

(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2010-11510

(P2010-11510A)

(43) 公開日 平成22年1月14日(2010.1.14)

(51) Int.Cl. F I テーマコード (参考)
HO4N 7/32 (2006.01) HO4N 7/137 Z 5C159

審査請求 有 請求項の数 2 O L (全 18 頁)

(21) 出願番号	特願2009-237136 (P2009-237136)	(71) 出願人	000005049
(22) 出願日	平成21年10月14日 (2009.10.14)		シャープ株式会社
(62) 分割の表示	特願2009-31373 (P2009-31373) の分割	(74) 代理人	100153110 弁理士 岡田 宏之
原出願日	平成14年3月15日 (2002.3.15)	(74) 代理人	100079843 弁理士 高野 明近
(31) 優先権主張番号	09/817, 701	(72) 発明者	シジュン サン アメリカ合衆国, 98683 ワシントン 州, バンクーバー, #47, 16900 エスイー 26番 ドライブ
(32) 優先日	平成13年3月26日 (2001.3.26)	(72) 発明者	シャオーミン レイ アメリカ合衆国, 98607 ワシントン 州, カマス, 4522 エヌダブリュ ヴ ァリー ストリート
(33) 優先権主張国	米国 (US)		

最終頁に続く

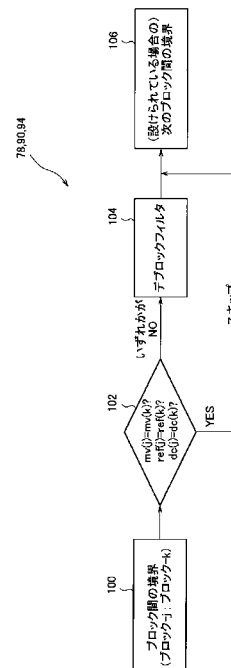
(54) 【発明の名称】 画像符号化方法および画像符号化装置

(57) 【要約】

【課題】動き補償するビデオ符号化におけるブロッキングアーティファクトを減少すること。

【解決手段】再構成された画像内の、隣接する2つのブロック間の境界を選択的にフィルタリングする画像符号化方法であって、再構成された画像を用い、符号化対象ブロック毎に動きベクトルを求め、動きベクトルに基づき動き補償予測を行なう動き補償予測ステップと、隣接する2つのブロックの各々に属する符号化パラメータを用いて、境界にフィルタリングを行なうか否かを境界毎に判定する判定ステップとを有する。そして、判定ステップは、境界に隣接するブロック間で同じ基準画像をポイントしている動きベクトルが互いに同一か否かまたは類似するか否かを評価し、評価に基づいて前記境界にフィルタリングを行なうか否かを判定し、境界に隣接するブロック間で動きベクトルが互いに異なるかまたは類似しない場合には、前記境界にフィルタリングを行なう。

【選択図】図6



【特許請求の範囲】**【請求項 1】**

再構成された画像内の、隣接する 2 つのブロック間の境界を選択的にフィルタリングする画像符号化方法であって、

前記再構成された画像を用い、符号化対象ブロック毎に動きベクトルを求め、該動きベクトルに基づき動き補償予測を行なう動き補償予測ステップと、

前記隣接する 2 つのブロックの各々に属する符号化パラメータを用いて、前記境界にフィルタリングを行なうか否かを境界毎に判定する判定ステップと有し、

前記判定ステップは、前記境界に隣接するブロック間で同じ基準画像をポイントしている動きベクトルが互いに同一か否かまたは類似するか否かを評価し、前記評価に基づいて前記境界にフィルタリングを行なうか否かを判定し、

前記境界に隣接するブロック間で動きベクトルが互いに異なるかまたは類似しない場合には、前記境界にフィルタリングを行なうことを特徴とする画像符号化方法。

【請求項 2】

再構成された画像内の、隣接する 2 つのブロック間の境界を選択的にフィルタリングする画像符号化装置であって、

前記再構成された画像を用い、符号化対象ブロック毎に動きベクトルを求め、該動きベクトルに基づき動き補償予測を行なう動き補償予測手段と、

前記隣接する 2 つのブロックの各々に属する符号化パラメータを用いて、前記境界にフィルタリングを行なうか否かを境界毎に判定する判定手段と備え、

前記判定手段は、前記境界に隣接するブロック間で同じ基準画像をポイントしている動きベクトルが互いに同一か否かまたは類似するか否かを評価し、前記評価に基づいて前記境界にフィルタリングを行なうか否かを判定し、

前記境界に隣接するブロック間で動きベクトルが互いに異なるかまたは類似しない場合には、前記境界にフィルタリングを行なうことを特徴とする画像符号化装置。

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は選択的な画像フィルタリングを行うための方法、画像の符号化または復号化を実行するための装置に関し、より詳細には、動き補償を行う画像符号化及び復号化技術に関する。

【背景技術】**【0002】**

多くのビデオ圧縮規格、例えば H.261、H.263、H.263+、MPEG-1、MPEG-2 および H.264 では、ブロックに基づく動き補償ビデオ符号化が使用される。損失の大きい圧縮プロセスは、復号化された画像内に、画像アーティファクトと称される視覚的なアーティファクトを発生させ得る。画像内のブロック境界に沿ってブロックングアーティファクトが発生し、これらアーティファクトは変換係数の粗い量子化によって生じる。

【0003】

再構成された画像内のアーティファクトを減少させるのに、画像フィルタリング技術を使用できる。再構成された画像とは、逆変換され、復号化された後に生成される画像である。これら技術における経験則では、画像のエッジを保存しながら、画像の他の部分をスムーズにしなければならない。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しなければならない。

【発明の概要】**【発明が解決しようとする課題】****【0004】**

画像ブロックの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、ブロックングアーティファクトを減少するように特別にフィルタリングされる。しかしながら、こ

10

20

30

40

50

のフィルタリングは画像内にボケたアーティファクトを生じさせ得る。隣接するブロックの間にブロックングアーティファクトがほとんどないか、または全くない場合、ローパスフィルタリングは不必要に画像内にボケを混入させると同時に、処理リソースを浪費してしまう。

【0005】

例えば、従来のフィルタリング方法では、フィルタ処理は再構成された画像フレームを一度にひとつしか検討しない。符号化器および復号化器の双方で利用できる動きベクトル情報は使用しない。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れば、(多数の基準フレームシステムに対して)各ブロックの残りの画像の間には大きな差はないことがある。これら2つの隣接するブロックのブロック境界は基準フレーム内で既にフィルタリングできるので、現在フレームに対して再びフィルタリングしてはならない。このような動きベクトル情報を検討することなく、デブロックングフィルタを使用する場合、従来のフィルタ処理はフレームからフレームへ同じ境界を何度も処理することがある。このような不要なフィルタリングによって不要なボケが生じるだけでなく、その結果、余分なフィルタ計算を行わなければならないことになる。

【0006】

本発明は従来技術に関連する上述のごとくの問題およびその他の問題に焦点をあてたもので、ブロックに基づく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及びポストフィルタリングを制御してブロックングアーティファクトを減少させるようにした選択的な画像フィルタリングを行う画像符号化方法および画像符号化装置を提供することを目的とするものである。

【課題を解決するための手段】

【0007】

第1の技術手段は、再構成された画像内の、隣接する2つのブロック間の境界を選択的にフィルタリングする画像符号化方法であって、前記再構成された画像を用い、符号化対象ブロック毎に動きベクトルを求め、該動きベクトルに基づき動き補償予測を行なう動き補償予測ステップと、前記隣接する2つのブロックの各々に属する符号化パラメータを用いて、前記境界にフィルタリングを行なうか否かを境界毎に判定する判定ステップと有し、前記判定ステップは、前記境界に隣接するブロック間で同じ基準画像をポイントしている動きベクトルが互いに同一か否かまたは類似するか否かを評価し、前記評価に基づいて前記境界にフィルタリングを行なうか否かを判定し、前記境界に隣接するブロック間で動きベクトルが互いに異なるかまたは類似しない場合には、前記境界にフィルタリングを行なうことを特徴としたものである。

【0008】

第2の技術手段は、再構成された画像内の、隣接する2つのブロック間の境界を選択的にフィルタリングする画像符号化装置であって、前記再構成された画像を用い、符号化対象ブロック毎に動きベクトルを求め、該動きベクトルに基づき動き補償予測を行なう動き補償予測手段と、前記隣接する2つのブロックの各々に属する符号化パラメータを用いて、前記境界にフィルタリングを行なうか否かを境界毎に判定する判定手段と備え、前記判定手段は、前記境界に隣接するブロック間で同じ基準画像をポイントしている動きベクトルが互いに同一か否かまたは類似するか否かを評価し、前記評価に基づいて前記境界にフィルタリングを行なうか否かを判定し、前記境界に隣接するブロック間で動きベクトルが互いに異なるかまたは類似しない場合には、前記境界にフィルタリングを行なうことを特徴としたものである。

【図面の簡単な説明】

【0009】

【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選択的にスキップする図である。

【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、変換係数を識別する処理を説明するための図

10

20

30

40

50

である。

【図 4】2つの隣接する画像ブロック間で残りの変換係数を比較する処理を説明するための図である。

【図 5】ビデオ画像を符号化し、復号化する処理を説明するための図である。

【図 6】コーデック内でデブロックフィルタリングを選択的にスキップする処理を説明するためのブロック図である。

【図 7】選択的なデブロックフィルタスキップからの結果を含む表である。

【図 8】隣接するブロック間のデブロックフィルタリングは、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、本発明の一実施形態のステップを示すフローチャートである。

10

【図 9】隣接するブロック間のデブロックフィルタリングは、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、本発明の一実施形態のステップを示すフローチャートである。

【図 10】隣接するブロック間のデブロックフィルタリングは、同じ基準フレームをポイント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、本発明の一実施形態のステップを示すフローチャートである。

【図 11】隣接するブロック間のデブロックフィルタリングは、単一基準フレームにおける隣接する基準ブロックをポイント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、本発明の一実施形態のステップを示すフローチャートである。

【図 12】隣接するブロック間のデブロックフィルタリングは、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、本発明の一実施形態のステップを示すフローチャートである。

20

【図 13】隣接するブロック間のデブロックフィルタリングは、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、本発明の一実施形態のステップを示すフローチャートである。

【図 14】隣接するブロック間のデブロックフィルタリングは、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輝度画像における隣接するブロックに依存する、本発明の一実施形態のステップを示す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は、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および類似する変換係数を含むパラメータを有する輝度画像における隣接するブロックに依存する、本発明の一実施形態のステップを示すフローチャートである。

30

【図 16】1つの画像が別個の輝度チャンネルとクロミナンスチャンネルとに分割され、各輝度画像またはクロミナンス画像における隣接するブロック間のデブロックフィルタリングは、類似する動きベクトルを含むパラメータを有する輝度画像内の隣接するブロックに依存する、本発明の一実施形態のステップを示すフローチャートである。

【図 17】1つの画像が別個の輝度チャンネルとクロミナンスチャンネルとに分割され、各輝度画像またはクロミナンス画像における隣接するブロック間のデブロックフィルタリングは、基準フレームにおける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および類似する変換係数を含むパラメータを有する輝度画像における隣接するブロックに依存する、本発明の一実施形態のステップを示すフローチャートである。

40

【発明を実施するための形態】

【0010】

本発明は、画像内の隣接ブロックを識別し、隣接ブロックに対する符号化パラメータを識別する。識別された隣接ブロックに対する符号化パラメータが類似していれば、識別された隣接ブロック間のデブロックフィルタリングをスキップし、識別された隣接ブロックに対する符号化パラメータが実質的に異なれば、デブロックフィルタリングをスキップしない。

【0011】

50

図 1 は、画像ブロック間の類似性に従ってブロックアーティファクトを選択的にフィルタリングする画像 1 2 を示す。この画像 1 2 は多数の画像ブロック 1 4 を含み、これら画像ブロック 1 4 の各々は、記憶されたり、伝送されたりする前に、別々に符号化される。ブロック 1 4 のうちの一部分の間の境界はブロッキングアーティファクト 1 8 を含む。ブロッキングアーティファクトとは符号化プロセスによって生じ得るブロック 1 4 の間の画像の不連続性のことである。隣接する画像ブロックの境界に存在するブロッキングアーティファクトを減少させるにはローパスフィルタを使用する。

【 0 0 1 2 】

例えば、ブロック 2 0 と 2 2 との間にはブロッキングアーティファクト 2 4 が存在する。ブロッキングアーティファクト 2 4 を除くか、または減少するために、ブロック 2 0 と 2 2 との間の境界 2 6 でローパスフィルタを使用する。一実施例におけるローパスフィルタは境界 2 6 の両側からピクセル 2 8 の 1 グループを選択する。このピクセル 2 8 のグループから平均ピクセル値を導き出す。次に、各個々のピクセルと平均ピクセル値とを比較する。さらに、平均ピクセル値の所定範囲外にあるグループ 2 8 内のピクセルを平均ピクセル値に交換する。

【 0 0 1 3 】

上記のように、隣接するピクセル間にブロッキングアーティファクト 2 4 がほとんどないか、または全くない場合、ピクセル 2 8 のグループを不必要にフィルタリングすると、画像内にボケを生じさせ、処理リソースを浪費してしまう可能性がある。スキップモードフィルタリング方式は隣接する画像ブロックに対する動き予想値および補償情報を使用する。動き予想値および補償情報とが類似すれば、デブロックフィルタリングをスキップする。これによって不要な画像のボケを防止できるだけでなく、フィルタリング演算の必要な回数も大幅に減少できる。

【 0 0 1 4 】

例えば、隣接する画像ブロック 3 0 と 3 2 とが類似する符号化パラメータを有する、と符号化プロセス中に判断されたとする。それに応じて、隣接するブロック 3 0 と 3 2 との間の境界 3 1 を横断するように延びるピクセル 3 4 のグループに対して、デブロックフィルタリングはスキップされる。画像 1 2 内のどの隣接するブロック間の水平または垂直境界部に対してもスキップモードフィルタリングを使用できる。

【 0 0 1 5 】

図 2 は、基準フレーム 4 2 および 4 8 と、現在符号化中または復号化中の現在フレーム 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しなければならないかどうかを判断するために、ブロック 4 4 と 4 6 に対する符号化パラメータとを比較する。比較する一つの符号化パラメータとして、ブロック 4 4 および 4 6 に対する動きベクトル (M V) がある。

【 0 0 1 6 】

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 から基準フレーム 4 2 内の関連するブロック 4 4 ' までをポイントし、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 ' までをポイントする。スキップモードフィルタリングは動きベクトル M V 1 および M V 2 が同じ基準フレーム 4 2 内の隣接するブロックをポイントするかどうかをチェックする。これら動きベクトルが基準フレーム内の隣接するブロックをポイントしている (M V 1 = M V 2) 場合、デブロックフィルタリングをスキップする。2 つの画像ブロック 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るかどうかを判断するために、他の符号化情報と共に、この動きベクトル情報を使用することもできる。

【 0 0 1 7 】

符号化プロセスおよび復号化プロセス中に 2 つ以上の基準フレームを使用することができる。例えば、別の基準フレーム 4 8 が存在することがあり、隣接するブロック 4 4 と 4 6 とが異なる基準フレームをポイントする動きベクトルを有することがある。一実施形態では、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

10

20

30

40

50

ックに対する動きベクトルが同じ基準フレームをポイントしているかどうかに応じて決まる。例えば、画像ブロック 44 は基準フレーム 48 をポイントする動きベクトル 49 を有し、画像ブロック 46 は基準フレーム 42 をポイントする動きベクトル MV2 を有するとする。この例では動きベクトル 49 と MV2 とは異なる基準フレームをポイントしているので、デブロックフィルタリングをスキップしない。

【0018】

図3は、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できる別の符号化パラメータを示す。画像フレーム 40 からの画像ブロックは先に図2に示されたように、動きベクトル MV1 によってポイントされた基準フレーム 42 からの基準ブロック 44' と比較される。画像ブロック 44 と基準ブロック 44' との間の比較から、残りのブロック 44'' が出力される。この残りのブロック 44'' に対し、変換 50 を実施すると、変換係数の変換されたブロック 44''' が形成される。一実施例では、この変換 50 は離散的コサイン変換であり、変換されたブロック 44''' は直流 (D.C.) 成分 52 と交流 (A.C.) 成分 53 とを含む。

10

【0019】

直流成分 52 は、画像ブロック 44 における最低周波数変換係数を示す。例えば、画像ブロック 44 内の平均エネルギーを示す係数である。一方、交流成分 53 は画像ブロック 44 内の、より高い周波数成分を示す変換係数を示す。例えば、画像ブロック 44 内のピクセル間の大きいエネルギーの差を示す変換係数である。

【0020】

20

図4は、変換された残りのブロック 44''' および 46''' を示す。2つの変換されたブロック 44''' と 46''' とからの直流成分 52 が、プロセッサ 54 で比較される。直流成分が互いに同一または所定の範囲内にあれば、プロセッサ 54 はデブロックフィルタ演算 56 を通知し、2つの隣接するブロック 44 と 46 の境界の間のデブロックフィルタリングをスキップする。直流成分 52 が類似しなければ、スキップ通知を開始せず、ブロック 44 と 46 との間の境界をデブロックフィルタリングする。

【0021】

一実施形態では、国際通信連合の通信セクター (ITU-T) 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が組み込まれている。この H.26L 方式は 4×4 の整数離散的コサイン変換 (DCT) ブロックしか使用しない。ここでは2つの隣接するブロックの直流成分しかチェックできない。しかしながら、画像ブロックがより大きいサイズ、例えば 8×8 または 16×16 ブロックである場合には、限られた低周波数の交流係数もチェックできる。例えばブロック 44''' に対する上部の直流成分 52 および3つのより低い周波数の交流変換係数 53 とブロック 46''' に対する上部の直流成分 52 および3つの、より低い周波数の交流変換係数 53 とを比較することができる。2つの隣接するブロック 44 と 46 との間の相対的類似性を識別するのに、直流変換係数および/または低周波数の交流変換係数の異なる組み合わせを使用できる。

30

【0022】

プロセッサ 54 は符号化プロセス中に発生された他の符号化パラメータ 55 も受信できる。これら符号化パラメータは、上記のような隣接ブロック 44 および 46 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 54 は隣接する画像ブロック 44 と 46 との間の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ために、これら符号化パラメータのすべてを使用する。画像に関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 54 または異なる処理回路で実施できる。符号化のすべてまたはほとんどを同じプロセッサで実行するケースでは、スキップモードはフィルタリングルーチン内にスキップパラメータを設定することによって簡単に可能になる。

40

【0023】

図5は、ブロックに基づく動き補償符号化器兼復号化器 (コーデック) 60 において、どのようにスキップモードフィルタリングを使用するかを示す。このコーデック 60 はフレーム間の符号化に使用される。ボックス 62 から比較器 64 へ現在フレームからの入力

50

ビデオブロックが与えられる。フレームバッファボックス 80 の出力端は予想された動きベクトル（および可能な基準フレーム数）に従って基準ブロック 81 を発生する。入力ビデオブロックと基準ブロック 81 の差は、ボックス 66 で変換され、次にボックス 68 で量子化される。この量子化された変換ブロックはボックス 70 内で可変長符号化器（VLC）によって符号化され、次に伝送されたり、記憶されたりする。

【0024】

コーデック 60 の符号化セクションは変換された画像をボックス 72 内で第 1 逆量子化（IQ）を行うことにより、変換され、量子化された画像を再構成する。次に、逆量子化された画像はボックス 74 内で逆変換され、再構成された残りの画像を発生する。この再構成された残りのブロックはボックス 76 内で基準ブロック 81 へ加算され、再構成された画像ブロックを発生する。一般に再構成された画像は、ボックス 78 内でループフィルタリングされ、量子化および変換処理によって生じたブロッキングアーティファクトを減少する。次に、フィルタリングされた画像はボックス 80 内でバッファ化され、基準フレームを形成する。ボックス 80 内のフレームバッファリングは動き予想および補償のためにこの再構成された基準フレームを使用する。基準ブロック 81 は比較器 64 内で入力ビデオブロックと比較される。ノード 71 にて符号化セクションから符号化された画像が出力され、次に記憶されるか、または伝送される。

【0025】

コーデック 60 の復号化器部分では、可変長復号化器（VLD）がボックス 82 内で符号化された画像を復号化する。この復号化された画像はボックス 84 内で逆量子化され、ボックス 86 内で逆変換される。ボックス 86 からの再構成された残りの画像は、加算ボックス 88 内で基準ブロック 91 へ加算され、その後、ブロッキングアーティファクトを減少するようにボックス 90 内でループフィルタリングされ、ボックス 92 内で基準フレームとしてバッファ化される。基準ブロック 91 は受信された動きベクトル情報に従ってボックス 92 から発生される。ボックス 90 からフィルタリング出力されるループは、ボックス 94 内で任意にポストフィルタリングされ、更に画像アーティファクトを減少し、その後、ボックス 96 内でビデオ画像としてディスプレイされる。スキップモードフィルタリング方式はボックス 78、90 および 94 内のフィルタリング機能の任意の組み合わせによって実行できる。

【0026】

ビデオ符号化中に利用できる動き予想値および補償情報は、ボックス 78、90 および / または 94 内で、いつデブロックフィルタリングをスキップするかを判断するのに使用する。これら符号化パラメータは既に符号化プロセスおよび復号化プロセス中に発生しているので、特にスキップモードフィルタリングに対して発生または伝送すべき別の符号化パラメータはない。

【0027】

図 6 は、図 5 内の符号化器および復号化器内のフィルタ 78、90 および / または 94 でどのようにスキップモードフィルタリングを使用するかを更に詳細に示している。任意の 2 つの隣接するブロッキング「j」と「k」との間のブロック間境界をまずボックス 100 で識別する。これら 2 つのブロックは画像フレームにて水平方向または垂直方向に隣接していてもよい。判断ボックス 102 はブロック j に対する動きベクトル $mv(j)$ とブロック k に対する動きベクトル $mv(k)$ とを比較する。最初に 2 つの隣接するブロック j と k が同じ基準フレームをポイント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、隣接するブロックに対する動きベクトルは同じ基準フレーム（ $ref(j) = ref(k)$ ）において、隣接するブロック（ $mv(j) = mv(k)$ ）をポイントする。

【0028】

次に、2 つの隣接するブロックに対する残りの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りの画像の間に大きな差異がなければ、例えば 2 つのブロック j と k が同一または類似する直流成分（ $dc(j) = dc(k)$ ）を有する場合、ボックス

10

20

30

40

50

104のデブロックフィルタリング処理をスキップする。次に、スキップモードフィルタリングはボックス106内の次のブロック間境界に移動し、判断ブロック102内で次の比較を行う。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双方に対してスキップモードフィルタリングを実行できる。

【0029】

一実施形態では、ブロックスキップを判断するために、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のみを使用する。別の実施形態では、ブロックスキップを判断するために残りの直流および/または交流係数のみを使用する。更に別の実施形態では、ブロックスキップを判断するために動きベクトル、基準フレームおよび残りの係数のすべてを使用する。

10

【0030】

空間的にサブサンプリングされるクロミナンスチャンネルに対し、スキップモードフィルタリング方式を適用できる。例えば4:2:0のカラーフォーマットシーケンスのケースでは、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うちの輝度成分に対する動きベクトルおよび直流成分の等価性にしか依存できない。動きベクトルと直流成分とが同一であれば、隣接する画像ブロックのうちの輝度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。別の実施形態では、隣接するブロックのうちの各輝度成分およびクロミナンス成分に対して動きベクトルと直流成分を別々に検討する。このケースでは、隣接ブロックに対する輝度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ができるが、一方、同じ隣接ブロックに対する他の輝度成分またはクロミナンス成分はデブロックフィルタリングしない。

20

【0031】

図7は、ITU-T H.26Lのテストモデル-ロングTML5.0を使用したスキップモードフィルタリングの結果を示す表110を含む。この表110はTMLフィルタリング規格の結果と上記スキップモードフィルタリングの結果とを比較したものである。表110の見出し項目SLAの下に、スキップモードフィルタリングを使った符号化の結果が示されている。

【0032】

次の4つの画像をテストした。すなわち每秒30フレーム(fps)における300フレームのAkio_cif、30fpsにおける300フレームのForeman_cif、10fpsにおける100フレームのForeman_qcif、30fpsにおける260フレームのTempeete_cifをテストした。25および30の量子化パラメータ(QP)を使用した。テストの結果から、スキップモードフィルタリングによって視覚的な画質に大きな劣化はないことがわかる。画質に対する画質信号対ノイズ比(PSSNR)は、輝度YチャンネルおよびクロミナンスUおよびVチャンネルに対してほぼ同じままである。しかしながら、スキップモードフィルタリングによって時間を40~70%節約できた。

30

【0033】

スキップモードフィルタリングは多数の画像フレームを符号化または復号化する任意のシステムで使用できる。例えばDVDプレイヤー、ビデオレコーダーまたは通信チャンネルを通して、例えばテレビチャンネルまたはインターネットを通して、画像データを伝送する任意のシステムで使用できる。

40

【0034】

上記スキップモードフィルタリングは演算の一部またはすべてを実行する専用プロセッサシステム、マイクロコントローラ、プログラマブル論理デバイス、またはマイクロプロセッサによって実現できる。ソフトウェアで実現できる上記演算もあれば、ハードウェアで実現できる演算もある。

【0035】

便宜的にこれら演算は種々の相互に接続された機能ブロックまたは別個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、このことは必要ではなく、ケースによっては

50

これら機能ブロックまたはモジュールを明確でない境界の単一の論理デバイス、プログラムまたは演算に等価的に集めることもできる。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または上記特徴は、それ自身によって、またはハードウェアもしくはソフトウェア内どちらかの他の演算と組み合わせて実現できる。

【 0 0 3 6 】

図 8 を参照し、本発明の一部の実施形態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を識別し (1 5 0)、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、それらの類似性を判断する (1 5 4)。符号化パラメータが類似していなければ、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リングを適用する (1 5 6)。符号化パラメータが類似していれば、デブロックフィルタリングをスキップし、次のステップ (1 5 8) に進む。同様に、デブロックフィルタリングを実行する場合は、フィルタリング後に次のステップ (1 5 8) に進む。

【 0 0 3 7 】

本発明の実施形態では図 9 に示されるように、符号化パラメータは動きベクトルである。これら実施形態ではビデオフレーム内の隣接するブロックを識別し (1 6 0)、動きベクトルを含む符号化パラメータを識別する (1 6 2)。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 (1 6 4)。動きベクトルが類似していなければ、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でき (1 6 6)、次のステップ (1 6 8) に進むことができる。動きベクトルが類似していれば、デブロックフィルタリングをスキップし、直接次のステップ (1 6 8) を実行する。

【 0 0 3 8 】

本発明の別の実施形態は、図 1 0 に示されるようにフィルタリングをスキップするかどうかを判断するために多数の符号化パラメータを使用できる。これら実施形態では、隣接するブロックを識別し (1 7 0)、隣接するブロックに対する符号化パラメータを決定する (1 7 2)。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有する動きベクトル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ければ (1 7 4)、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できる (1 7 6)。動きベクトルが類似していれば (1 7 4)、更にフィルタリングプロセスを実行するかどうか判断するために別のパラメータを使用できる。本例では動きベクトルが同じ基準フレームをポイントするかどうかを判断する (1 7 8) ために、動きベクトルを比較することができる。動きベクトルが同じ基準フレームをポイントしていなければ、ブロック間でデブロックフィルタリングを実行できる (1 7 6)。動きベクトルが同じ基準フレームをポイントしていれば、フィルタリングをスキップし、次のステップ (1 7 9) に進むことができる。

【 0 0 3 9 】

フィルタリングを判断するのに別の動きベクトルパラメータを使用できる。図 1 1 に示された実施形態では、ベクトルがポイント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判断するのに使用できるパラメータである。これら実施形態では、隣接するブロックを識別し (2 0 0)、隣接ブロックに対する符号化パラメータを識別する (2 0 2)。次に、動きベクトルを比較し、それらの類似性を判断する (2 0 4)。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を進めることができる (2 0 8)。動きベクトルが類似していれば、別の比較を行って、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ポイントしているかどうかを判断できる。ベクトルが同じフレームをポイントしていなければ、デブロックフィルタリングを進めることができる (2 0 8)。ベクトルが同一基準フレームをポイントしていれば、ベクトルがポイントするブロックを比較することができる (2 1 0)。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ポイントしていなければ、デブロックフィルタリングを進めることができる (2 0 8)。ベクトルが同一基準フレーム内の隣接するブロックをポイントしていれば、デブロックフィルタリングをスキップし、次のステップ (2 1 2) を実行できる。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、ブロック間で大きなアーティファクトを有する可能性が低い、隣接するブロックは、デブロックフィルタリングを行わない。このようにデブロックフィルタリングをスキップすることにより、フィルタリングプロセスによって生じたボケおよび画像劣化を防止できる。不要なフィルタリングを回避すると、処理時間も節約できる。こうして画質が改善され、プロセスにおいて必要となる計算回数は、より少なく済むことになる。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、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。これら無数の組み合わせについては特に詳細に説明しないが、当業者が考えつくことができる範囲内のものであるので、特許請求の範囲内にあると考えられる。

【0040】

本発明の別の実施形態は、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使用できる。図12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)。これら符号化パラメータは変換係数だけでなく動きベクトルパラメータも含むことができる。

【0041】

次に、類似性を判断するために動きベクトルを比較する(184)。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を実行できる(186)。動きベクトルが類似していれば、動きベクトルが同じ基準フレームをポイントしているかどうかを判断するために、動きベクトルデータを分析する。動きベクトルが同じ基準フレームをポイントしていなければ(185)、フィルタリングを進めることができる(186)。

【0042】

動きベクトルが同じ基準フレームをポイントしていれば(185)、変換係数を比較し、更にフィルタリングプロセスを行うかどうかを判断する。本例では、離散的コサイン変換(DCT)方法または他の方法を通して得られる直流変換係数を隣接するブロックに関して比較できる。直流変換係数が類似していなければ(187)、デブロックフィルタリングを実行できる(186)。直流変換係数が類似していれば、フィルタリングをスキップし、本方法およびシステムは次のステップ(188)に進むことができる。

【0043】

本発明の更に別の実施形態は、フィルタリングオプションを判断するのに、交流変換係数を利用できる。図13には、交流変換係数を評価する別のステップと共に、図12を参照して説明した実施形態に類似する実施形態が示されている。これら実施形態では、ブロック(190)およびそれらの符号化パラメータ(191)を識別する。動きベクトルにおける類似性(192)、動きベクトルターゲットフレーム(193)および直流変換係数(194)も比較する。これらパラメータにおいて類似性がある場合、交流変換係数を比較し(196)、これらが類似していれば、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(197)。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まで進む(197)。

【0044】

交流変換係数はより大きいブロックで重要性を有しやすいが、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できる。

【0045】

本発明の一部の実施形態では、画像のフォーマットおよび利用するカラースペースに応じて1つの画像を種々の輝度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。次の例では、YUVカラースペースについて説明するが、これら実施形態では多くの別のフォーマットおよびカラースペースを使用できる。Ci e L A B、Y c r C eおよびその他のスペースも使用できる。別の実施形態では、R G Bのようなカラースペースを使用できる。

【0046】

図14を参照して、本発明の一部の実施形態について説明する。これら実施形態では、

10

20

30

40

50

画像から輝度データを抽出し、輝度画像を形成する(220)。次に、輝度画像において、隣接するブロックを識別し(222)、隣接するブロックに対する符号化パラメータも識別する(224)。他の実施形態と同じように、隣接するブロックの動きベクトルを比較し、類似性を判断する(226)。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を実行し(230)、ベクトルが類似していれば、ベクトルが同じ基準フレームをポイントしているかどうか判断する(228)ために更に分析を行う。ベクトルが異なる基準フレームをポイントしていれば、輝度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(230)。ベクトルが同じフレームをポイントしていれば、デブロックフィルタリングをスキップし、先のフィルタリングを行うことなく、次のステップを実行する(232)。フィルタリングを実行する場合、フィルタリングプロセスの後に次のステップを実行する(232)。従って、輝度チャンネル内のデータの分析を使用して、輝度データと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。

10

20

30

40

50

【0047】

図15に示されている他の関連する実施形態では、輝度画像を形成し(240)、輝度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)。輝度画像ブロックに対して符号化パラメータも識別する(244)。その後、動きベクトルを比較し、類似性を判断する(246)。大きな類似性が存在していなければ、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(252)。動きベクトルが類似していれば、動きベクトルのターゲットフレームを比較し、ベクトルが同じ基準フレームをポイントしているかどうかを判断する。ベクトルが同じ基準フレームをポイントしていなければ、フィルタリングを実行する。ベクトルが同じ基準フレームをポイントしていれば、輝度(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ければ、フィルタリングを実行し、変換係数が類似していれば、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、フィルタリング動作の後に次のステップ(254)を実行する。

【0048】

画像は更に、一般に輝度チャンネルとクロミナンスチャンネルに対応する成分チャンネルに分割できる。本発明の一部の実施形態では、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。

【0049】

一例として、図16を参照し、実施形態を説明できる。この実施形態では、1つの画像を別々の輝度チャンネル(Y)と多数のクロミナンスチャンネル(U、V)に分割する(260)。これら実施形態では、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)。符号化パラメータ、例えば動きベクトルデータも各チャンネルにおけるこれらブロックに対して識別する(264、274、284)。次に、他の実施形態のように符号化パラメータを比較し、類似性を判断できる。これら実施形態では、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し、各チャンネルにおけるフィルタリングオプションを判断できる。チャンネル画像に対する動きベクトルが類似していなければ(266、276、286)、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を実行する(270、280、290)。動きベクトルが類似していれば、ターゲット基準フレームを比較する(268、278、288)。チャンネルにおける隣接するブロックに対するベクトルが同じ基準フレームをポイントしていれば、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ポイントしていれば、フィルタリングを実行する(270、280、290)。

【0050】

他の実施形態のように、これらのチャンネル化された実施形態はフィルタリングオプションを行うかどうかの判断をするために、変換係数データを使用できる。図17に示されるように、図16を参照して説明した方法およびシステムは更にチャンネル変換係数を比較できる(310、322、334)。係数が類似していなければ、フィルタリングを実行する(312、324、336)。係数が類似していれば、フィルタリングをスキップ

する。

【 0 0 5 1 】

各チャンネルにおけるフィルタリング動作をするかどうかを判断するのに、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。これら実施形態に対して直流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判断し、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。例えば一部の実施形態では、双方のクロミナンスチャンネルを組み合わせ、分析できる。別のチャンネルにおけるフィルタリングオプションを判断するのに、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まれた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判断してもよいし、この逆に、Vクロミナンスチャンネルから得たパラメータを比較して、U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判断してもよい。

10

【 0 0 5 2 】

以上で本発明の種々の実施形態における発明の原理について説明し、図示したが、本発明はかかる原理から逸脱することなく、配置および細部について発明を変形できることは明らかである。特許請求の範囲内にあるすべての変形例および変更例について権利を請求するものである。

【 符号の説明 】

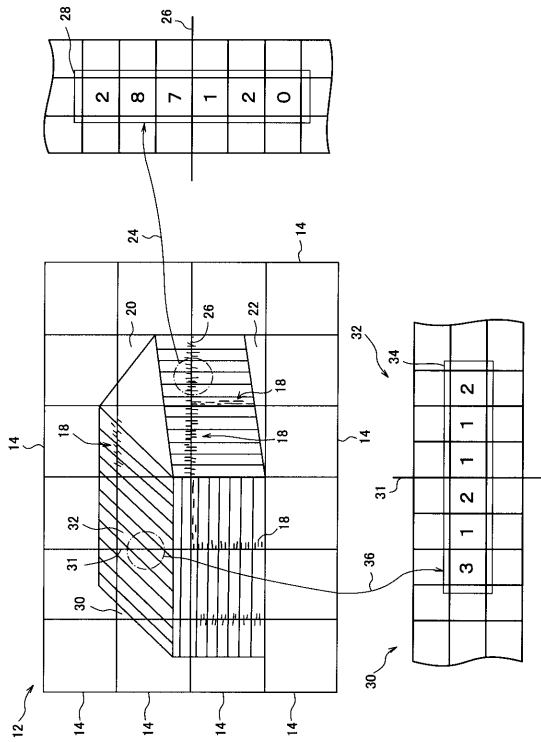
【 0 0 5 3 】

1 2 ... 画像、1 4 ... 画像ブロック、1 8 ... ブロッキングアーティファクト、2 0 , 2 2 ... ブロック、2 4 ... ブロッキングアーティファクト、2 6 ... 境界、2 8 ... ピクセル、3 0 , 3 2 ... 画像ブロック、3 1 ... 境界、3 4 ... ピクセル、4 0 ... 現在フレーム、4 2 , 4 8 ... 基準フレーム、4 4 , 4 6 ... 画像ブロック、M V 1 , M V 2 , 4 9 ... 動きベクトル、4 4 ' , 4 6 ' ... 基準フレームブロック、4 4 '' ... 残りのブロック、4 4 ''', 4 6 ''', ... 変換されたブロック、5 0 ... 変換、5 2 ... 直流成分、5 3 ... 交流成分、5 4 ... プロセッサ、5 5 ... 符号化パラメータ、5 6 ... デブロックフィルタ、6 0 ... ブロックに基づく動き補償符号化器兼復号化器 (コーデック)、6 2 ... 入力ビデオ、6 4 ... 比較器、6 6 ... 変換、6 8 ... 量子化、7 0 ... 可変長符号化器 (VLC)、7 1 ... ノード、7 2 , 8 4 ... 逆量子化、7 4 , 8 6 ... 逆変換、7 6 , 8 8 ... 加算、7 8 , 9 0 ... ループフィルタ、8 0 , 9 2 ... フレームバッファ、8 1 , 9 1 ... 基準ブロック、8 2 ... 可変長復号化器 (VLD)、9 4 ... ポストフィルタ、9 6 ... 出力ビデオ、1 0 0 ... ブロック間境界、1 0 2 ... 判断、1 0 4 ... デブロックフィルタ、1 0 6 ... 次のブロック間の境界、1 1 0 ... TMLフィルタリング規格の結果と上記スキップモードフィルタリングの結果との比較表。

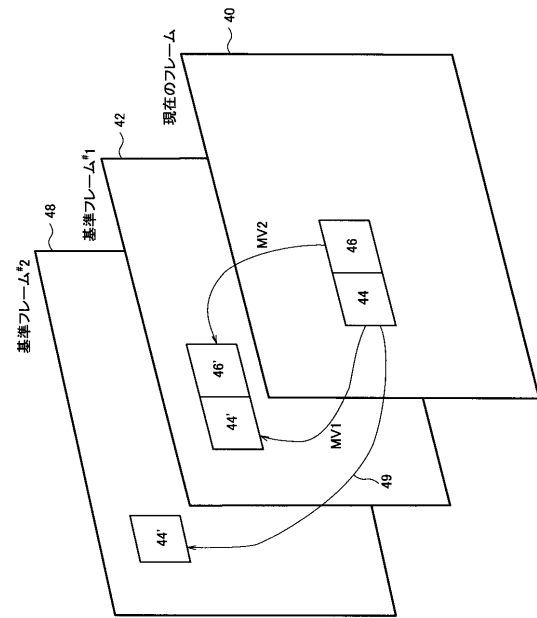
20

30

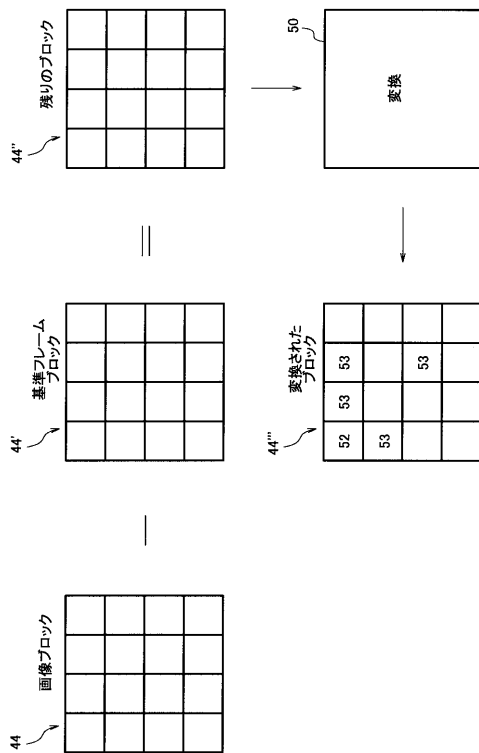
【 図 1 】



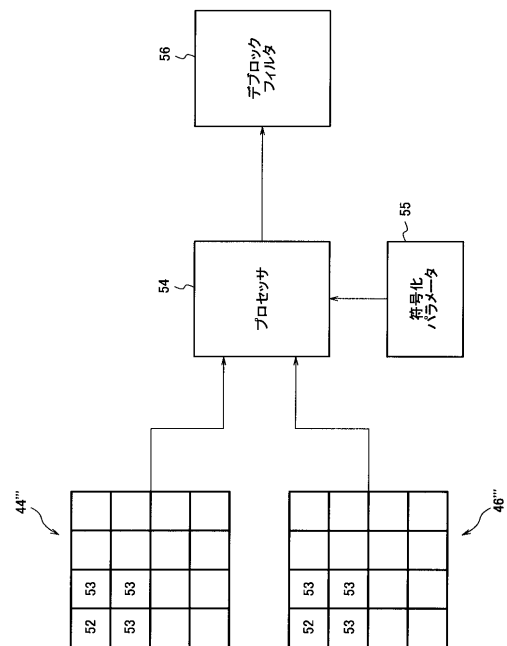
【 図 2 】



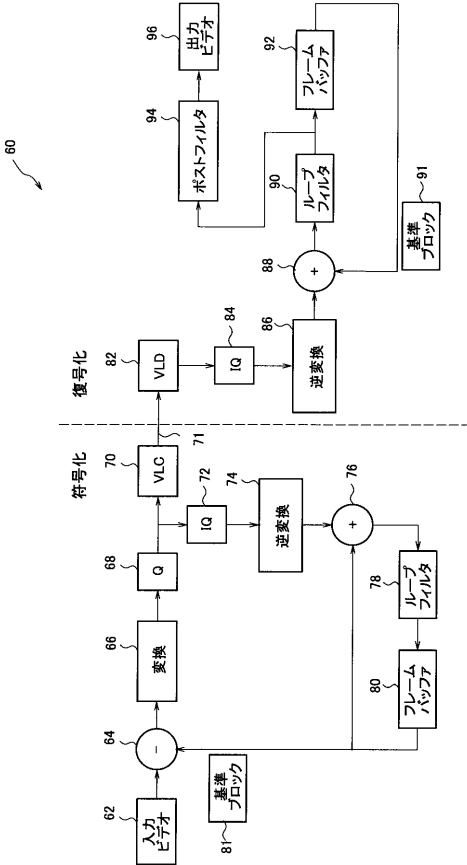
【 図 3 】



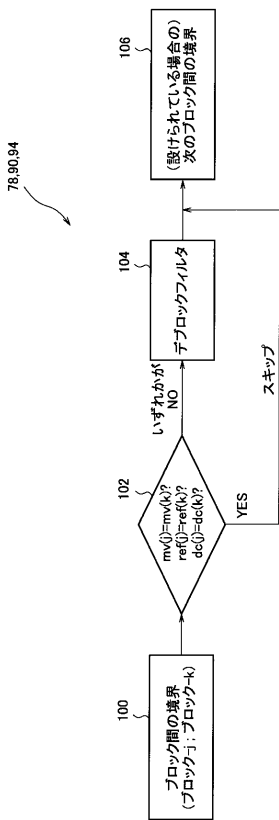
【 図 4 】



【図 5】



【図 6】



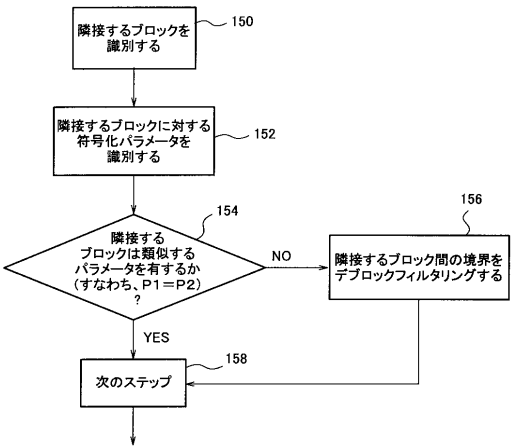
【図 7】

110

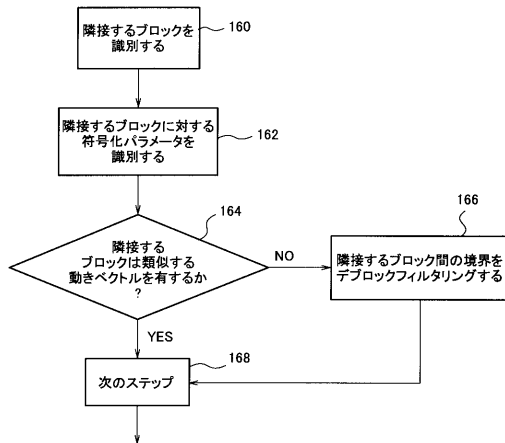
表 1. TMLと提案されているループフィルタリング方式との比較

ビデオシーケンス	QP	ビットレート(bps)			PSNR(Y)			PSNR(U)			PSNR(V)			新しいループフィルタの相対的時間的節約率
		TML	SLA	TML	SLA	TML	SLA	TML	SLA	TML	SLA	TML	SLA	
Akivo_cif 30fpsで300フレーム	25	33151	32346	34050	34161	38934	39042	40300	40369	63.0%				
	30	22775	22295	30797	30920	36610	36964	38680	38771	68.2%				
Foreman_cif 30fpsで100フレーム	25	165115	162740	30835	31006	38124	38174	38966	39030	43.1%				
	30	101357	100215	27580	27836	36745	36811	37267	37362	42.1%				
Foreman_qcif 10fpsで100フレーム	25	26681	26677	29822	29931	37586	37631	37773	37938	38.4%				
	30	15999	15822	26250	26435	36432	36323	36222	36283	41.3%				
Tempete_cif 30fpsで260フレーム	25	336200	329115	28277	28490	33982	34143	36009	36184	45.3%				
	30	188133	159789	24563	24927	32334	36628	34512	34799	50.6%				

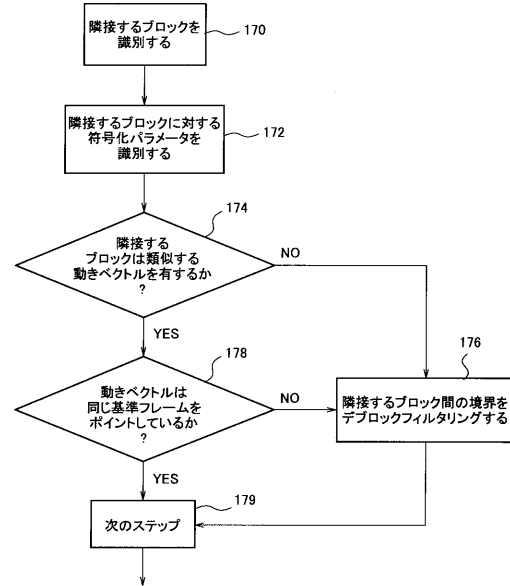
【図 8】



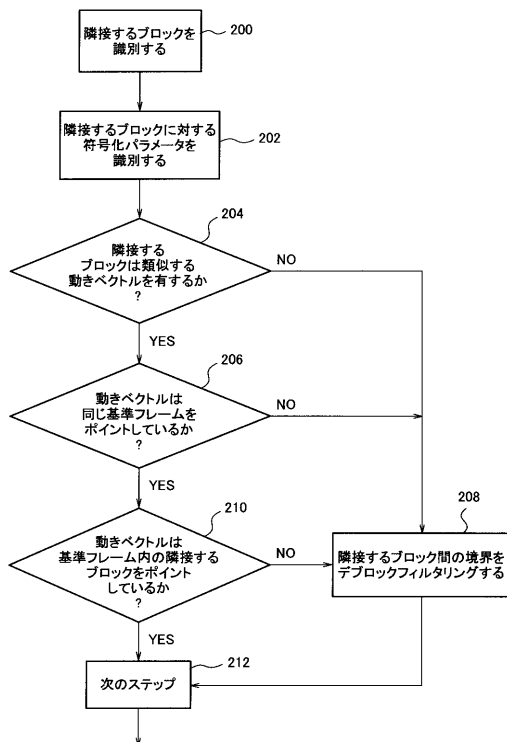
【図 9】



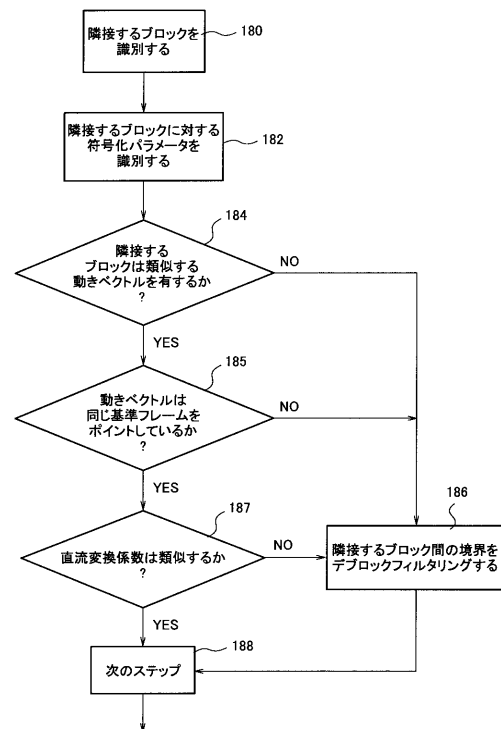
【図 10】



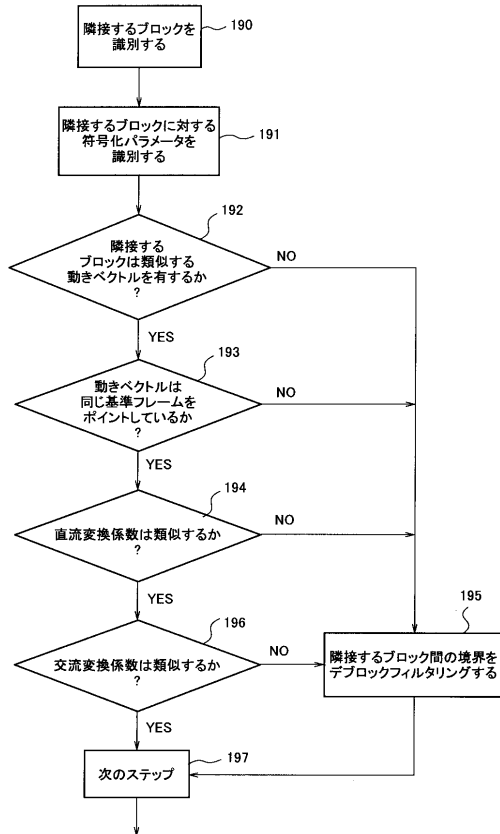
【図 11】



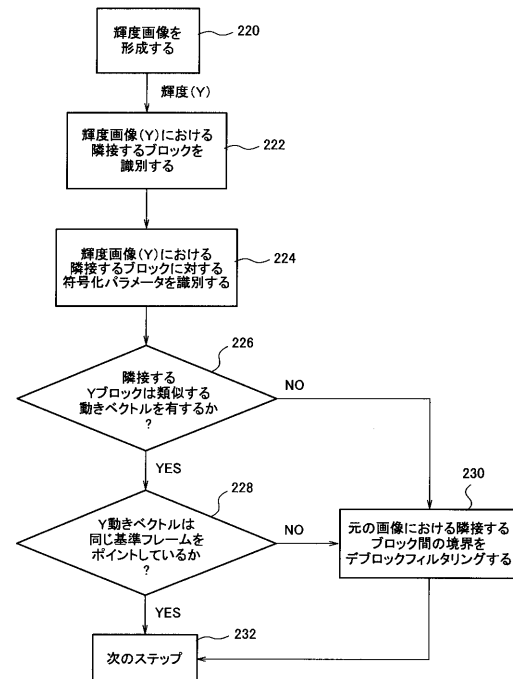
【図 12】



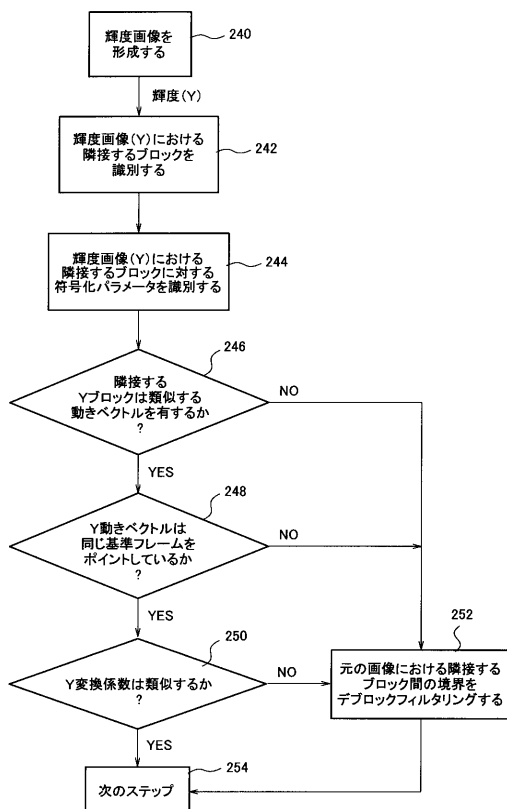
【図 13】



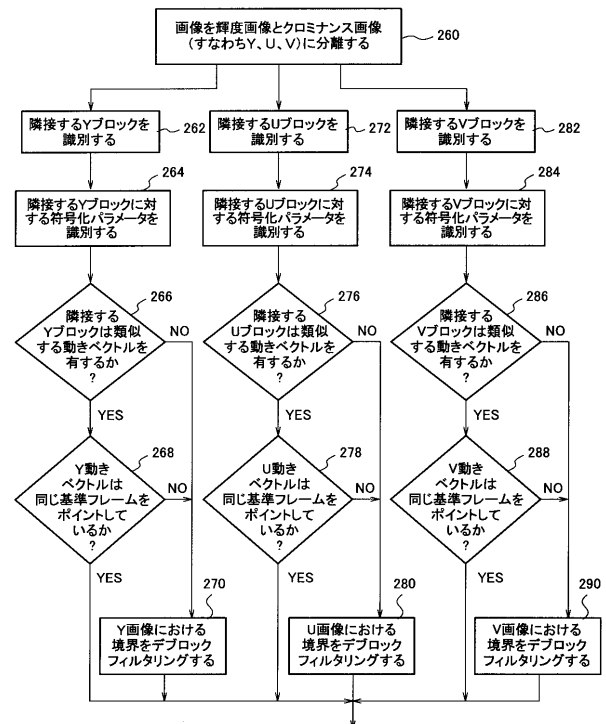
【図 14】



【図 15】



【図 16】



【図 17】



フロントページの続き

F ターム(参考) 5C159 KK03 MA05 MA12 MA21 MC11 MC38 ME01 NN08 NN21 PP04
PP16 TA69 TB08 TC04 TC12 TD05 UA02 UA12 UA16 UA33