なくとも部分的に前のフレームまたは将来のフレームから予測される場合、ブロック114においてブロック j および k をチェックし、任意の係数を符号化するかどうかを判断する。これら係数は、例えば離散的コサイン変換係数とすることができる。ブロック j および k のいずれかが0でない係数を含む場合、ブロックのうちの少なくとも一方は、一般に残差と称される係数を使用するブロックへの変更と共に、前のフレームまたは将来のフレームからの予測を示す。ブロック j および k のいずれかが、0でない係数（および予測される動き）を含む場合、境界強度はブロック116にて、2にセットされる。このことは、画像が予測される事象の発生を示すが、この予測は残差を使って修正される。従って、これらの画像はブロッキング歪を含む可能性がある。

【 0 0 4 0 】

ブロック j および k の双方が、動き予測され、一般に残差係数と称される非ゼロ係数を含まない場合、境界のどちら側かでのピクセルが互いに十分異なるかどうか、ブロック118で判断される。この判断は同様に、残差係数が十分小さいかどうかを判断するのにも使用できる。十分な差がある場合、ブロッキング歪が存在する可能性がある。最初に2つのブロックが異なる基準フレームを使用しているかどうか、すなわち $R(j) \neq R(k)$ であるかどうかの判断を行う。ブロック j と k とが2つの異なる基準フレームからのものである場合、ブロック120にて1の値を境界強度に割り当てる。これとは異なり、2つの画像ブロックの動きベクトルの絶対値の差をチェックし、それが垂直方向または水平方向のいずれかに1ピクセル以上であるかどうかを判断する。すなわち、

【 0 0 4 1 】

【 数 1 】

$$|V(j,x)-V(k,x)| \geq 1 \text{ pixel or } |V(j,y)-V(k,y)| \geq 1$$

【 0 0 4 2 】

を判断する。所望すれば、使用するテストに応じ、1未満または1より大きい他の閾値も同じように使用できる。動きベクトルの絶対値の差が1以上である場合、境界強度に1の値を割り当てる。

【 0 0 4 3 】

2つのブロック j および k が動き予測され、残差がなく、同じフレームに基づいており

、わずかな差しか有しない場合、境界強度の値に 0 の値を割り当てる。境界強度の値に 0 の値を割り当てた場合、境界をフィルタリングしないか、または他の方法で境界強度の値に適合してフィルタリングする。このシステムは、所望すれば、境界強度が 0 の場合、軽いフィルタリングを行うことができると理解すべきである。

【 0 0 4 4 】

境界強度の値、すなわち 1、2 および 3 を使用して、ループフィルタ内のピクセル値適応レンジを制御する。所望すれば、異なる各境界強度を異なるフィルタリングの基礎とすることができる。例えば、一部の実施例では、3 つの種類のフィルタを使用できる。例えば $B_s = 1$ のときに第 1 フィルタを使用し、 $B_s = 2$ のときに第 2 フィルタを使用し、 $B_s = 3$ のときに第 3 フィルタを使用する。結果として、より大きな差を生じさせる他のフィルタリングと比較し、最小フィルタリングにより非フィルタリングを実行できることがわかる。図 8 に示された例では、 B_s の値が大きくなればなるほど、フィルタリングも大きくなる。このフィルタリングは、任意の適切な技術、例えば ISO/IEC MPEG と ITU-T VCEG (JVT-C167) の共同ビデオチーム (JVT) の共同委員会草稿 (CD) に記載されているような方法、または画像歪をフィルタリングするための他の公知の方法によって実行できる。

【 0 0 4 5 】

マルチ画像フレームを符号化または復号する任意のシステムと共に、スキップモードフィルタリングを使用できる。例えば DVD プレイヤー、ビデオレコーダまたはテレビチャンネルもしくはインターネットのような通信チャンネルを通して画像データを伝送する任意のシステムである。このシステムは、符号化パラメータとして量子化パラメータを単独または他の符号化パラメータと組み合わせて使用することができる。更に、このシステムは量子化パラメータを単独で使用しなくてもよいし、またはフィルタリング目的のために量子化パラメータを全く使用しなくてもよい。

【 0 0 4 6 】

上記スキップモードフィルタリングは専用プロセッサシステム、マイクロコントローラ、プログラマブルロジックデバイス、または前記演算の一部または全てを実行するマイクロプロセッサを用いて実現できる。上記演算の一部をソフトウェアで実施し、他の演算をハードウェアで実施してもよい。

【 0 0 4 7 】

便宜上、前記演算は種々の相互接続された機能ブロックまたは異なるソフトウェアモジュールとして記述される。しかしながら、このことは必要ではなく、これら機能ブロックまたはモジュールが単一のロジックデバイス、プログラムまたは境界の不明瞭な演算に等価的に集約されるケースもあり得る。いずれの場合においても、機能ブロックおよびソフトウェアモジュール、もしくは説明した特徴をそれらだけで実施したり、もしくはハードウェアまたはソフトウェアの他の演算と組み合わせて実現することもできる。

【 0 0 4 8 】

図 9 に示されるような本発明の一部の実施例では、画像データ符号化装置 9 0 4 内に画像データ 9 0 2 を入力できる。この符号化装置 9 0 4 は、本発明の一部の実施例に対してこれまで説明したような適応型フィルタリング部分を含む。画像データ符号化装置 9 0 4 からの出力は、符号化された画像データであり、任意の、コンピュータで読み取り可能な記憶媒体 9 0 6 に記憶できる。記憶媒体としてディスク媒体、メモリカード媒体、またはデジタルテープ媒体を挙げることができるが、これらだけに限定されるものではない。記憶媒体 9 0 6 は、短期記憶デバイスとして機能する。符号化された画像データはこの記憶媒体 9 0 6 から読み出され、本発明の一部の実施例に対してこれまで説明したような適応型フィルタリング部分を含む。画像データ復号装置 9 0 8 によって復号される。この復号された画像データは、出力・復号化画像データ 9 1 0 としてディスプレイまたは他のデバイスへ提供される。

【 0 0 4 9 】

図 1 0 に示されるような本発明の一部の実施例では、画像データ 1 0 0 2 を符号化し、次に記憶媒体 1 0 0 6 に符号化された画像データを記憶することができ、画像データ復号

10

20

30

40

50

装置 1008 は図 9 に示されたものと同一である。図 10 では、Bs データ符号化部 1012 は、各ブロック境界に対する境界強度 Bs の値を受信し、DPCM、多値ランレングス符号化、無損失特徴を有する変換符号化などを含む任意のデータ符号化方法によってこの値を符号化する。図 8 に示されるように境界強度 Bs を発生できる。次に、符号化された境界強度は記憶媒体 1006 に記憶される。一実施例では、符号化された境界強度を符号化された画像データとは別個に記憶できる。別の例では、符号化された境界強度と符号化された画像データは、記憶媒体 1006 上に記憶する前に多重化される。

【0050】

符号化された境界強度を記憶媒体 1006 から読み出し、Bs データ復号部 1014 により復号し、画像データ復号装置 1008 へ復号された境界強度を入力することができる。本発明の適応型フィルタリングを実行するために、画像データ復号装置 1008 で復号された境界強度を利用するとき、境界強度を発生するために図 8 に記載したプロセスを繰り返す必要はなく、これによって適応型フィルタリングのための処理パワーを節約できる。

10

【0051】

図 11 に示されるような本発明の一部の実施例では、本発明の一部実施例に対してこれまで説明したような適応型フィルタリング部分を含む画像データ符号化装置 1104 に画像データ 1102 を入力できる。画像データ符号化装置 1104 からの出力は、符号化された画像データであり、ネットワーク、例えば LAN、WAN またはインターネット 1106 を通して送ることができる。符号化された画像データは、ネットワーク 1106 とも通信する画像復号装置 1108 により受信し、復号される。画像データ復号装置 1108 は、本発明の一部の実施例に対してこれまで説明したような適応型フィルタリング部分を含む。復号された画像データは復号された出力画像データ 1110 としてディスプレイまたは他のデバイスへ提供される。

20

【0052】

図 12 に示したような本発明の一部の実施例では、画像データ 1202 を符号化し、符号化された画像データを、次にネットワーク、例えば LAN、WAN またはインターネット 1206 を通して送ることができる。この画像データ符号化装置 1204 および画像データ復号装置 1208 の基本的手順は、図 11 のものと同じである。図 12 では、B のデータ符号化部 1212 は、各ブロックに対する境界強度 Bs の値を受信し、DPCM、多値ランレングス符号化、無損失特徴を有する変換符号化などを含む任意のデータ符号化方法によってこの値を符号化する。図 11 に示されるように境界強度 Bs を発生できる。次に、符号化された境界強度はネットワーク 1206 を通して送ることができる。一例では、符号化された境界強度を符号化された画像データとは別個に送ることができる。別の例では、ネットワーク 1206 を通して送る前に、符号化された境界強度と符号化された画像データとを多重化することができる。

30

【0053】

符号化された境界強度をネットワーク 1206 から受信し、Bs のデータ復号部 1214 により復号し、復号した境界強度を画像データ復号装置 1208 に入力し、本発明の適応型フィルタリングを実行する。境界強度を発生するのに、図 11 に示されたプロセスを繰り返す必要はなく、これによって適用型フィルタリングのための処理パワーを節約できる。

40

【0054】

図 13 を参照して、本発明の一部の実施例について説明できる。これらシステムおよび方法では、ビデオフレーム内の隣接するブロック 150 を識別し、これら隣接するブロックに対する符号化パラメータを識別する。次に隣接するブロックに対する符号化パラメータを比較し、それらの類似性 154 を判断する。これら符号化パラメータが類似していないとき、隣接するブロック間の境界に沿ってデブロックフィルタ 156 を使用する。符号化パラメータが類似しているときは、デブロックフィルタリングをスキップし、プロセスは次のステップ 158 まで進む。同様に、デブロックフィルタリングを実行するとき、フ

50

フィルタリング後に、プロセスは次のステップ 158 に進む。

【0055】

図 14 に示されるような本発明の一部の実施例では、符号化パラメータは動きベクトルである。これら実施例では、ビデオフレーム内の隣接するブロックを識別し (160)、動きベクトルを含む符号化パラメータを識別する (162)。これら動きベクトルを比較し、それらの類似性を判断する (164)。動きベクトルが類似していないと、隣接するブロック間でデブロックフィルタリングを実行 (166) し、プロセスは次のステップ 168 に進む。動きベクトルが類似しているとき、デブロックフィルタリングをスキップし、次のステップ 168 を直接実行する。

【0056】

図 15 に示されるような本発明の別の実施例は、フィルタリングをスキップするのかどうかの判断のために、多数の符号化パラメータを利用できる。これら実施例では、隣接するブロックを識別し (170)、隣接するブロックに対する符号化パラメータが決められる (172)。これら符号化パラメータは、動きベクトルのターゲットフレームを含む動きベクトルの属性を含むことができる。隣接するブロックの動きベクトルが類似していないとき (174)、隣接するブロック間のデブロックフィルタリングを実行できる (176)。動きベクトルが類似しているとき (174)、更にフィルタリングプロセスを限定するために別のパラメータを使用できる。この場合、動きベクトルを比較し、これらベクトルが同じ基準フレームを指し示しているかどうかを判断できる (178)。これらベクトルが同じ基準フレームを指し示していない場合、ブロックの間にデブロックフィルタリングを実行できる (176)。これらベクトルが同じ基準フレームを指し示している場合、フィルタリングをスキップし、プロセスは次のステップまで進むことができる (179)。

【0057】

更に別の動きベクトルパラメータを使ってフィルタリングを決定できる。図 16 に示された実施例では、ベクトルが指し示しているブロックのロケーションは、フィルタリングオプションを決定するのに使用できるパラメータである。これら実施例では隣接するブロックを識別 (200) し、隣接するブロックに対する符号化パラメータを識別する (202)。次に動きベクトルを比較し、それらの類似性を判断する (204)。これらベクトルが類似していない場合、デブロックフィルタリングを続けることができる (208)。動きベクトルが類似している場合、別の比較を行い、隣接するブロックの動きベクトルが同じ基準フレームを指し示しているかどうかを判断される。ベクトルが同一フレームを指し示していない場合、デブロックフィルタリングを続けることができる (208)。ベクトルが同一基準フレームを指し示している場合、これらベクトルが指し示しているブロックが比較される (210)。動きベクトルが同一基準フレーム内の隣接するブロックを指し示していない場合、デブロックフィルタリングを進めることができる (208)。ベクトルが同一基準フレーム内の隣接するブロックを指し示しているとき、デブロックフィルタリングをスキップし、次のステップが実行される (212)。このように、基準フレーム内の隣接するブロックを基準とし、ブロック間に大きい歪を有する可能性のない隣接するブロックはデブロックフィルタリングされない。このようなデブロックフィルタリングのスキップによって、フィルタリングプロセスによって生じるばやけおよび画像の劣化を防止できる。不必要なフィルタリングを回避できるので、処理時間も節約できる。従って、画質が改善され、プロセス内ではわずかな計算で済む。フィルタスキップを決定するのに、これら動きベクトルパラメータの種々の組み合わせを使用できることに留意すべきである。特にこれら多数の組み合わせについては、詳細に説明しないが、これら組み合わせは当業者が想到できるものであり、特許請求の範囲内に入るものである。

【0058】

本発明の別の実施例は、デブロックフィルタリングを行うべきかどうかを判断するのに変換係数を利用できる。図 17 を参照すると、フレーム内の隣接するブロックを識別し (180)、隣接するブロックに対する符号化パラメータを識別する (182)。これら符

10

20

30

40

50

号化パラメータは、動きベクトルパラメータだけでなく、変換係数も含むことができる。

【 0 0 5 9 】

次に、動きベクトルを比較 (1 8 4) し、類似性を判断する。動きベクトルが類似していなければ、デブロックフィルタリングが実行される (1 8 6)。動きベクトルが類似していれば、動きベクトルデータを分析し、動きベクトルが同じ基準フレームを指し示しているかどうかを判断する。動きベクトルが同一の基準フレームを指し示していない場合 (1 8 5)、フィルタリングを進めることができる (1 8 6)。

【 0 0 6 0 】

動きベクトルが同一の基準フレームを指し示している場合、更にフィルタリングプロセスの限定をするために変換係数が比較される。本例では、離散的コサイン変換 (D C T) 方法またはその他の方法を使って得られた直流変換係数が隣接するブロックに対して比較される。直流変換係数が類似していない場合 (1 8 7)、デブロックフィルタリングを実行できる (1 8 6)。直流変換係数が類似している場合、フィルタリングをスキップし、この方法およびシステムは次のステップ 1 8 8 に進むことができる。

【 0 0 6 1 】

本発明の更に別の実施例は、フィルタリングオプションを決定するために交流変換係数を利用できる。図 1 8 を参照すると、ここには交流変換係数を評価する別のステップを備えた、図 1 7 を参照して説明した実施例に類似する実施例が示されている。これら実施例では、ブロック 1 9 0 およびそれらの符号化パラメータを識別する (1 9 1)。動きベクトルの類似性 1 9 2、動きベクトルのターゲットフレームの類似性 1 9 3 および直流変換係数における類似性も比較する (1 9 4)。これらパラメータにおける類似性が存在する場合、交流変換係数を比較する (1 9 6)。これらが類似している場合、デブロックフィルタリングをスキップし、プロセス内の次のステップを実行する (1 9 7)。交流係数が類似していなければ、隣接するブロック間でフィルタリングを実行し、プロセスは次のステップに進む (1 9 7)。

【 0 0 6 2 】

交流変換係数は、より大きいブロックにおいて重要性を有する可能性が高いが、より小さいブロック、例えば 4×4 ブロックを利用する方法でも使用することができる。

【 0 0 6 3 】

本発明の一部の実施例では、使用する画像のフォーマットおよびカラースペースに応じて 1 つの画像を種々のルミナンスチャンネルおよびクロミナンスチャンネルに分離できる。次の実施例では、Y U V カラースペースについて説明する。しかしながら、これら実施例では他の多くのフォーマットおよびカラースペースが使用される。CIE LAB、YCrCb および他のスペースも使用できる。別の実施例では、R G B のようなカラースペースも使用できる。

【 0 0 6 4 】

図 1 9 を参照し、本発明の一部の実施例について説明する。これら実施例では、画像からルミナンスデータを抽出し、ルミナンス画像を作成する (2 2 0)。次にルミナンス画像において、隣接するブロックを識別 (2 2 2) し、隣接ブロックに対する符号化パラメータも識別する (2 2 4)。他の実施例の場合のように、隣接するブロックの動きベクトルを比較し、類似性について判断する (2 2 6)。動きベクトルが類似していないとき、デブロックフィルタリングを実行する (2 3 0)。ベクトルが類似しているときには別の分析を実行し、ベクトルが同一基準フレームを指し示しているかどうかを判断する (2 2 8)。ベクトルが異なる基準フレームを指し示しているとき、ルミナンス画像内の隣接するブロックに対応する元の画像の隣接するブロック間でデブロックフィルタリングを実行する (2 3 0)。ベクトルが同じフレームを指し示しているときには、デブロックフィルタリングをスキップし、前のフィルタリングを実行することなく、次のステップを実行する (2 3 2)。フィルタリングを実行するとき、フィルタリングプロセス後に次のステップを実行する (2 3 2)。従って、ルミナンスチャンネル内のデータの分析を使って、ルミナンスデータおよびクロミナンスデータの双方を含む元の画像内のフィルタリングプロ

セスを決定する。

【0065】

図20に示される別の関連する実施例では、ルミナンス画像を作成(240)し、ルミナンスおよび元の画像内で対応する隣接ブロックを識別する(242)。ルミナンス画像ブロックに対する符号化パラメータも識別される(244)。その後、動きベクトルを比較し、類似性を判断する(246)。重大な類似性が存在しない場合、元の画像内の隣接するブロック間でフィルタリングを実行する。動きベクトルが類似している場合、動きベクトルのターゲットフレームを比較し、ベクトルが同一基準フレームを指し示しているかどうかを判断する。ベクトルが同一基準フレームを指し示していない場合、フィルタリングを実行する。ベクトルが同一基準フレームを指し示している場合、ルミナンス(Y)画像の変換係数を比較する。Y変換係数が類似していない場合、フィルタリングを実行する。変換係数が類似している場合、フィルタリングをスキップし、次のステップ254を実行する。同様に、フィルタリング演算の後で次のステップを実行する(254)。

10

【0066】

画像を、ルミナンスチャンネルおよびクロミナンスチャンネルに一般に対応する成分チャンネルに更に分割できる。本発明の一部の実施例では、そのチャンネルにユニークなパラメータに従って各チャンネルをフィルタリングできる。

【0067】

一例として、図21を参照し、実施例について説明する。これら実施例では、画像は別個のルミナンス(Y)チャンネルおよび多数のクロミナンス(U、V)チャンネルに分割されている(260)。これら実施例では、各チャンネルに対応する画像内で隣接するブロックを識別する(262、272、282)。各チャンネルにおけるこれらのブロックに対して動きベクトルデータのような符号化パラメータも識別される(264、274、284)。これら符号化パラメータを比較し、他の実施例の場合のように類似性を判断できる。これら実施例では、各チャンネルにおけるフィルタリングオプションを決定するために、チャンネル固有の動きベクトルに対する動きベクトルの類似性を使用できる。チャンネル画像に対する動きベクトルが類似していない場合、隣接するブロック間の特定のチャンネル内でフィルタリングが実行される(270、280、290)。動きベクトルが類似している場合、ターゲット基準フレームを比較する(268、278、288)。チャンネル内の隣接するブロックに対するベクトルが同一基準フレームを指し示している場合、フィルタリングをスキップする。ベクトルが異なる基準フレームを指し示している場合、フィルタリングを実行する(270、280、290)。

20

30

【0068】

別の実施例の場合のように、これらチャンネル化された実施例はフィルタオプションを限定するのに変換係数データを利用できる。図22に示されるように、図21を参照して説明した方法およびシステムは、チャンネル変換係数を更に比較できる(310、322、334)。これら係数が類似していない場合、フィルタリングを実行する(312、324、336)。これら係数が類似している場合、フィルタリングをスキップする。

【0069】

各チャンネルにおけるフィルタリング演算を限定する場合、パラメータの種々の組み合わせを利用できることに留意すべきである。これら実施例に対し、直流変換係数および交流変換係数を利用できる。更に、フィルタリングオプションを決定し、フィルタリングを実行するために、種々のチャンネルおよびこれらチャンネルの組み合わせを使用できる。例えば一部の実施例では、双方のクロミナンスチャンネルを組み合わせ、共に分析できる。1つのチャンネルからのデータおよびパラメータを使用し、別のチャンネルでのフィルタリングオプションを決定することもできる。例えばUクロミナンスチャンネルから取り込んだパラメータを比較し、Vクロミナンスチャンネルにおけるフィルタリングオプションを決定したり、その逆を行うことができる。

40

【0070】

本発明の一部の実施例は、H.264/AVCのスケーラブルなビデオ符号化拡張に関連するも

50

のである。一部の実施例では、空間的スケーラブルなビデオ符号化のためのピクチャアップサンプリングの問題に取り組むためのフィルタリングに関係している。より詳細には、本発明の一部の実施例は、H.264/MPEG-4 AVCのスケーラブルビデオ符号化拡張、特にJVT（MPEGとVCEGの共同ビデオチーム）により、2005年4月に採用された拡張空間的スケーラブル（ESS）ビデオ符号化機能のために設計されたアップサンプリング手順を提供するものである。

【0071】

目下のところ、本明細書で参考例として援用するJSVM WD-1.0[MPEG Doc.N6901]は、ディアドリック空間的スケーラビリティ、すなわち2つの連続する空間レイヤのピクチャ幅と高さの比（ピクセル数に換算した）が2に等しい構成を用意するだけである。このことは、明らかにより一般的なアプリケーション、例えば放送用のSDからHDのスケーラビリティに対する制限となる。

【0072】

拡張された空間的スケーラビリティ、すなわち2つの連続する空間レイヤのピクチャ幅と高さの比が必ずしも2の累乗に等しくなく、より高いレベルのピクチャがより低いレベルの対応するピクチャ内に存在しない領域（一般にピクチャの境のまわりの領域）を含むことがあり得る、管理構成を提供するツール[MPEG Doc.m11669]（本明細書で参考例として援用する）がこれまで提案されている。この提案[MPEG Doc.m11669]は、より高いレイヤのピクチャディメンジョンとより低いレイヤのピクチャディメンジョンの比が2の累乗ではないような、より包括的なケースに対するWD-1.0[MPEG Doc.N6901]のレイヤ間予測を拡張したものである。

【0073】

本発明の実施例は、より一般的なアプリケーションのニーズにより良好に適合するピクチャレベルに対し、拡張された空間的スケーラビリティ、すなわちクロッピングウィンドウを有する非ディアドリックなスケーリングを適用する方法を提供するものである。空間的スケーラビリティのピクチャレベルの適応をサポートするために、本発明の実施例は、これまで提案されたレイヤ間予測方法のより精巧なものを提供している。更に、従来の提案では用意されなかったいくつかの問題も、これら実施例で取り扱われている。

【0074】

本明細書および特許請求の範囲の目的のために「ピクチャ」なる用語はピクセルのアレイ、デジタル画像、デジタル画像の細分割、デジタル画像のデータチャンネルまたは画像データの別の表現を含み得る。

【0075】

図23は、画像ピクチャに対応する2つのピクチャを示す。

【0076】

本発明の実施例は2つ以上の連続する空間レイヤ、より低いレイヤ（ベースレイヤとして見なされる）253およびより高いレイヤ（エンハンスメントレイヤとして見なされる）251に関係する。これらレイヤは次の幾何学的関係によってリンクできる（図23に示されている）。エンハンスメントレイヤピクチャの幅250および高さ252は、それぞれ w_{enh} および h_{enh} として定義できる。同様に、ベースレイヤピクチャのディメンジョンも w_{base} 254および h_{base} 256として定義できる。ベースレイヤ253は、エンハンスメントレイヤピクチャの座標系における座標262（ x_{orig} 、 y_{orig} ）に位置するディメンジョン $w_{extract}$ 258および $h_{extract}$ 260のエンハンスメントレイヤピクチャ251のサブ領域のサブサンプリング264されたバージョンとし得る。パラメータ（ x_{orig} 、 y_{orig} 、 $w_{extract}$ 、 $h_{extract}$ 、 w_{base} 、 h_{base} ）は、より高いレイヤのピクチャ251とより低いレイヤのピクチャ253との間の幾何学的関係を定める。

【0077】

本発明の実施例によって用意される問題は、復号されたベースレイヤを知ることによってエンハンスメントレイヤのマクロブロックを符号化/復号することである。エンハンス

10

20

30

40

50

メントレイヤのマクロブロックは、図 2 4 に示されるように、ベースレイヤに対応するブロックを有しないか（エンハンスメントレイヤピクチャの境上に）、または 1 個～数個のベースレイヤに対応するマクロブロックを有し得る。従って、WD-1.0 [MPEG Doc.N6901] におけるものとは異なるレイヤ間予測の管理が必要である。図 2 4 は、点線 2 7 4 によりマクロブロックの境界をマークしたアップサンプリングベースレイヤピクチャ 2 7 2 と、マクロブロックの境界を実線 2 7 6 でマークしたエンハンスメントレイヤピクチャ 2 7 0 との間にオーバーラップするマクロブロックを示す。

【 0 0 7 8 】

これまでに [MPEG Doc.m1669] において、 $w_{extract}$ および $h_{extract}$ を 16 の倍数に制限することが提案されている。この制限は、ピクチャレベルの適応化を制限する。この代わりに、本発明の実施例は、 $w_{extract}$ および $h_{extract}$ を 2 の倍数に制限するものである。クロミナンスのアップ/ダウンサンプリングにおける起こり得るフェーズシフトを調整する際の複雑さを解消するために、更に本発明の実施例は、 x_{orig} および y_{orig} を 2 の倍数にすることをさらに要求する。クロミナンスフェーズシフトの問題は、これまでには取り組まれていない。

【 0 0 7 9 】

図 2 3 に示されたディメンジョンおよびその他のパラメータを次の記号または変数名で表示できる。

【 0 0 8 0 】

【 数 2 】

- $scaled_base_left_offset = x_{orig}$
- $scaled_base_top_offset = y_{orig}$
- $scaled_base_right_offset = w_{enh} - x_{orig} - w_{extract}$
- $scaled_base_bottom_offset = h_{enh} - y_{orig} - h_{extract}$
- $scaled_base_width = w_{extract}$
- $scaled_base_height = h_{extract}$

【 0 0 8 1 】

レイヤ間動き予測

所定の高いレイヤのマクロブロックは「BASE__LAYER__MODE」または「QP__REFINEMENT__MODE」のいずれかを使用するスケーリングされたベースレイヤの動きデータを使って、レイヤ間予測を利用できる。WD-1.0 [MPEG Doc.N6901] の場合のように、これらマクロブロックモードは、マクロブロック分割を含む動き/予測情報がベースレイヤから直接得られることを示している。ベースレイヤからの動きデータを引き継ぐことにより、予測マクロブロック MB__pred を構成できる。「BASE__LAYER__MODE」を使用するとき、基準インデックスおよび動きベクトルと同じように、マクロブロック分割は予測マクロブロック MB__pred の分割である。「QP__REFINEMENT__MODE」も同様であるが、4 分の 1 サンプルの動きベクトル改良点を有する点が異なっている。

【 0 0 8 2 】

次の 4 つのステップで、MB__pred を導くことが、これまで提案されている。

- MB__pred の各 4×4 ブロックに対し、ベースレイヤの動きデータからの動きデータの引き継ぎ
- MB__pred の各 8×8 ブロックに対する分割の選択
- MB__pred に対するモード選択および
- 動きベクトルのスケーリング

【 0 0 8 3 】

しかしながら、本発明の実施例はピクチャレベルの適応化をサポートするためにいくつ

かの式において変更を行う。

【 0 0 8 4 】

4 × 4 ブロックの引き継ぎ

図 2 5 は、4 つのコーナー 2 8 1、2 8 2、2 8 3 および 2 8 4 を有する 4 × 4 ブロック b 2 8 0 を示す。このプロセスは、ブロックのうちの 4 つのコーナー 2 8 1、2 8 2、2 8 3 および 2 8 4 の各々をチェックすることから成る。(x、y) を高いレイヤの座標系内のコーナーピクセル c の位置とする。(x_{b a s e}、y_{b a s e}) を、次に定義するようなベースレイヤの座標系における対応する位置とする。

【 0 0 8 5 】

【 数 3 】

10

$$\begin{cases} x_{base} = \left[\left[(x - x_{orig}) \cdot w_{base} + w_{extract} / 2 \right] / w_{extract} \right] \\ y_{base} = \left[\left[(y - y_{orig}) \cdot h_{base} + h_{extract} / 2 \right] / h_{extract} \right] \end{cases} \quad (1)$$

【 0 0 8 6 】

次に、ピクセル (x、y) の同じ位置のマクロブロックは、ピクセル (x_{b a s e}、y_{b a s e}) を含むベースレイヤのマクロブロックとなる。同じようにピクセル (x、y) の同じ位置の 8 × 8 ブロックは、ピクセル (x_{b a s e}、y_{b a s e}) を含むベースレイヤの 8 × 8 ブロックとなり、ピクセル (x、y) の同じ位置の 4 × 4 ブロックはピクセル (x_{b a s e}、y_{b a s e}) を含むベースレイヤの 4 × 4 ブロックとなる。

20

【 0 0 8 7 】

b に対する動きデータの引き継ぎプロセスを次のように記述できる。

- 各コーナー c に対し、各リスト l i s t x (l i s t x = l i s t 0 または l i s t 1) のうちの基準インデックス r (c , l i s t x) および動きベクトル m v (c , l i s t x) を同じ位置のベースレイヤ 4 × 4 ブロックのこれらの値にセットする。

- 各コーナーに対し、同じ位置のマクロブロックが存在していないか、またはイントラモードとなっている場合、b をイントラブロックとしてセットする。

30

- そうでない場合、各リスト l i s t x に対し、

○コーナーのいずれもこのリストを使用していない場合、このリストに対する基準インデックスおよび動きベクトルを b にセットしない。

○このリストを使用している場合、

■ b に対してセットした基準インデックス r b (l i s t x) を 4 つのコーナーの現在の基準インデックスの最小値とする。

【 0 0 8 8 】

【 数 4 】

40

$$r_b(listx) = \min_c(r(c, listx)) \quad (2)$$

【 0 0 8 9 】

■ b に対してセットされた動きベクトル m v b (l i s t x) を基準インデックス r b (l i s t x) を有する 4 つのコーナーの現在の動きベクトルの平均とする。

【 0 0 9 0 】

8 × 8 の分割の選択

一旦、各 4 × 4 ブロックの動きデータを設定すると、このデータが属す 8 × 8 ブロックの実際の分割を決定し、かつ禁止された構成を回避するにはマージングプロセスが必要である。下記の記載では、8 × 8 ブロックのうちの 4 × 4 ブロックを、図 2 6 に示されるよ

50

うに識別する。

【0091】

各 8×8 ブロックに対し、次のプロセスを適用できる。

- 4つの 4×4 ブロックがイントラブロックとして分類されている場合、Bをイントラブロックとして見なす。

- そうでない場合、Bの分割の選択を実行する。

○各 4×4 ブロックに対し、同じ基準インデックスを割り当てるために次のプロセスを適用する。

各リスト $listx$ に対し、

■ 4×4 ブロックがこのリストを使用していない場合、このリストのうちの基準インデックスおよび動きベクトルのいずれも、Bにセットしない。 10

■ そうでない場合、

・ Bに対する基準インデックス $r_B(listx)$ を次のように4つの 4×4 ブロックの現在の基準インデックスの最小値として計算する。

【0092】

【数5】

$$r_B(listx) = \min_b(r_b(listx)) \quad (3)$$

【0093】

・ 同じ基準インデックス $r_B(listx)$ を有する 4×4 ブロックの平均動きベクトル $mV_{mean}(listx)$ を計算する。 20

・ (1) イントラブロックとして分類されている 4×4 ブロックまたは (2) このリストを使用していない 4×4 ブロックまたは (3) $r_B(listx)$ とは異なる基準インデックス $r_b(listx)$ を有する 4×4 ブロックが、基準インデックスおよび動きベクトルとして $r_B(listx)$ および $mV_{mean}(listx)$ を有するように強制される。

○次に、Bに対する分割モードの選択を実行する。動きベクトルが同一である場合、2つの 4×4 ブロックを同一と見なす。次のようにマージングプロセスを実行する。

■ b_1 が b_2 と同一であり、 b_3 が b_4 と同一である場合において、 30

・ b_1 が b_3 と同一であれば、BLK__ 8×8 を選択し、

・ そうでない場合、BLK__ 8×4 を選択する。

■ b_1 が b_2 と同一でなく、 b_3 が b_4 と同一でない場合において、 b_1 が b_3 と同一であり、 b_2 が b_4 と同一であれば、BLK__ 4×8 を選択し、

■ そうでない場合、BLK__ 4×4 を選択する。

【0094】

予測マクロブロックモードの選択

一部の実施例では、MB__predモードを決定するためのプロセスを実行できる。次の記載において、マクロブロック300のうちの 8×8 ブロック301～304を、図27に示されるように識別する。 40

次の場合に2つの 8×8 ブロックを同一であると見なす。

- すなわち2つの 8×8 ブロックのうち的一方または双方がイントラブロックとして分類される場合、または

- 双方のブロックの分割モードがBLK__ 8×8 であり、各 8×8 ブロックのうちの $list0$ と $list1$ の基準インデックスおよび動きベクトル（存在する場合）が同一である場合。

【0095】

次のプロセスを使ってモード選択を実行する。

- すべての 8×8 ブロックがイントラブロックとして分類される場合、MB__predをイントラマクロブロックとして分類する。 50

- そうでない場合、 MB_pred をインターマクロブロックとする。このモード選択は次のように実行する。

○イントラとして分類された 8×8 ブロックに対し、強制的に $BLK_8 \times 8$ の分割をする。これら基準インデックスおよび動きベクトルを次のように計算する。 B_INTRA を、かかる 8×8 ブロックとする。

各リスト $listx$ に対し、

■ 8×8 ブロックのいずれも、このリストを使用しない場合、このリストの基準インデックスおよび動きベクトルを B_INTRA に割り当てない。

■ そうでない場合、次のステップを適用する。

・ 基準インデックス $r_{min}(listx)$ を次のように 8×8 ブロックの現在の基準インデックスの最小値として計算する。 10

【0096】

【数6】

$$r_{min}(listx) = \min_B(r_B(listx)) \quad (4)$$

【0097】

・ 同じ基準インデックス $r_{min}(listx)$ を有する 4×4 ブロックの平均動きベクトル $mV_{mean}(listx)$ を計算する。

・ $r_{min}(listx)$ を B_INTRA に割り当て、 B_INTRA の各 4×4 ブロックが強制的に基準インデックスおよび動きベクトルとして $r_{min}(listx)$ および $mV_{mean}(listx)$ を有するようにする。 20

○次に、 B に対する分割モードの選択を実行する。分割モードが $BLK_8 \times 8$ であり、各 8×8 ブロックの $list0$ と $list1$ の基準インデックスおよび動きベクトル(存在する場合)が同一であれば、2つの 8×8 ブロックを同一であると見なす。マーキングプロセスを次のように行う。

■ $B1$ が $B2$ と同一であり、 $B3$ が $B4$ と同一である場合において、

・ $B1$ が $B3$ と同一であれば、 $MODE_16 \times 16$ を選択する。

・ そうでない場合、 $MODE_16 \times 8$ を選択する。

■ $B1$ が $B2$ と同一でなく、 $B3$ が $B4$ と同一でない場合において、 $B1$ が $B3$ と同一であり、 $B2$ が $B4$ と同一である場合、 $MODE_8 \times 16$ を選択する。 30

■ そうでない場合、 $MODE_8 \times 8$ を選択する。

【0098】

動きベクトルのスケーリング

上記のように導いた予測マクロブロック MB_pred の現在のどの動きベクトルに対しても、動きベクトルのデスケーリングを実行できる。次の式を使ってベクトル $v_{ms} = (d_{sx}, d_{sy})$ において、動きベクトル $mv = (d_x, d_y)$ をスケーリングできる。

【0099】

【数7】

40

$$\begin{cases} d_{sx} = \frac{(d_x \cdot w_{extract} + \text{sign}[d_x] \cdot w_{base}/2)}{w_{base}} + 4 \cdot (x_{orig,r} - x_{orig}) \\ d_{sy} = \frac{(d_y \cdot h_{extract} + \text{sign}[d_y] \cdot h_{base}/2)}{h_{base}} + 4 \cdot (y_{orig,r} - y_{orig}) \end{cases} \quad (5)$$

【0100】

ここで、 x が正のとき、 $\text{sign}[x]$ は1に等しく、 x が負のとき(-1)であり、 50

x が 0 のとき 0 となる。添え字「r」を有する記号は、対応する基準ピクチャの幾何学的パラメータを示す。

【0101】

レイヤ間テクスチャの予測

テクスチャのアップサンプリング

本発明の一部の実施例では、レイヤ間動き予測と同じ原理に基づき、レイヤ間テクスチャ予測を行うことができる。2 - l o b e d 又は 3 - l o b e d L a n c z o s - w i n d o w e d s i n c 関数を用いてベースレイヤのテクスチャのアップサンプリングを行うことができる。これらフィルタはエリアシングの低減、シャープさおよび最小リングの点で最良の妥協を行うと見なされる。2 - l o b e d L a n c z o s - w i n d o w e d s i n c 関数は次のように定義される。

【0102】

【数 8】

$$Lanczos\ 2(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} \frac{\sin(\pi \frac{x}{2})}{\pi \frac{x}{2}}, & |x| < 2 \\ 0, & |x| \geq 2 \end{cases} \quad (6)$$

【0103】

このアップサンプリングステップはフルフレームで、またはブロックごとのいずれかで処理できる。イントラテクスチャ予測に対しては、フレーム境界で繰り返しパディングを使用する。残差予測に対しては、ブロック境界（変換に応じて 4×4 または 8×8 ）で繰り返しパディングを使用する。

【0104】

一実施例では、L a n c z o s 2 関数に従い、ベースレイヤピクチャ内の対応する成分のサンプルグリッドに対する 16 分の 1 のサンプルスペースを単位とする 16 の異なる補間フェーズに対し、下記の表 1 に次のような 16 個の 4 タップアップサンプリングフィルタが定められる。

【0105】

位置 (x、y) における現在のレイヤ内のルミナンスサンプルに対し、ベースレイヤピクチャ内の対応するサンプルに関するフェーズシフトが次のように得られる。

【0106】

【数 9】

$$\begin{cases} p_{x,L} = \left[\frac{(x - x_{orig}) \cdot w_{base} \cdot 16}{w_{extract}} \right] - 16 \cdot \left[\frac{(x - x_{orig}) \cdot w_{base}}{w_{extract}} \right] \\ p_{y,L} = \left[\frac{(y - y_{orig}) \cdot h_{base} \cdot 16}{h_{extract}} \right] - 16 \cdot \left[\frac{(y - y_{orig}) \cdot h_{base}}{h_{extract}} \right] \end{cases} \quad (7)$$

【0107】

【表 1】

表1:アップサンプリングに対する4タップの補間フィルタ

フェーズ	(4タップの)補間フィルタ係数			
	e[-1]	e[0]	e[1]	e[2]
0	0	128	0	0
1	-4	127	5	0
2	-8	124	13	-1
3	-10	118	21	-1
4	-11	111	30	-2
5	-11	103	40	-4
6	-10	93	50	-5
7	-9	82	61	-6
8	-8	72	72	-8
9	-6	61	82	-9
10	-5	50	93	-10
11	-4	40	103	-11
12	-2	30	111	-11
13	-1	21	118	-10
14	-1	13	124	-8
15	0	5	127	-4

【 0 1 0 8 】

クロミナンスサンプルの座標系における位置 (x_c 、 y_c) における現在のレイヤ内のクロミナンスサンプルに対し、ベースレイヤピクチャ内の対応するサンプルに関するフェーズシフトは次のように得られる。

【 0 1 0 9 】

【 数 1 0 】

$$\begin{cases} p_{x,c} = \left[\frac{(x_c - x_{orig,c}) \cdot w_{base,c} \cdot 16}{w_{extract,c}} - 16 \cdot \left[\frac{(x_c - x_{orig,c}) \cdot w_{base,c}}{w_{extract,c}} \right] \right] \\ p_{y,c} = \left[\frac{(y_c - y_{orig,c}) \cdot h_{base,c} \cdot 16}{h_{extract,c}} - 16 \cdot \left[\frac{(y_c - y_{orig,c}) \cdot h_{base,c}}{h_{extract,c}} \right] \right] \end{cases} \quad (8)$$

【 0 1 1 0 】

ここで、

【 0 1 1 1 】

【 数 1 1 】

$$w_{base,c} = w_{base} \cdot BasePicMbWidthC / 16 \quad (9)$$

$$w_{extract,c} = w_{extract} \cdot MbWidthC / 16 \quad (10)$$

$$h_{base,c} = h_{base} \cdot BasePicMbHeightC / 16 \quad (11)$$

$$h_{extract,c} = h_{extract} \cdot MbHeightC / 16 \quad (12)$$

$$x_{orig,c} = x_{orig} \cdot MbWidthC / 16 \quad (13)$$

$$y_{orig,c} = y_{orig} \cdot MbHeightC / 16 \quad (14)$$

10

20

30

40

50

【 0 1 1 2 】

である。

得られた各フェーズシフトに従い、補間のために表 1 から 4 タップフィルタを選択できる。

【 0 1 1 3 】

レイヤ間イントラテクスチャ予測

WD-1.0 [MPEG Doc.N6901] において、I __ B L モードは、対応する全てのベースレイヤのマクロブロックをイントラ符号化することを必要とする。本発明の実施例では、要求条件は、対応するベースレイヤのマクロブロックがインター符号化されているかまたは存在しない場合を認めるように、緩和される。

10

【 0 1 1 4 】

I __ B L モードにおいて、符号化されたマクロブロックに対するイントラ予測信号を発生するために、ベースレイヤ信号の同一場所のブロック（存在する場合）を直接デブロックし、補間する。4つの入力サンプル（ $X[n-1]$ 、 $X[n]$ 、 $X[n+1]$ 、 $X[n+2]$ ）に対し、4タップ補間フィルタの出力値 Y は次のように得られる。

【 0 1 1 5 】

【数 1 2】

$$Y = \text{Clip1}_Y((e[-1]X[n-1] + e[0]X[n] + e[1]X[n+1] + e[2]X[n+2] + 64) / 128) \quad (15)$$

の場合

20

$$\text{Clip1}_Y(x) = \min(\max(0, x), (1 \ll \text{BitDepth}_Y) - 1)$$

【 0 1 1 6 】

ここで、 BitDepth_Y はルミナンスサンプルに対するルミナンスチャンネルデータのビット深度を示す。または

【 0 1 1 7 】

【数 1 3】

$$Y = \text{Clip1}_C((e[-1]X[n-1] + e[0]X[n] + e[1]X[n+1] + e[2]X[n+2] + 64) / 128) \quad (16)$$

の場合

30

$$\text{Clip1}_C(x) = \min(\max(0, x), (1 \ll \text{BitDepth}_C) - 1)$$

【 0 1 1 8 】

ここで、 BitDepth_C はクロミナンスサンプルに対するクロミナンスチャンネルデータのビット深度を示す。

【 0 1 1 9 】

式 15 および 16 では、丸め演算を適用しているので、フィルタリングの順序は、まず水平方向、またはまず垂直方向に指定できる。最初に水平方向にフィルタ演算を実行し、次に垂直方向にフィルタ演算を実行することが推奨される。このようなアップサンプリングプロセスは、下記に定義される `extended_spatial_scalability` がイネーブルされるときにしか呼び出されない。

40

【 0 1 2 0 】

このアップサンプリングフィルタ演算後にクロッピングウィンドウの外側の画像領域を満たすのに、一定の値を使用する。この定数は、ルミナンスに対しては $(1 \ll (\text{BitDepth}_Y - 1))$ となりクロミナンスに対しては、 $(1 \ll (\text{BitDepth}_C - 1))$ となる。

【 0 1 2 1 】

レイヤ間残差予測

レイヤ間イントラテクスチャ予測と同様に、ベースレイヤの残差をアップサンプリングするときは、同じ 4 タップフィルタまたは他のフィルタを使用できるが、式 15 および 1

50

6 内の関数とは異なる丸め関数およびクリッピング関数を使用する。

【 0 1 2 2 】

4 つの入力された残差サンプル ($X[n-1]$ 、 $X[n]$ 、 $X[n+1]$ 、 $X[n+2]$) に対して、フィルタの出力値 Y は次のように得られる。ルミナンス残差サンプルに対して、

【 0 1 2 3 】

【 数 1 4 】

$$Y = \text{Clip1}_{Y,r}((e[-1]X[n-1] + e[0]X[n] + e[1]X[n+1] + e[2]X[n+2]) / 128) \quad (17)$$

10

【 0 1 2 4 】

クロミナンス残差サンプルに対しては、

【 0 1 2 5 】

【 数 1 5 】

$$Y = \text{Clip1}_{C,r}((e[-1]X[n-1] + e[0]X[n] + e[1]X[n+1] + e[2]X[n+2]) / 128) \quad (18)$$

【 0 1 2 6 】

残差アップサンプリングに対するクリッピング関数は次のように定義される。

【 0 1 2 7 】

20

【 数 1 6 】

$$\text{Clip1}_{Y,r}(x) = \text{Clip3}(1 - (1 \ll \text{BitDepth}_Y), (1 \ll \text{BitDepth}_Y) - 1, x) \quad (19)$$

$$\text{Clip1}_{C,r}(x) = \text{Clip3}(1 - (1 \ll \text{BitDepth}_C), (1 \ll \text{BitDepth}_C) - 1, x) \quad (20)$$

【 0 1 2 8 】

ここで、 $\text{Clip3}(a, b, x) = \min(\max(a, x), b)$ である。

【 0 1 2 9 】

同様に、アップサンプリングフィルタ演算後、クロッピングウィンドウの外側の画像領域を含む残差予測を利用できないピクセル位置を満たすために、定数値を使用する。この定数はすべてのカラー成分に対して 0 となる。

30

【 0 1 3 0 】

シンタックスおよびセマンティックスの変更

表 (テーブル) 形態のシンタックス

本発明の実施例は、大きい太文字テキストで下記のように示された次の変更を利用できる。主な変更は、記号、すなわち `extended_spatial_scalability` のシーケンスパラメータの組内の追加、従ってシーケンスパラメータセット内の次のような 4 つのパラメータである。

【 0 1 3 1 】

40

【 数 1 7 】

```
scaled_base_left_offset_divided_by_two,
scaled_base_top_offset_divided_by_two,
scaled_base_right_offset_divided_by_two,
scaled_base_bottom_offset_divided_by_two
```

【 0 1 3 2 】

およびベースレイヤのアップサンプリングプロセス内で使用される幾何学的変換に関連する `slice_data_in_scalable_extension()` である。

50

【 0 1 3 3 】

【 表 2 】

スケーラブルな拡張内のシーケンスパラメータセットのシンタックス

seq_parameter_set_rbsp() {	C	デスクリプタ
.....	...	...
extended_spatial_scalability	0	u(2)
If(extended_spatial_scalability ==1) {		
scaled_base_left_offset_divided_by_two	0	ue(v)
scaled_base_top_offset_divided_by_two	0	ue(v)
scaled_base_right_offset_divided_by_two	0	ue(v)
scaled_base_bottom_offset_divided_by_two	0	ue(v)
}		
.....	...	...
rbbsp_trailing_bits()	0	
}		

10

【 0 1 3 4 】

【 表 3 】

スケーラブルな拡張内のスライスデータシンタックス

slice_data_in_scalable_extension() {	C	デスクリプタ
If(extended_spatial_scalability ==2) {		
scaled_base_left_offset_divided_by_two	2	ue(v)
scaled_base_top_offset_divided_by_two	2	ue(v)
scaled_base_right_offset_divided_by_two	2	ue(v)
scaled_base_bottom_offset_divided_by_two	2	ue(v)
}		
If(extended_spatial_scalability)		
HalfSpatResBaseFlag = 0		
else		
HalfSpatResBaseFlag = half_spat_res_base_pic()		
.....	...	...
}		

30

【 0 1 3 5 】

40

【表 4】

スケラブルな拡張内のマクロブロック層のシンタックス

macroblock_layer_in_scalable_extension() {	C	デスクリプタ
If(base_id_plus1 != 0 && adaptive_prediction_flag) {		
base_mode_flag	2	ae(v)
If(! base_mode_flag && (HalfSpatResBaseFlag extended_spatial_scalability) && ! intra_base_mb(CurrMbAddr))		
base_mode_refinement_flag	2	ae(v)
}		
.....	...	...
}		

10

【0136】

セマンティックス

スケラブルな拡張内のシーケンスパラメータセットのシンタックス

`extended_spatial_scalability` は、ベースレイヤのアップサンプリングのための幾何学的パラメータに関連するシンタックス要素の存在を指定する。`extended_spatial_scalability` が 0 に等しいとき、ビットストリーム内には幾何学的パラメータは存在しない。`extended_spatial_scalability` が 1 に等しいとき、シーケンスパラメータセット内に幾何学的パラメータが存在する。`extended_spatial_scalability` が 2 のとき、`slice_data_in_scalable_extension` 内に幾何学的パラメータが存在する。`extended_spatial_scalability` に対し、3 の値を保留する。`extended_spatial_scalability` が存在しないとき、0 に等しいと推定する。

20

【0137】

`scaled_base_left_offset_divided_by_two` は、アップサンプリングされたベースレイヤピクチャの上部左ピクセルと現在のピクチャの上部左ピクセルとの間の水平オフセットの半分を指定する。この `scaled_base_left_offset_divided_by_two` が存在しないとき、これを 0 に等しいと推定する。

30

【0138】

`scaled_base_top_offset_divided_by_two` は、アップサンプリングされたベースレイヤピクチャの上部左ピクセルと現在のピクチャの上部左ピクセルとの間の垂直オフセットの半分を指定する。この `scaled_base_top_offset_divided_by_two` が存在しないとき、これを 0 に等しいと推定する。

40

【0139】

`scaled_base_right_offset_divided_by_two` は、アップサンプリングされたベースレイヤピクチャの底部右ピクセルと現在のピクチャの底部右ピクセルとの間の水平オフセットの半分を指定する。この `scaled_base_right_offset_divided_by_two` が存在しないとき、これを 0 に等しいと推定する。

【0140】

`scaled_base_bottom_offset_divided_by_two` は、アップサンプリングされたベースレイヤピクチャの底部右ピクセルと現在のピクチャ

50

ャの底部右ピクセルとの間の垂直オフセットの半分を指定する。この `scaled_base_se__bottom_offset_divided_by_two` が存在しないとき、これを 0 に等しいと推定する。

【0141】

現在のレイヤ内のルミナンスサンプリンググリッドに関する 1 つのサンプルのスペースを単位とする符号のない整数として、すべての幾何学的パラメータを指定する。下記の幾何学的パラメータに基づき、いくつかの追加記号 (`scaled_base_left_offset`、`scaled_base_top_offset`、`scaled_base_right_offset`、`scaled_base_bottom_offset`、`scaled_base_width`、`scaled_base_height`) を定義する。

10

【0142】

【数18】

$$\begin{aligned} scaled_base_left_offset &= 2 \cdot scaled_base_left_offset_divided_by_two \\ scaled_base_top_offset &= 2 \cdot scaled_base_top_offset_divided_by_two \\ scaled_base_right_offset &= 2 \cdot scaled_base_right_offset_divided_by_two \\ scaled_base_bottom_offset &= 2 \cdot scaled_base_bottom_offset_divided_by_two \\ scaled_base_width &= PicWidthInMbs \cdot 16 - scaled_base_left_offset - \\ &\quad scaled_base_right_offset \\ scaled_base_height &= PicHeightInMapUnits \cdot 16 - scaled_base_top_offset - \\ &\quad scaled_base_bottom_offset \end{aligned}$$

20

【0143】

スケーラブルな拡張内のスライスデータシンタックス

スライスデータ内のシンタックス要素のセマンティックスは、シーケンスパラメータセット内の同じシンタックス要素のセマンティックスと同一である。

【0144】

復号プロセス

30

予測データに対する復号プロセス

WD-1.0 [MPEG Doc.N6901] と比較すると、次のプロセスを追加しなければならない。各マクロブロックに対し、次のことが当てはまる。

- `extended_spatial_scalability` が 1 または 2 に等しく、`base_layer_mode_flag` が 1 に等しい場合、第 3 章に記載されているプロセスを使ってマクロブロック分割を含む動きベクトルフィールドを得る。WD-1.0 [MPEG Doc.N6901] の場合のように、全ての対応するベースレイヤのマクロブロックがイントラ符号化されている場合、現在のマクロブロックのモードを `I_BL` にセットする。

- そうでなく、`extended_spatial_scalability` が 1 または 2 に等しく、`base_layer_mode_flag` が 0 であり、`base_layer_refinement` が 1 に等しい場合、ベースレイヤ改善モードの信号を発する。このベースレイヤ改善モードはベースレイヤ予測モードに類似している。次の第 3 章では、マクロブロック分割だけでなく、基準インデックスおよび動きベクトルを得る。しかしながら、各動きベクトルに対し、更に 4 分の 1 サンプルの動きベクトル改善 `mv_d_ref_1X` (各動きベクトル成分に対し、-1、0 または +1) を送信し、得られた動きベクトルに加える。

40

プロセスの残りは、WD-1.0 [MPEG Doc.N6901] に記載されているプロセスと同一である。

【0145】

サブバンドピクチャに対する復号プロセス

50

WD-1.0 [MPEG Doc.N6901] と比較すると、次のプロセスを追加しなければならない。

`extended__spatial__scalability` が 1 または 2 に等しい場合、`I__BL` モードの `MB` に対するイントラ予測信号を次のプロセスにより発生する。

- 並べられたベースレイヤブロック / マクロブロックをフィルタにかける。

- デブロックされたデータを補間することにより、イントラ予測信号を発生する。第 4 章に記載されたプロセスを使ってこの補間を実行する。プロセスの他の部分は WD-1.0 [MPEG Doc.N6901] に記載されているプロセスと同一である。

そうでなく、`extended__spatial__scalability` が 1 または 2 に等しく、`residual__prediction__flag` が 1 に等しい場合、次のことが当てはまる。

- ベースレイヤのブロックの残差信号をアップサンプリングし、現在のマクロブロックの残差信号に追加する。第 4 章に記載のプロセスを使って補間を実行する。

【 0 1 4 6 】

ループフィルタに対する変更

`extended__spatial__scalability` が 1 または 2 に等しいとき、`I__BL` モード（レイヤ間テクスチャ予測）のブロックに対し、フィルタ強度判断をする間にループフィルタにわずかな変更を行わなければならない。

- 隣接ブロックがイントラ符号化されているが、`I__BL` モードにない場合、`Bs` は 4 となる（この最初の部分は WD-1.0 [MPEG Doc.N6901] と同一である）。

- そうでなく、隣接ブロックのいずれかが係数を有する場合、`Bs` は 2 となる。

- そうでなく、隣接ブロックが `I__BL` モードにない場合、`Bs` は 1 となる。

- そうでない場合、`Bs` は 0 となる。

【 0 1 4 7 】

6 タップフィルタの実施例

本発明の一部の実施例は、特に J V T（MPEG と VCEG の共同ビデオチーム）により、2005 年 4 月に採用された、拡張空間的スケーラブル（ESS）ビデオ符号化機能のための、H.264/MPEG-4 AVC のスケーラブルビデオ符号化拡張と共に使用されている。

【 0 1 4 8 】

現在の SVC 設計では、アップサンプリングプロセスはインター予測のための H.264 に指定された 4 分の 1 ルミナンスサンプル補間手順に基づく。この方法は、空間的にスケーラブルな符号化に適用したとき、2 つの固有の欠点を有する。すなわち（1）補間分解能が 4 分の 1 サンプルに限定されること、および（2）4 分の 1 サンプルの位置を得るには半分のサンプル補間を実行しなければならないことである。

【 0 1 4 9 】

本発明の一部の実施例は、（1）より細かい補間分解能および（2）直接的補間により、これら欠点を除いている。その結果、これら実施例はアップサンプリングしたピクチャの画質を改善しながら、計算上の複雑さを低下させている。

【 0 1 5 0 】

本発明の実施例のアップサンプリング技術は、16 個の 6 タップフィルタを用いた直接補間に基づく。フィルタの選択は 16 分の 1 のピクチャサンプルを単位とする 0 ~ 15 の範囲の補間位置またはフェーズに従っている。フィルタの組は、SVC のハーフサンプル補間プロセスおよび H.264 のハーフサンプルルミナンスインター予測とバックワードコンバーチブルとなるように設計されている。従って、これら実施例の技術は、ハードウェア / ソフトウェアの実現の観点から、H.264 の自然の拡張とすることができる。

【 0 1 5 1 】

従来の空間的スケーラブルなビデオ符号化システムは、一般に空間または解像度スケーリングファクタが 2 または 2 の累乗であるケースを取り扱う。2005 年 4 月に、空間スケーリングファクタが 2 の累乗に限定されない、より包括的なアプリケーションを取り扱うよう、SVC 共同スケーラブルビデオモデル（JSVM）に拡張空間的スケーラビリティが採用された。しかしながら、レイヤ間テクスチャ予測のためのアップサンプリング方法は、

10

20

30

40

50

まだ開発中の課題である。2005年4月のJVTの会議中、テクスチャアップサンプリングのためにH.264において指定された4分の1ルミナンスサンプル補間プロセスを暫定的に採用するとの決定がなされた。

【0152】

本発明のこれら実施例では、図23を参照してこれまでの実施例に対して説明した同じ幾何学的関係を同じように当てはめる。

【0153】

これまで説明した実施例では、ベースレイヤピクチャ内の対応する成分の整数サンプルグリッドに関する16分の1のサンプルスペースを単位とする16の異なるフェーズの補間に対し、16個の4タップアップサンプリングフィルタのセットを定めた。しかしながら、これら4タップフィルタは、初期のH.264設計に対してバックワードコンパチブルではない。従って、これら実施例は16個の6タップフィルタの新しいセットおよび対応するフィルタリング手順を含むことができる。一実施例では、表5に記載した6タップフィルタを使用でき、別の実施例では、表6に記載された6タップフィルタを使用できる。

10

【0154】

【表5】

表2:16フェーズの補間フィルタの第1の例

フェーズ	(6タップの)補間フィルタ係数					
	e[-2]	e[-1]	e[0]	e[1]	e[2]	e[3]
0	0	0	32	0	0	0
1	0	-2	32	2	0	0
2	1	-3	31	4	-1	0
3	1	-4	30	7	-2	0
4	1	-4	28	9	-2	0
5	1	-5	27	11	-3	1
6	1	-5	25	14	-3	0
7	1	-5	22	17	-4	1
8	1	-5	20	20	-5	1
9	1	-4	17	22	-5	1
10	0	-3	14	25	-5	1
11	1	-3	11	27	-5	1
12	0	-2	9	28	-4	1
13	0	-2	7	30	-4	1
14	0	-1	4	31	-3	1
15	0	0	2	32	-2	0

20

30

【0155】

【表 6】

表3:16フェーズの補間フィルタの第2の例

フェーズ	(6タップの)補間フィルタ係数					
	e[-2]	e[-1]	e[0]	e[1]	e[2]	e[3]
0	0	0	32	0	0	0
1	0	-2	32	2	0	0
2	1	-3	31	4	-1	0
3	1	-4	30	6	-1	0
4	1	-4	28	9	-2	0
5	1	-4	27	11	-3	0
6	1	-5	25	14	-3	0
7	1	-5	22	17	-4	1
8	1	-5	20	20	-5	1
9	1	-4	17	22	-5	1
10	0	-3	14	25	-5	1
11	0	-3	11	27	-4	1
12	0	-2	9	28	-4	1
13	0	-1	6	30	-4	1
14	0	-1	4	31	-3	1
15	0	0	2	32	-2	0

【0156】

整数のルミナンスサンプルを単位とするエンハンスメントピクチャ内のルミナンスサンプル位置 (x、y) を仮定すると、次のようにベースピクチャの16分の1ルミナンスサンプルを単位とするベースピクチャ内の対応する位置 (p_x、_L、p_y、_L) を導くことができる。

【0157】

【数19】

$$\begin{cases} p_{x,L}(x) = \left[(x - x_{orig}) \cdot w_{base} \cdot R_L + \frac{R_L}{2} (w_{base} - w_{extract}) \right] // w_{extract} \\ p_{y,L}(y) = \left[(y - y_{orig}) \cdot h_{base} \cdot R_L + \frac{R_L}{2} (h_{base} - h_{extract}) \right] // h_{extract} \end{cases} \quad (21)$$

【0158】

ここで、R_L = 16 (16分の1サンプルの解像度の補間に対して) であり、図23のように (x_{orig}、y_{orig}) は、現在のピクチャの単一ルミナンスサンプルを単位とする現在のピクチャ内のクロッピングウィンドウの上部左コーナーの位置を示し、(w_{base}、h_{base}) は、ベースピクチャの単一ルミナンスサンプルを単位とするベースピクチャの解像度であり、(w_{extract}、h_{extract}) は現在のピクチャの単一ルミナンスサンプルを単位とするクロッピングウィンドウの解像度であり、「//」は簡略化された割り算演算子を意味する。

【0159】

同様に、単一のクロミナンスサンプルを単位とするエンハンスメントピクチャ内のクロミナンスサンプル位置 (x_c、y_c) を仮定すると、ベースピクチャの16分の1クロミナンスサンプルを単位とするベースピクチャ内の対応する位置 (p_x、_c、p_y、_c) は次のように導かれる。

【0160】

10

20

30

40

50

【数 2 0】

$$\begin{cases} p_{x,c}(x_c) = \left[(x_c - x_{orig,c}) \cdot w_{base,c} \cdot R_C + \frac{R_C}{4} (2 + p_{enh,x}) w_{base,c} - \frac{R_C}{4} (2 + p_{base,x}) w_{extract,c} \right] // w_{extract,c} \\ p_{y,c}(y_c) = \left[(y_c - y_{orig,c}) \cdot h_{base,c} \cdot R_C + \frac{R_C}{4} (2 + p_{enh,y}) h_{base,c} - \frac{R_C}{4} (2 + p_{base,y}) h_{extract,c} \right] // h_{extract,c} \end{cases} \quad (22)$$

【0161】

ここで、 $R_C = 16$ であり、 $(x_{orig,c}, y_{orig,c})$ は、現在のピクチャの単クロミナンスサンプルを単位とする現在のピクチャ内のクロッピングウィンドウの上部左コーナーの位置を示し、 $(w_{base,c}, h_{base,c})$ は、ベースピクチャの単クロミナンスサンプルを単位とするベースピクチャの解像度であり、 $(w_{extract,c}, h_{extract,c})$ は現在のピクチャの単クロミナンスサンプルを単位とするクロッピングウィンドウの解像度であり、 $(p_{base,x}, p_{base,y})$ はベースピクチャの4分の1クロミナンスサンプルを単位とするベースピクチャの相対的クロミナンスフェーズシフトを示し、 $(p_{enh,x}, p_{enh,y})$ は現在のピクチャの4分の1クロミナンスサンプルを単位とする現在のピクチャの相対的クロミナンスフェーズシフトを示す。

【0162】

式21および22から導かれる補間位置に基づき、表5または表6から6タップフィルタを選択できる。一部の実施例では、補間位置がハーフサンプル位置であるとき、フィルタはハーフサンプル補間のために定められたH.264内のフィルタと同一である。従って、本発明のこれら実施例の技術に対して同様のハードウェア/ソフトウェアモジュールを使用できる。

【0163】

しかしながら、レイヤ間残差アップサンプリングに対しテクスチャアップサンプリングのための6タップフィルタまたは上記4タップフィルタの代わりに、バイリニア補間フィルタと共に同様の直接補間方法を使用できる。

【0164】

一部の実施例では、補間プロセスは次のとおりである。

【0165】

1. エンハンスメントピクチャ内のマクロブロックのうちの上部左ルミナンスサンプルに対する位置 (x_P, y_P) を定める。 $chroma_format_idc$ が0に等しくないとき、すなわちクロミナンスチャンネルが存在するとき、同一マクロブロックのうちの上部左クロミナンスサンプルに対する位置 (x_C, y_C) を定める。

【0166】

2. 次のようにベースレイヤピクチャ内のマクロブロックの相対的位置を導く。

【0167】

【数 2 1】

$$\begin{cases} xB = p_{x,L}(x_P) >> 4 \\ yB = p_{y,L}(y_P) >> 4 \end{cases} \quad (23)$$

$$\begin{cases} xB1 = (p_{x,L}(x_P + 15) + 15) >> 4 \\ yB1 = (p_{y,L}(y_P + 15) + 15) >> 4 \end{cases} \quad (24)$$

【0168】

$chroma_format_idc$ が0に等しくないとき、次のようになる。

【0169】

10

20

30

40

50

【数 2 2】

$$\begin{cases} xCB = p_{x,C}(xC) >> 4 \\ yCB = p_{y,C}(yC) >> 4 \end{cases} \quad (25)$$

$$\begin{cases} xCB1 = (p_{x,C}(xC + MbWidthC - 1) + 15) >> 4 \\ yCB1 = (p_{y,C}(yC + MbHeightC - 1) + 15) >> 4 \end{cases} \quad (26)$$

【0 1 7 0】

ここで、MbWidthCおよびMbHeightCは水平方向および垂直方向のMB
当たりのクロミナスサンプルの数をそれぞれ示す。 10

【0 1 7 1】

3. テクスチャ補間プロセス

このプロセスへの入力は次のものを含む。

- ・ベースピクチャ内の整数ルミナスサンプル位置 (x B、y B) および (x B 1、y B 1)
- ・ $x = -2 + xB \dots (xB1 + 2)$ および $y = -2 + yB \dots (yB1 + 2)$ の場合のベースピクチャに対するルミナスサンプルアレイ $base_L[x, y]$
- ・chroma__format__idc が 0 に等しくないとき、
 - ベースピクチャ内の整数クロミナスサンプル位置 (xCB、yCB) および (xCB 1、yCB 1)
 - $x = -2 + xCB \dots (xCB1 + 2)$ および $y = -2 + yCB \dots (yCB1 + 2)$ の場合のベースピクチャに対する 2 つのクロミナスサンプルアレイ $base_{cb}[x, y]$ および $base_{cr}[x, y]$

【0 1 7 2】

このプロセスの出力は次のものを含む。

- ・ $x = 0 \dots 15$ および $y = 0 \dots 15$ の場合の、ルミナスサンプルマクロブロックアレイ $pred_L[x, y]$
- ・chroma__format__idc が 0 でない場合の、 $x = 0 \dots MbWidthC - 1$ 、および $y = 0 \dots MbHeightC - 1$ の場合の、2 つのクロミナスサンプルマクロブロックアレイ $pred_{cb}[x, y]$ および $pred_{cr}[x, y]$

【0 1 7 3】

$x = 0 \dots 15$ および $y = 0 \dots 15$ の場合の、ルミナスサンプル $pred_L[x, y]$ は次のように導かれる。

- ・ $x = -2 + xB \dots (xB1 + 2)$ および $y = 0 \dots 15$ の場合の、 $temp_L[x, y]$ を一時的ルミナスサンプルアレイとする。
- ・ $x = -2 + xB \dots (xB1 + 2)$ および $y = 0 \dots 15$ の場合の、各 $temp_L[x, y]$ は次のように導かれる。

○ベースレイヤ内の対応する部分的サンプル位置 y f は次のように導かれる。

$$yf = P_{y, L}(y + yP)$$

○yInt および yFrac を次のように定義する。

【0 1 7 4】

【数 2 3】

$$yInt = (yf >> 4)$$

$$yFrac = yf \% 16$$

【0 1 7 5】

○yFrac をフェーズとして使用し、表 5 から $j = -2 \dots 3$ の場合の 6 タップフィルタ $e[j]$ を選択し、 $temp_L[x, y]$ は次のように導かれる。

【0 1 7 6】

【数 2 4】

$$\begin{aligned} \text{temp}_L[x,y] = & \text{base}_L[x,y\text{Int}-2]*e[-2] + \text{base}_L[x,y\text{Int}-1]*e[-1] + \\ & \text{base}_L[x,y\text{Int}]*e[0] + \\ & \text{base}_L[x,y\text{Int}+1]*e[1] + \text{base}_L[x,y\text{Int}+2]*e[2] + \\ & \text{base}_L[x,y\text{Int}+3]*e[3] \end{aligned}$$

【0 1 7 7】

・ $x = 0 \dots 15$ および $y = 0 \dots 15$ の場合の各サンプル $\text{pred}_L[x, y]$ は次のように導かれる。

10

○ ベースレイヤ内の対応する部分的サンプル位置 x_f は次のように導かれる。

$$x_f = P_{x, L}(x + x_P)$$

○ $x\text{Int}$ および $x\text{Frac}$ は次のように導かれる。

【0 1 7 8】

【数 2 5】

$$\begin{aligned} x\text{Int} &= (x_f \gg 4) \\ x\text{Frac} &= x_f \% 16 \end{aligned}$$

【0 1 7 9】

○ $x\text{Frac}$ をフェーズとして使用し、表 5 から $j = -2 \dots 3$ の場合の 6 タップフィルタ $e[j]$ を選択し、 $\text{pred}_L[x, y]$ は次のように導かれる。

20

【0 1 8 0】

【数 2 6】

$$\begin{aligned} \text{pred}_L[x,y] = & \text{Clip1}_Y((\text{temp}_L[x\text{Int}-2,y]*e[-2] + \text{temp}_L[x\text{Int}-1,y]*e[1] + \\ & \text{temp}_L[x\text{Int},y]*e[0] + \\ & \text{temp}_L[x\text{Int}+1,y]*e[1] + \text{temp}_L[x\text{Int}+2,y]*e[2] + \\ & \text{temp}_L[x\text{Int}+3,y]*e[3] + 512)/1024) \end{aligned}$$

【0 1 8 1】

$\text{chroma_format_id}_c$ が 0 でないとき、 $x = 0 \dots \text{MbWidth}_C - 1$ 、 $y = 0 \dots \text{MbHeight}_C - 1$ の場合のクロミナンスサンプル $\text{pred}_c[x, y]$ (ここで、 C は Cb または Cr である) は次のように導かれる。

30

・ $x = -2 + x_{CB} \dots (x_{CB} + 2)$ および $y = 0 \dots \text{MbHeight}_C - 1$ の場合の $\text{temp}_{L_{Cb}}[x, y]$ および $\text{temp}_{L_{Cr}}[x, y]$ を一時的クロミナンスサンプルアレイとする。

・ C が Cb および Cr であり、 $x = -2 + x_{CB} \dots (x_{CB} + 2)$ および $y = 0 \dots \text{MbHeight}_C - 1$ である場合の各 $\text{temp}_C[x, y]$ は次のように導かれる。

○ ベースレイヤ内の対応する部分サンプル位置 y_{fC} は次のように導かれる。

$$y_{fC} = p_{y, c}(y + y_C)$$

○ $y\text{Int}_C$ および $y\text{Frac}_C$ を次のように定める。

40

【0 1 8 2】

【数 2 7】

$$\begin{aligned} y\text{Int}_C &= (y_{fC} \gg 4) \\ y\text{Frac}_C &= y_{fC} \% 16 \end{aligned}$$

【0 1 8 3】

○ $y\text{Frac}_C$ をフェーズとして使用し、表 5 から $j = -2 \dots 3$ の場合の 6 タップフィルタ $e[j]$ を選択し、 $\text{temp}_c[x, y]$ は次のように導かれる。

【0 1 8 4】

【数 2 8】

$$\begin{aligned} \text{tempC}[x,y] = & \text{baseC}[x,\text{yIntC}-2]*e[-2] + \text{baseC}[x,\text{yIntC}-1]*e[-1] + \\ & \text{baseC}[x,\text{yIntC}]*e[0] + \\ & \text{baseC}[x,\text{yIntC}+1]*e[1] + \text{baseC}[x,\text{yIntC}+2]*e[2] + \\ & \text{baseC}[x,\text{yIntC}+3]*e[3] \end{aligned}$$

【0 1 8 5】

・C が C b および C r であり、 $x = 0 \dots \text{MbWidthC} - 1$ および $y = 0 \dots \text{MbHeightC} - 1$ の場合の各サンプル $\text{predC}[x, y]$ は次のように導かれる。

○ベースレイヤ内の対応する部分的サンプル位置 xfC は次のように導かれる。

10

$$\text{xfC} = \text{p}_x, \text{c}(\text{x} + \text{x}_\text{C})$$

○ xIntC および xFracC を次のように定義する。

【0 1 8 6】

【数 2 9】

$$\begin{aligned} \text{xIntC} &= (\text{xfC} \gg 4) \\ \text{xFracC} &= \text{xfC} \% 16 \end{aligned}$$

【0 1 8 7】

○ xFracC をフェーズとして使用し、表 5 から $j = -2 \dots 3$ の場合の 6 タップフィルタ $e[j]$ を選択し、 $\text{predC}[x, y]$ は次のように導かれる。

20

【0 1 8 8】

【数 3 0】

$$\begin{aligned} \text{predC}[x,y] = & \text{Clip1C}((\text{tempC}[\text{xIntC}-2,y]*e[-2] + \text{tempC}[\text{xIntC}-1,y]*e[-1] + \\ & \text{tempC}[\text{xIntC},y]*e[0] + \text{tempC}[\text{xIntC}+1,y]*e[1] + \text{tempC}[\text{xIntC}+2,y]*e[2] + \\ & \text{tempC}[\text{xIntC}+3,y]*e[3] + 512)/1024) \end{aligned}$$

【0 1 8 9】

4. テクスチャ補間プロセス

このプロセスへの入力は次のものを含む。

30

・ $\text{basePic}(x_B, y_B)$ および (x_{B1}, y_{B1}) における整数ルミナンスサンプル位置

・ $x = -x_B \dots x_{B1}$ および $y = y_B \dots y_{B1}$ の場合のルミナンス残差サンプルアレイ $\text{resBaseL}[x, y]$

・ chroma_format_idC が 0 に等しくないとき、

○ $\text{basePic}(x_{CB}, y_{CB})$ および (x_{CB1}, y_{CB1}) における整数クロミナンスサンプル位置

○ $x = x_{CB} \dots x_{CB1}$ および $y = y_{CB} \dots y_{CB1}$ の場合の 2 つのクロミナンス残差サンプルアレイ $\text{resBaseCb}[x, y]$ および $\text{resBaseCr}[x, y]$

【0 1 9 0】

40

このプロセスの出力は次のものを含む。

・ $x = 0 \dots 15$ および $y = 0 \dots 15$ の場合のルミナンスサンプルアレイ $\text{resPredL}[x, y]$

・ chroma_format_idC が 0 に等しくないとき、 $x = 0 \dots \text{MbWidthC} - 1$ および $y = 0 \dots \text{MbHeightC} - 1$ の場合の 2 つのクロミナンスサンプルアレイ $\text{resPredCb}[x, y]$ および $\text{resPredCr}[x, y]$

【0 1 9 1】

$x = 0 \dots 15$ および $y = 0 \dots 15$ の場合のルミナンス残差サンプル $\text{resPredL}[x, y]$ は次のように導かれる。

・ $x = x_B \dots x_{B1}$ および $y = 0 \dots 15$ の場合の $\text{tempL}[x, y]$ を一時的ルミナ

50

ンスサンプルアレイとする。

・ $x = -x_B \dots x_B - 1$ および $y = 0 \dots 15$ の場合の $temp_L[x, y]$ は次のように導かれる。

○ ベースレイヤ内の対応する部分的サンプル位置 y_f は次のように導かれる。

$$y_f = P_{y, L}(y + y_P)$$

○ y_{Int} および y_{Frac} を次のように定める。

【0192】

【数31】

$$\begin{aligned} y_{Int} &= (y_f \gg 4) \\ y_{Frac} &= y_f \% 16 \end{aligned}$$

10

【0193】

○ $temp_L[x, y]$ は次のように導かれる。

【0194】

【数32】

$$temp_L[x, y] = resBase_L[x, y_{Int}] * (16 - y_{Frac}) + resBase_L[x, y_{Int} + 1] * y_{Frac}$$

【0195】

・ $x = 0 \dots 15$ および $y = 0 \dots 15$ の場合の各残差サンプル $resPred_L[x, y]$ は次のように導かれる。 20

○ ベースレイヤ内の対応する部分的サンプル位置 x_f は次のように導かれる。

$$x_f = p_{x, L}(x + x_P)$$

○ x_{Int} および x_{Frac} を次のように定める。

【0196】

【数33】

$$\begin{aligned} x_{Int} &= (x_f \gg 4) \\ x_{Frac} &= x_f \% 16 \end{aligned}$$

【0197】

30

○ $resPred_L[x, y]$ は次のように導かれる。

【0198】

【数34】

$$resPred_L[x, y] = Clip1_{Y_r}((temp_L[x_{Int}, y] * (16 - x_{Frac}) + temp_L[x_{Int} + 1, y] * x_{Frac}) / 256)$$

の場合

$$Clip1_{Y_r}(x) = Clip3(1 - (1 \ll BitDepth_Y), (1 \ll BitDepth_Y) - 1, x)$$

40

【0199】

ここで、 $BitDepth_Y$ はルミナンスチャンネルデータのビット深さを示す。

【0200】

$chroma_format_idc$ が 0 に等しくないとき、 $x = 0 \dots MbWidthC - 1$ 、 $y = 0 \dots MbHeightC - 1$ の場合のクロミナンス残差サンプル $resPred_c[x, y]$ (ここで、 C は Cb または Cr である) は次のように導かれる。

・ $x = x_{CB} \dots x_{CB} - 1$ および $y = 0 \dots MbHeightC - 1$ の場合の $temp_{l_{Cb}}[x, y]$ および $temp_{l_{Cr}}[x, y]$ を一時的クロミナンスサンプルアレイとする。

・ C が Cb および Cr であり、 $x = -x_{CB} \dots x_{CB} - 1$ および $y = 0 \dots MbHeightC - 1$ の場合の $temp_{l_{Cb}}[x, y]$ および $temp_{l_{Cr}}[x, y]$ は次のように導かれる。 50

$tC - 1$ である場合の各 $temp_c[x, y]$ は次のように導かれる。

○ベースレイヤ内の対応する部分サンプル位置 yfC は次のように導かれる。

$$yfC = p_{y, c}(y + yC)$$

○ $yIntC$ および $yFracC$ を次のように定める。

【0201】

【数35】

$$\begin{aligned} yIntC &= (yfC \gg 4) \\ yFracC &= yfC \% 16 \end{aligned}$$

【0202】

10

○ $temp_c[x, y]$ を次のように導く。

【0203】

【数36】

$$\begin{aligned} temp_c[x, y] &= resBase_c[x, yIntC] * (16 - yFracC) + resBase_c[x, yIntC \\ &+ 1] * yFracC \end{aligned}$$

【0204】

・ C が Cb および Cr であり、 $x = 0 \dots MbWidthC - 1$ および $y = 0 \dots MbHeightC - 1$ の場合の各サンプル $resPred_c[x, y]$ は次のように導かれる。

○ベースレイヤ内の対応する部分的サンプル位置 xfC は次のように導かれる。

20

$$xfC = p_{x, c}(x + xC)$$

○ $xIntC$ および $xFracC$ を次のように定義する。

【0205】

【数37】

$$\begin{aligned} xIntC &= (xfC \gg 4) \\ xFracC &= xfC \% 16 \end{aligned}$$

【0206】

○ $resPred_c[x, y]$ は次のように導かれる。

【0207】

30

【数38】

$$\begin{aligned} resPred_c[x, y] &= Clip1_{C,r}((temp_c[xIntC, y] * (16 - xFracC) + temp_c[xIntC + \\ &1, y] * xFracC) / 256) \end{aligned}$$

の場合

$$Clip1_{C,r}(x) = Clip3(1 - (1 \ll BitDepth_C), (1 \ll BitDepth_C) - 1, x)$$

【0208】

ここで、 $BitDepth_C$ は、クロミナンスチャンネルデータのビット深さを示す。

40

【0209】

本発明の一部の実施例は、空間的スケーラブルビデオ符号化のためのデブロッキングフィルタを備える。これら実施例のうちの一部の実施例におけるフィルタリング方法は、特に JVT (MPEGとVCEGの共同ビデオチーム) により、2005年4月に採用された、拡張空間的スケーラブル (ESS) ビデオ符号化機能のための、H.264/MPEG-4 AVCのスケーラブルビデオ符号化拡張のために設計されている。

【0210】

従来の方法では、フィルタリングプロセスは可能な種々の空間的解像度を有する全てのレイヤにわたって同一であった。レイヤ間テクスチャ予測を利用して符号化されるブロックはフィルタリングプロセス中にイントラ符号化されたブロックと見なされていた。この

50

ような従来の方法は、空間的スケーラブル符号化に適用したときに2つの欠点を有する。すなわち(1)低い解像度のレイヤからの予測は不必要にボケることがあり、従って、(2)プロセスは不必要により多くの計算サイクルを費やす。

【0211】

本発明の実施例は、一部のブロック境界に対するフィルタ演算をスキップし、異なるブロック境界に対して異なるフィルタを使用し、異なるブロック境界におけるフィルタのグレースhiftを変え、または特定のブロック境界に対してフィルタ特性を調整することにより、これら欠点の双方を除去できる。この結果、これら実施例は計算上の複雑さを低減し、アップサンプリングされた画像の質を改善できる。

【0212】

これら実施例では、インターブロックのための現存のAVC設計におけるフィルタリングの決定が適用されるように、レイヤ間テクスチャ予測を使用して符号化されたブロックをインターブロックと見なす。一部の実施例では、非空間的にスケーラブル符号化と共に、隣接するブロックに対してこれまで説明した適応型ブロック境界フィルタリングを空間的スケーラブル符号化に適用される。H.264に採用されたこれら方法は、空間的スケーラブルビデオ符号化に適用される。

【0213】

本発明の一部の実施例では、画像ブロック境界のためのデブロッキングフィルタは、0~4の範囲内または他のある範囲内の値を有することができる制御パラメータ境界強度(B_s)によって特徴づけられる。 B_s の値が大きくなればなるほど、対応する境界に使用されるフィルタ演算も強くなる。 B_s が0に等しいとき、フィルタ演算をスキップするか、または最小にできる。

【0214】

現在のSVC設計では、レイヤ間テクスチャ予測に基づくマクロブロック予測モードをI__BLモードと称す。従来の方法を使用すると、I__BLマクロブロックに関連する全てのブロック境界をフィルタリングしなければならない。すなわち全てのブロック境界に対し、 $B_s > 0$ となる。

【0215】

本発明の実施例は、SVC内の記号Spatial Scalability Typeが0に等しくないときに、空間的スケーラブル符号化のためのI__BLモードのブロックのためのフィルタ強度判断方法を含む。この目的は、計算上の複雑さを低減し、ベースレイヤからの予測のボケを防止することにある。

【0216】

一部の実施例では、I__BLモードでのブロックに対し、ブロックと隣接するブロックとの間の境界の B_s を次のように導くことができる。

1. 隣接するブロックがイントラ符号化されており、I__BLモードでない場合、 B_s は4である。
2. そうでなくて、隣接するブロックのいずれかが0でない係数を有する場合、 B_s は2となる。
3. そうでなくて、隣接するブロックが同一ベースレイヤのピクチャに基づき、I__BLモードでない場合、 B_s は1となる。
4. その他の場合、 B_s は0となる。

【0217】

JVTのSVC拡張に係る本発明の実施例では、Spatial Scalability Typeが0に等しくなく、ルミナンスサンプル p_0 または q_0 がI__BLマクロブロック予測モードを使用して符号化されたマクロブロック内にある場合、変数 B_s は次のように導かれる。

- ・ルミナンスサンプル p_0 または q_0 がI__BLモード以外のイントラ予測モードを使用して符号化されたマクロブロック内にある場合、4に等しい B_s の値が出力となる。
- ・以下の条件の1つが成り立つ場合、2に等しい B_s の値が出力となる。

i . サンプル p_0 を含むルミナンスブロックまたはサンプル q_0 を含むルミナンスブロックが 0 でない変換係数レベルを含む。

ii . サンプル p_0 を含むルミナンスブロックまたはサンプル q_0 を含むルミナンスブロックに対し、シンタックス要素 `nal_unit_type` は 20 に等しく、`residual_prediction_flag` は 1 に等しく、予測アレイ `resPredX` はサブ章 S.8.5.14 で導かれたように 0 でないサンプルを含む。ここで、X は適用可能な成分 L、 C_b または C_r を示す。

・ 次の条件のうちの 1 つが成り立つ場合、1 に等しい B_s の値が出力となる。

i . ルミナンスサンプル p_0 または q_0 のいずれかが、インター予測モードを使用して符号化されたマクロブロック内にある。

ii . ルミナンスサンプル p_0 または q_0 は、異なる `base_id_plus1` を有する 2 つの別個のスライス内にある。

・ そうでない場合、0 に等しい B_s の値が出力となる。

・ サンプル p_0 および q_0 の双方が、`I_B_L` マクロブロック予測モードを使用して符号化されたマクロブロック内にある場合、1 に等しい B_s の値が出力となる。

【0218】

図 28 を参照し、本発明の一部の実施例について説明する。これら実施例では、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロックの間の境界は、種々のフィルタリング方法を適用するために特徴づけられる。これらフィルタリング方法は、種々のフィルタリング方法をトリガーするか、またはフィルタリングパラメータを調節するために使用できる境界強度インジケータ 312、316 および 320、322 に関連付けできる。

【0219】

これら実施例では、ブロック境界により分離された 2 つの隣接するブロックの特徴を分析し、ブロックに隣接するブロック境界の特徴を定める。一部の実施例では、ブロック間の境界の特徴を定める。

【0220】

実施例では、まずブロックの特徴を分析し、ブロックのうち 1 つをレイヤ間テクスチャ予測を使って符号化するかどうかを判断する (310)。レイヤ間テクスチャ予測を使用して前記隣接するブロックのうちの少なくとも 1 つを符号化する場合、ブロックを分析して、レイヤ間テクスチャ予測以外のイントラ予測方法 (レイヤ内テクスチャ予測方法) により、いずれかのブロックが符号化されているかどうかを判断する (311)。レイヤ間テクスチャ予測以外のイントラ予測方法により、ブロックのうちの 1 つが符号化されている場合、第 1 の境界強度インジケータを使ってターゲット境界の特徴を定める (312)。

【0221】

ブロックのうちの 1 つが、レイヤ間テクスチャ予測以外のイントラ予測方法によって符号化されていない場合、ブロックの特徴を分析し、隣接するブロックのいずれかまたは隣接するブロックのうちの 1 つを予測したブロックが 0 でない変換係数を有するかどうかを判断する (314)。隣接するブロックのいずれかまたは隣接するブロックのうちの 1 つを予測したブロックが 0 でない変換係数を有する場合、第 2 の境界強度インジケータを使ってターゲット境界の特徴を定める (316)。

【0222】

ブロックのうちの 1 つがレイヤ間テクスチャ予測以外のイントラ予測方法では符号化されておらず (311)、隣接するブロックのいずれかがまたは隣接するブロックのうちの 1 つを予測したブロックが、0 でない変換係数を有していない場合 (314)、異なる基準ブロックを参照して、隣接するブロックを予測するかどうかを判断する (318)。隣接するブロックが異なる基準ブロックを参照して予測されている場合 (318)、第 3 の境界強度インジケータを使ってターゲット境界の特徴を定める (320)。

【0223】

10

20

30

40

50

ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法で符号化されておらず(311)、隣接するブロックのいずれもがまたは隣接するブロックのうちの1つを予測したブロックが0でない変換係数を有しておらず(314)、異なる基準ブロックを参照して隣接するブロックを予測していない(318)場合、第4の境界強度インジケータを使ってターゲット境界の特徴を定める(322)。

【0224】

一部の実施例では、境界強度インジケータを使って特定の境界フィルタリングオプションをトリガーできる。一部の実施例では、各インジケータに対して異なるフィルタリング方法を使用できる。一部の実施例では、インジケータに対してフィルタリング方法パラメータを調節できる。一部の実施例では、インジケータがどれだけアグレッシブに境界をフィルタリングするかをトリガーできる。一部の実施例では、第1の境界強度インジケータが境界の最もアグレッシブなフィルタリングをトリガーし、第2、第3および第4の境界強度インジケータが順に、より低いアグレッシブなフィルタリングを次々にトリガーするようになっている。一部の実施例では、第4の境界強度インジケータまたは別のインジケータが関連する境界に対してフィルタリングを全くトリガーしないようになっている。

【0225】

図29を参照し、本発明の一部の実施例について説明する。これら実施例では、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロックの間の境界が種々のフィルタリング方法の適用に対して特徴づけられる。種々のフィルタリング方法をトリガーするか、またはフィルタリングパラメータを調節するのに使用できる境界強度インジケータ336、340、344、348および352、354に、これらフィルタリング方法を関連付けできる。

【0226】

これら実施例では、ブロック境界により分離された2つの隣接するブロックの特徴を分析し、ブロックに隣接するブロック境界の特徴を定める。一部の実施例では、ブロック間の境界の特徴を定める。

【0227】

実施例では、まずブロックの特徴を分析し、これらブロックが空間的スケーラビリティレイヤ内にあるかどうかを判断する(330)。ブロックのうちの1つをレイヤ間テクスチャ予測を使って符号化するかどうかを判断する(332)。前記隣接するブロックのうちの少なくとも1つがレイヤ間テクスチャ予測を使って符号化される場合、ブロックを分析し、いずれかのブロックがレイヤ間テクスチャ予測以外のイントラ予測方法により符号化されているかどうかを判断する(334)。ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法によって符号化されている場合、第1の境界強度インジケータを使ってターゲット境界の特徴を定める(336)。

【0228】

ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法によって符号化されていない場合、隣接するブロックのいずれかが0でない変換係数を有するかどうかを判断するために、ブロックの特徴を分析する(338)。隣接するブロックのうちのいずれかが0でない変換係数を有する場合、第2の境界強度インジケータを使ってターゲット境界の特徴を定める(340)。

【0229】

ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法によって符号化されていない場合、ブロックの特徴を分析し、隣接するブロックのうちの1つを予測したブロックが0でない変換係数を有するかどうかを判断する(342)。隣接するブロックのうちの1つを予測したブロックが0でない変換係数を有する場合、第3の境界強度インジケータを使ってターゲット境界の特徴を定める(344)。

【0230】

ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法では符号化されず(334)、隣接するブロックのいずれもがまたは隣接するブロックのうちの1つを

予測したブロックが0でない変換係数を有しない場合(338、342)、隣接するブロックのうちの1つがインター予測モードを使って符号化するかどうかを決定する判断がなされる(346)。隣接するブロックのうちの1つがインター予測モードを使って符号化される場合、第4の境界強度インジケータを使ってターゲット境界の特徴を定める(348)。

【0231】

ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法では符号化されておらず(334)、隣接するブロックのいずれもがまたは隣接するブロックのうちの1つを予測したブロックが0でない変換係数を有しておらず(338、342)、ブロックがインター予測モードで符号化されない場合(346)、異なる基準ブロックを参照して、隣接するブロックを予測するかどうかを決定する判断を行うことができる(350)。異なる基準ブロックを参照して隣接するブロックを予測する場合(350)、第5の境界強度インジケータを使ってターゲット境界の特徴を定める(352)。

10

【0232】

ブロックのうちの1つがレイヤ間テクスチャ予測以外のイントラ予測方法では符号化されず(334)、隣接するブロックのいずれもがまたは隣接するブロックのうちの1つを予測したブロックが0でない変換係数を有しておらず(338、342)、ブロックがインター予測モードで符号化されず(346)、異なる基準ブロックを参照して隣接するブロックが予測されていない(350)場合、第6の境界強度インジケータを使ってターゲット境界の特徴を定めることができる(354)。

20

【0233】

図30を参照して、本発明の一部の実施例について説明する。これら実施例では、空間的スケーラビリティエンハンスメントレイヤ内の隣接するブロックの間の境界は、種々のフィルタリング方法の適用のために特徴づけられる。種々のフィルタリング方法をトリガーまたはフィルタパラメータを調節するために使用できる境界強度インジケータ365、367、371および373と、これらフィルタリング方法とを関連付けできる。一部の実施例では、0の境界強度インジケータはフィルタ演算のスキップを示す。

【0234】

これら実施例では、ブロック境界によって分離された2つの隣接するブロックの特徴を分析し、これらブロックに隣接したブロック境界の特徴を定める。一部の実施例では、ブロックの間の境界の特徴を定める。

30

【0235】

これら実施例では、SpatialScalabilityTypeは非ゼロでなければならない(360)。次に、ブロックのうちの1つからのルミナンスサンプルをレイヤ間テクスチャ予測を使って符号化するかどうかを決定する別の判断を行う(362)(I__BL)。前記隣接するブロックのうちの少なくとも1つをI__BLを使って符号化する場合、これらブロックを分析し、いずれかのブロックがI__BL以外のイントラ予測方法で符号化されているかどうかを判断する(364)。I__BL以外のイントラ予測方法を用いてブロックのうちの1つを符号化している場合、第1の境界強度インジケータを使ってターゲット境界の特徴を定める(365)。一部の実施例では、第1境界強度インジケータは、最強、すなわち最もアグレッシブなデブロッキングフィルタ演算をトリガーする。一部の実施例では、この第1インジケータは4に等しくなる。

40

【0236】

I__BL以外のイントラ予測方法を用いて、ブロックのうちの1つが符号化されていない場合、ブロック特性を分析し、隣接するブロックのうちのいずれかのルミナンスサンプルが0でない変換係数を有するかどうかを判断する(366)。隣接するブロックのうちのいずれかのルミナンスサンプルが非ゼロ変換係数を有する場合、第2の境界強度インジケータを使用してターゲット境界の特徴を定める(367)。一部の実施例では、この第2の境界強度インジケータは、中間の、すなわち第2に最もアグレッシブなデブロッキングフィルタ演算をトリガーする。一部の実施例では、この第2のインジケータは2に等し

50

くなる。

【0237】

I __ B L 以外のイントラ予測方法を用いて、ブロックのうちの1つが符号化されておらず(364)、いずれかのブロックからのルミナンスサンプルのいずれもが非ゼロ変換係数を有していない場合、隣接するブロックのうちの1つを予測したブロックが非ゼロ変換係数を有するかどうかを決定する判断がなされる(368)。隣接するブロックのうちの1つを予測したブロックが非ゼロ変換係数を有する場合、第2の境界強度インジケータを再び使用し、ターゲット境界の特徴を定めることができる(367)。

【0238】

ブロックのうちの1つがI __ B L 以外のイントラ予測方法を用いて符号化されておらず(364)、隣接するブロックのいずれも(366)が、または隣接するブロックのうちの1つを予測したブロックが、非ゼロ変換係数を有していない(368)場合、隣接するブロックのうちの1つのルミナンスサンプルをインター予測モードを使って符号化するかどうかを決定する判断がなされる(370)。インター予測モードを使って隣接するブロックのうちの1つのルミナンスサンプルを符号化する(370)場合、第3の境界強度インジケータを使ってターゲット境界の特徴を定めることができる(371)。一部の実施例では、この第3の境界強度インジケータは、より弱い、すなわち第3番目にアグレッシブなデブロッキングフィルタ演算をトリガーする。一部の実施例では、この第3のインジケータは1に等しくなる。

【0239】

ブロックのうちの1つがI __ B L 以外のイントラ予測方法により符号化されていない(364)場合、隣接するブロックのいずれも(366)が、また隣接するブロックのうちの1つを予測したブロックも、非ゼロ変換係数を有しておらず(368)、隣接するブロックのルミナンスサンプルがインター予測モードで符号化されない(370)場合、隣接するブロックのいずれかが異なる基準ブロックから予測されるかどうかを決定する判断がなされる(372)。隣接するブロックのいずれかのルミナンスサンプルが異なる基準ブロックを参照して予測される(370)場合、第3の境界強度インジケータを再び使って、ターゲット境界の特徴を定めることができる(371)。

【0240】

ブロックのうちの1つがI __ B L 以外のイントラ予測方法により符号化されていない(364)場合、隣接するブロックのいずれも(366)が、また、隣接するブロックのうちの1つを予測したブロックも、非ゼロ変換係数を有しておらず(368)、隣接するブロックのルミナンスサンプルがインター予測モードで符号化されない場合(370)、隣接するブロックからのルミナンスサンプルが異なる基準ブロックから予測されていない(372)場合、第4の境界強度インジケータを使ってターゲット境界の特徴を定めることができる(373)。一部の実施例では、この第4の境界強度インジケータは、より弱い、すなわち第4番目にアグレッシブなデブロッキングフィルタ演算をトリガーできる。一部の実施例では、この第4番目のインジケータはフィルタリングが行われないことを示すことができる。一部の実施例では、この第4番目のインジケータは0に等しくなる。

【0241】

便宜上、演算を相互に関連した種々の機能的ブロックまたは個別のソフトウェアモジュールとして説明した。しかしながら、このようなことは必要なことではなく、これら機能ブロックまたはモジュールを境界の不明確な状態で、単一ロジックデバイス、プログラムまたは演算に等価的に集めるケースもあり得る。いずれの場合においても、機能的ブロックおよびソフトウェアモジュールまたは説明した特徴は、それら自身で、またはハードウェアもしくはソフトウェアのいずれかにおける別の演算と組み合わせて実現できる。

【0242】

これまで本明細書で用いた用語および表現は、説明するための用語として用いたものであり、制限するために用いたものではなく、かかる用語及び表現の使用に当たり、本明細書に説明し、図示した特徴またはその一部の均等物を排除するものではなく、本発明の範

10

20

30

40

50

囲は特許請求の範囲のみによって定められ、限定されることが認識できよう。

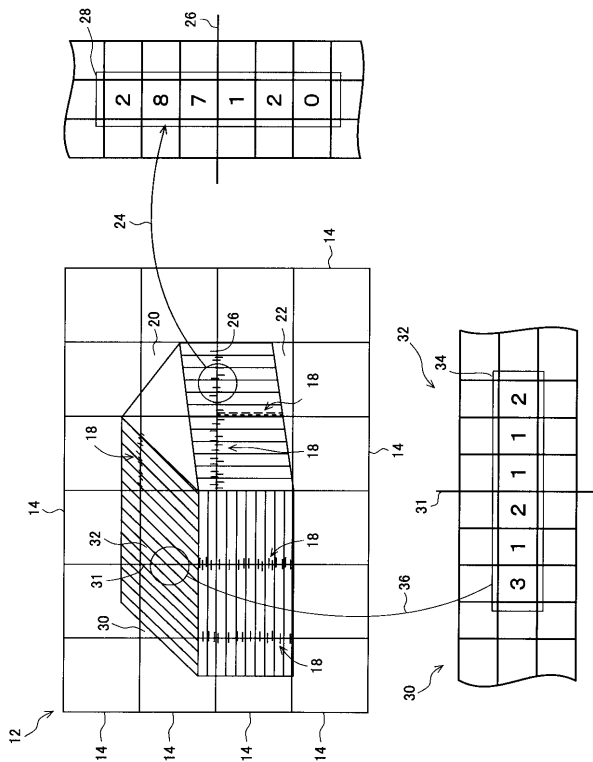
【 0 2 4 3 】

本願は、シジュン・サンによって発明され、2005年3月18日出願された「ピクチャレベルの適用性を有する拡張空間的スケーラビリティ」を発明の名称とする、米国仮特許出願第60/663,161号の権利を主張するものであり、本願は、更に、シジュン・サンによって発明され、2005年5月20日出願された「拡張空間的スケーラビリティにおけるアップサンプリングのための直接補間」を発明の名称とする米国仮特許出願第60/683,060号の権利も主張するものである。本願は更に、シジュン・サンによって発明され、2005年6月1日出願された「空間的スケーラブルなビデオ符号化のための低複雑度のデブロッキングフィルタ方法」を発明の名称とする米国仮特許出願第60/686,676号の権利も主張するものである。本願は、シジュン・サンおよびシャオミン・レイにより発明され、2002年3月29日出願された「ブロックをベースとする動き補償されたビデオ符号化におけるループフィルタリングまたはポストフィルタリングを制御するための方法および装置」を発明の名称とする米国特許出願第10/112,683号の一部継続出願でもあるが、シジュン・サン外により発明され、2001年3月26日出願された「ブロックをベースとする動き補償されたビデオ符号化におけるループフィルタリングまたはポストフィルタリングを制御するための方法および装置」を発明の名称とする米国特許出願第09/817,701号の継続である。本願は更に、シジュン・サンにより発明され、2004年3月11日出願された「境界強度に基づく適応型フィルタリング」を発明の名称とする米国特許出願第10/799,384号の一部継続出願でもあり、米国特許出願第10/799,384号はシジュン・サン外により発明され、2002年9月11日出願されたPCT特許出願第PCT/JP02/09306号の継続出願であり、PCT特許出願第PCT/JP02/09306号は、シジュン・サンおよびシャオミン・レイにより発明され、2001年9月14日出願された「境界強度に基づく適応型フィルタリング」を発明の名称とする米国特許出願第09/953,329号の継続出願である。

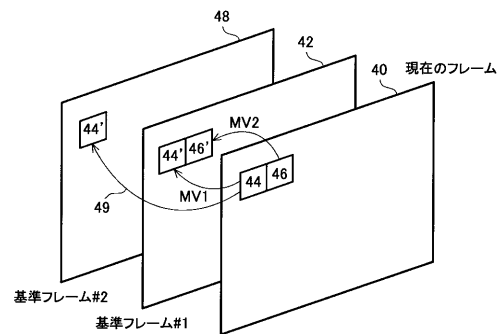
10

20

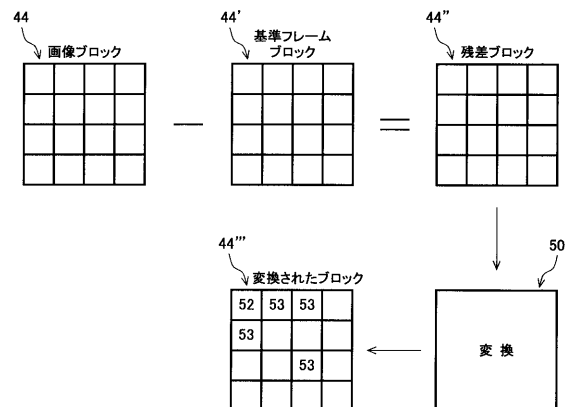
【 図 1 】



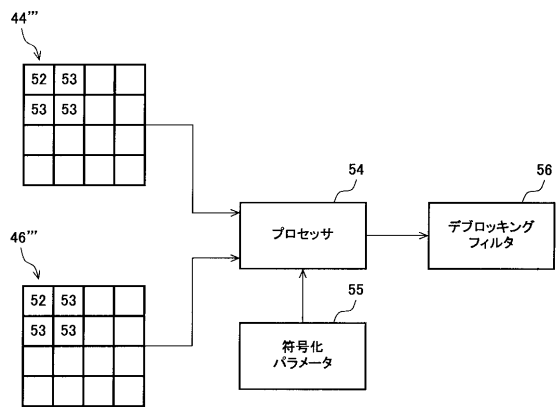
【 図 2 】



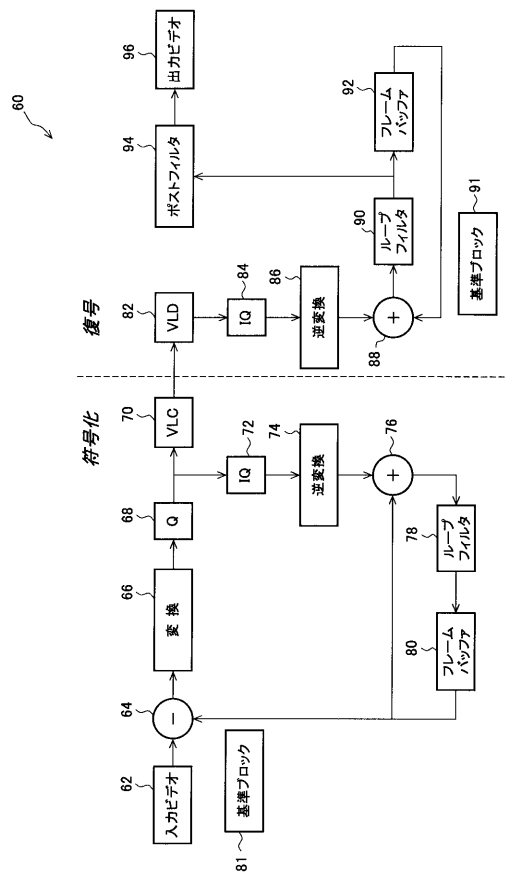
【 図 3 】



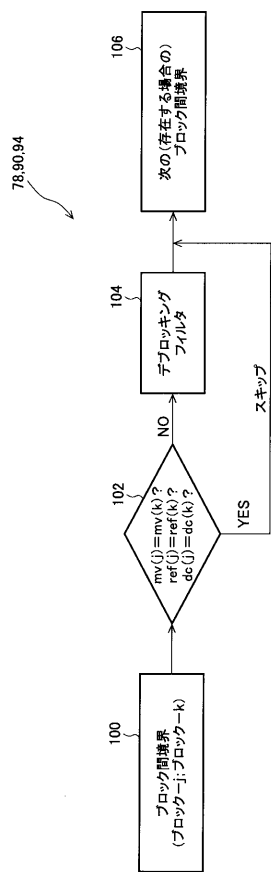
【図 4】



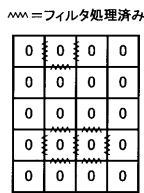
【図 5】



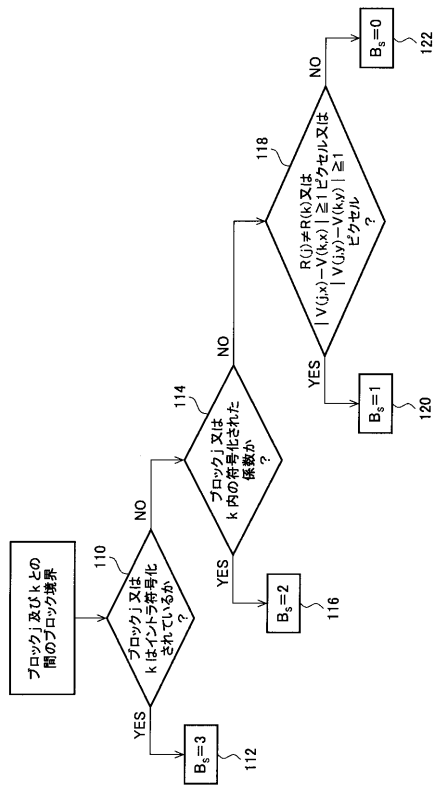
【図 6】



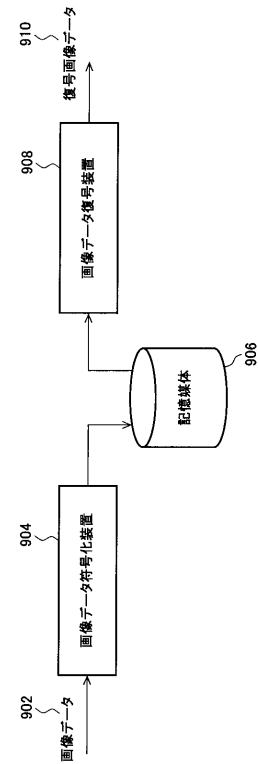
【図 7】



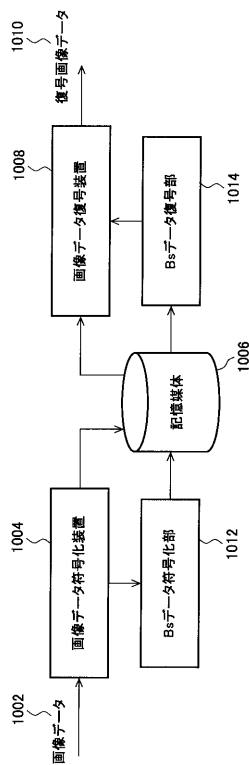
【図 8】



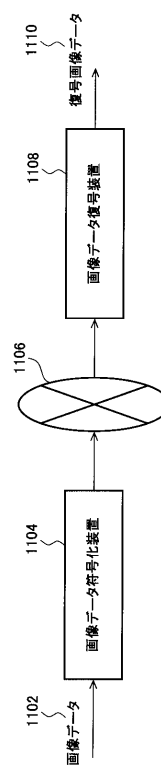
【図 9】



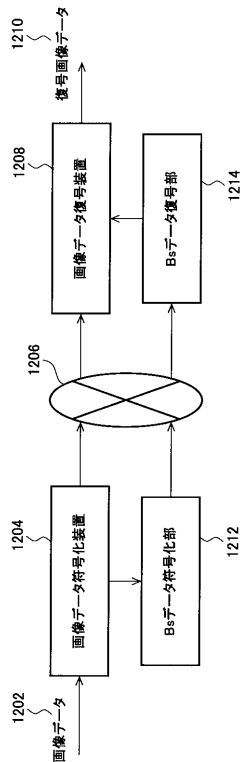
【図 10】



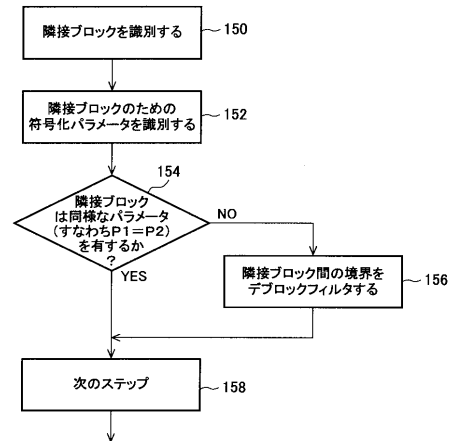
【図 11】



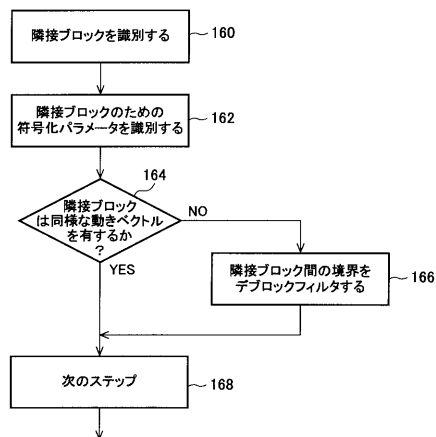
【図 12】



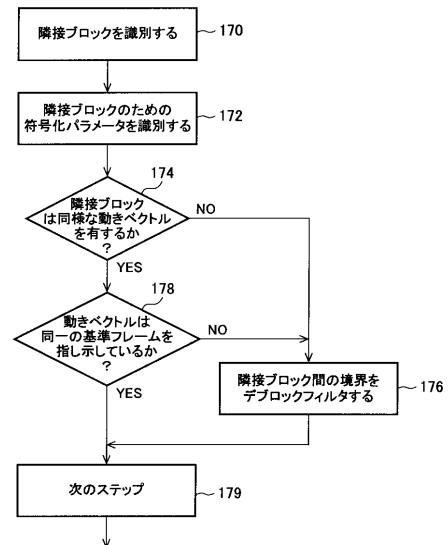
【図 13】



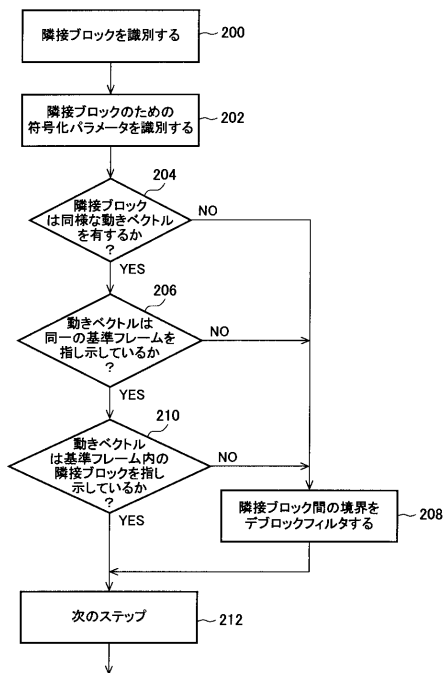
【図 14】



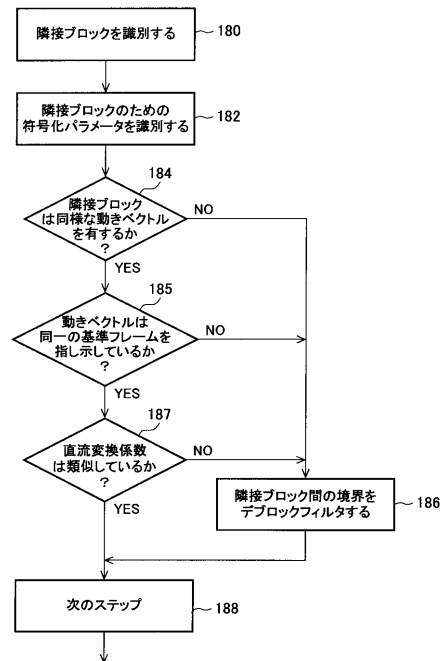
【図 15】



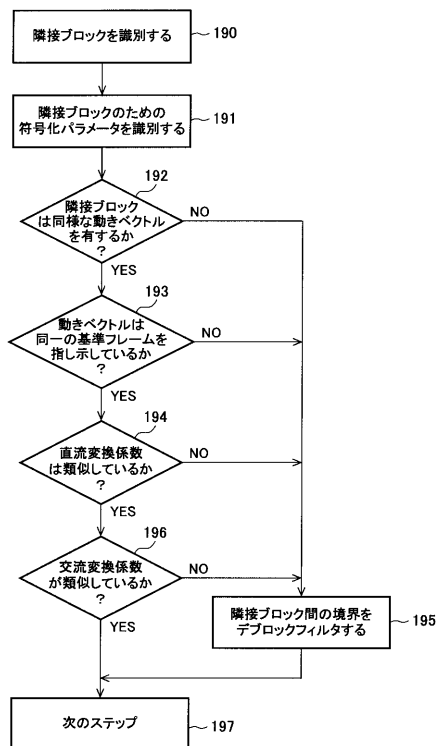
【図 16】



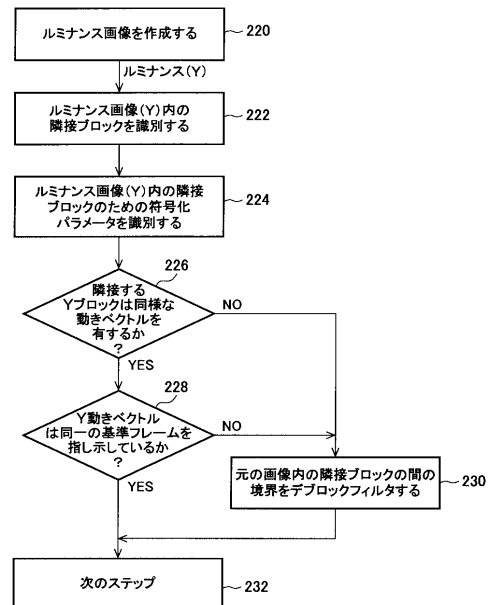
【図 17】



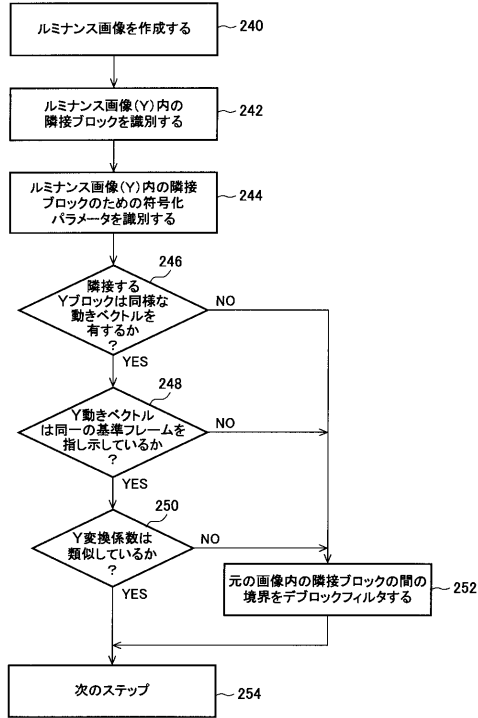
【図 18】



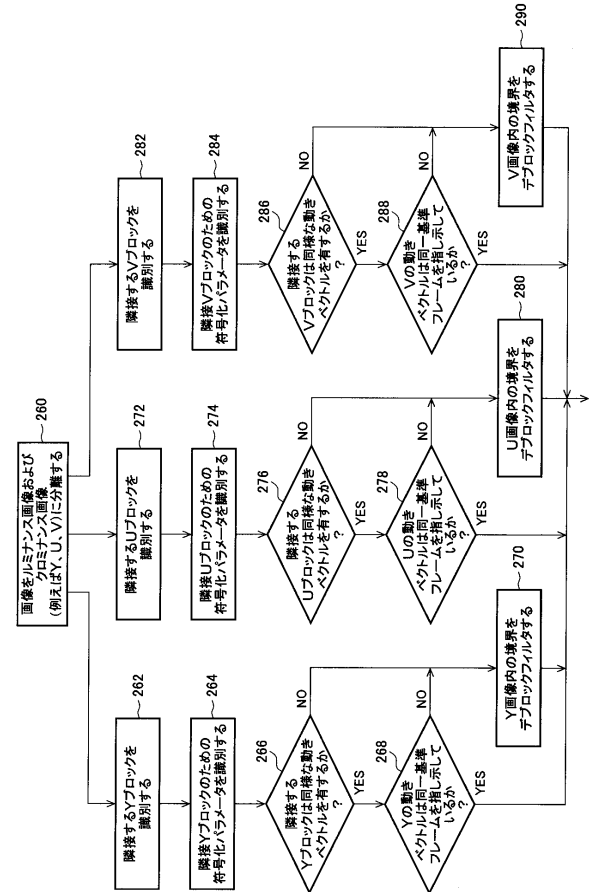
【図 19】



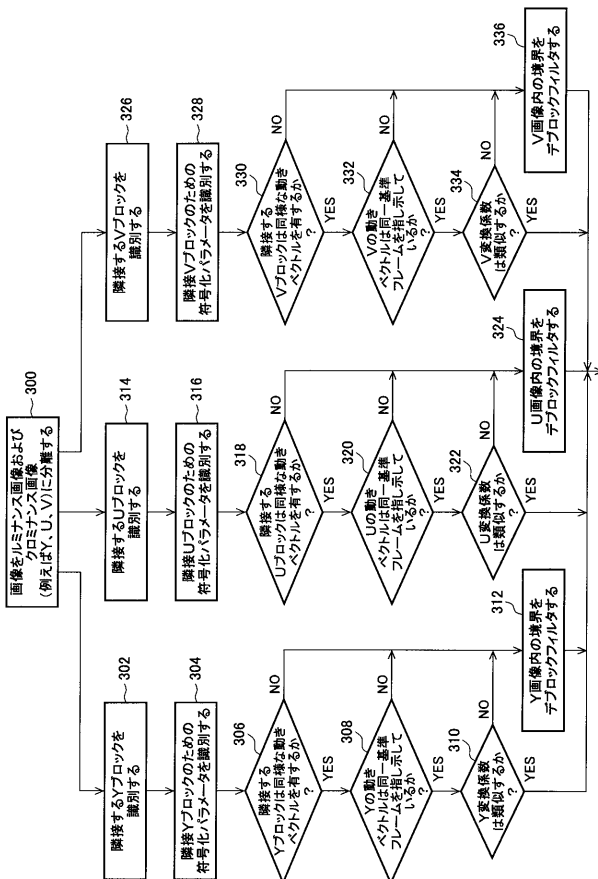
【図 20】



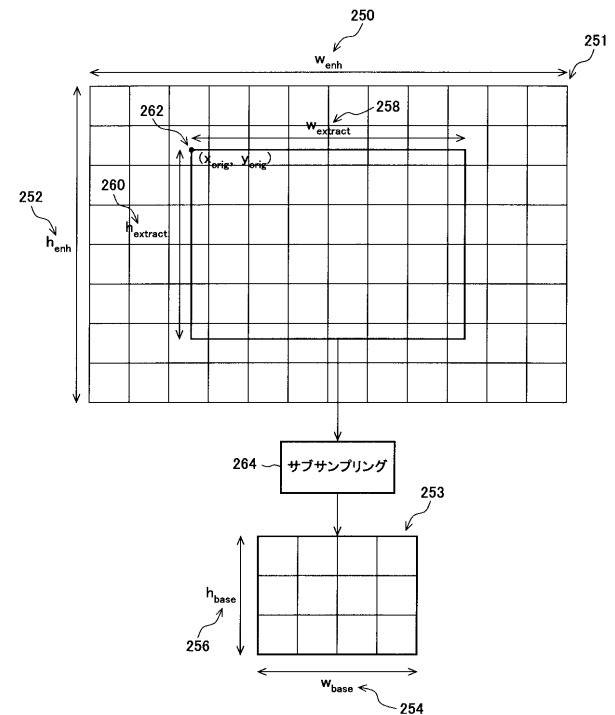
【図 21】



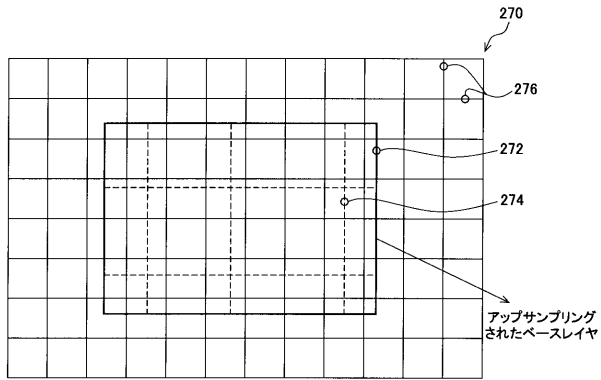
【図 22】



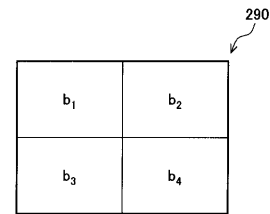
【図 23】



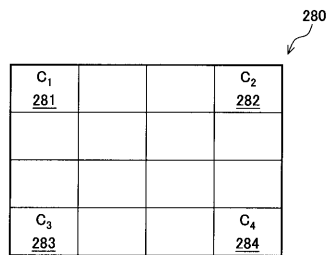
【図 2 4】



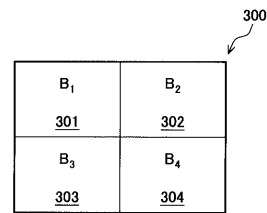
【図 2 6】



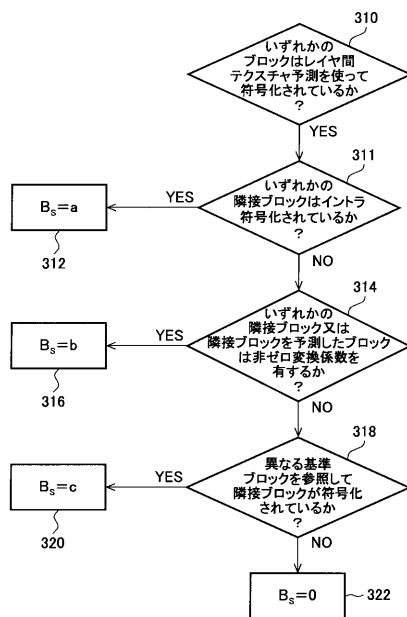
【図 2 5】



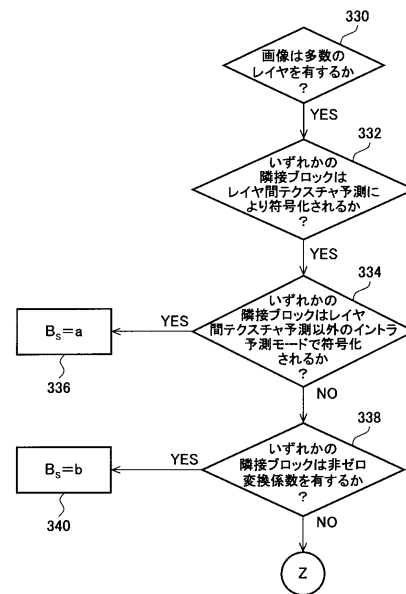
【図 2 7】



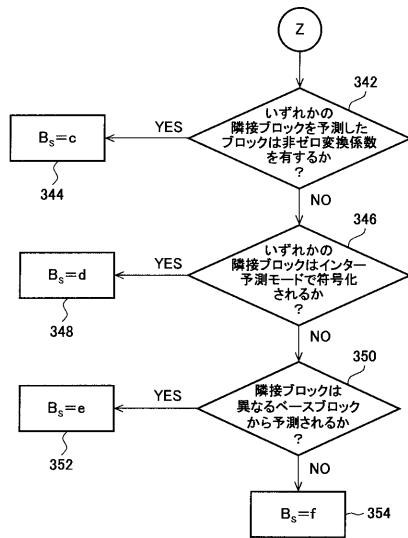
【図 2 8】



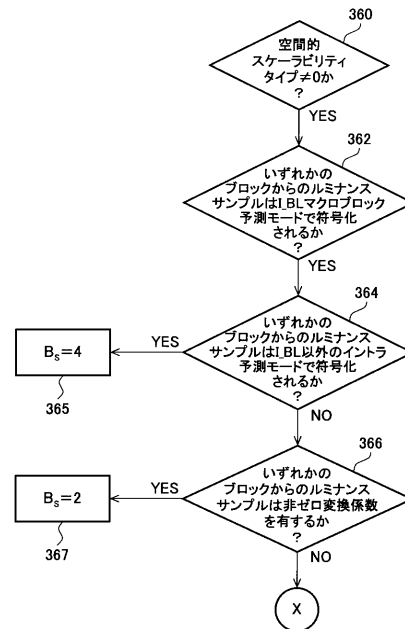
【図 2 9 A】



【図 29 B】



【図 30 A】



【図 30 B】

