

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5579936号
(P5579936)

(45) 発行日 平成26年8月27日 (2014. 8. 27)

(24) 登録日 平成26年7月18日 (2014. 7. 18)

(51) Int. Cl.	F I
HO 4 N 19/117 (2014. 01)	HO 4 N 19/117
HO 4 N 19/137 (2014. 01)	HO 4 N 19/137
HO 4 N 19/196 (2014. 01)	HO 4 N 19/196
HO 4 N 19/61 (2014. 01)	HO 4 N 19/61
HO 4 N 19/86 (2014. 01)	HO 4 N 19/86

請求項の数 29 (全 26 頁)

(21) 出願番号	特願2013-531582 (P2013-531582)	(73) 特許権者	503260918
(86) (22) 出願日	平成23年8月10日 (2011. 8. 10)		アップル インコーポレイテッド
(65) 公表番号	特表2013-542670 (P2013-542670A)		アメリカ合衆国 95014 カリフォルニア州 クパチーノ インフィニット ループ 1
(43) 公表日	平成25年11月21日 (2013. 11. 21)	(74) 代理人	100076428
(86) 国際出願番号	PCT/US2011/047205		弁理士 大塚 康徳
(87) 国際公開番号	W02012/047373	(74) 代理人	100112508
(87) 国際公開日	平成24年4月12日 (2012. 4. 12)		弁理士 高柳 司郎
審査請求日	平成25年4月8日 (2013. 4. 8)	(74) 代理人	100115071
(31) 優先権主張番号	12/895, 688		弁理士 大塚 康弘
(32) 優先日	平成22年9月30日 (2010. 9. 30)	(74) 代理人	100116894
(33) 優先権主張国	米国 (US)		弁理士 木村 秀二
		(74) 代理人	100130409
			弁理士 下山 治

最終頁に続く

(54) 【発明の名称】 最適化されたデブロッキングフィルタ

(57) 【特許請求の範囲】

【請求項 1】

符号化ビデオに対するデブロッキングパラメータを判定する方法であって、
デブロッキングベクトルに従って復号化ピクチャのデブロッキングを実行するステップと、
ここで、前記デブロッキングベクトルは、それぞれが当該デブロッキングベクトルの各次元となるデブロッキングパラメータで表わされる；

前記デブロッキングされた復号化ピクチャからデブロッキング誤差を推定するステップと、

前記推定された誤差が所定の閾値を上回る場合には、ベクトルの次元によって表される各デブロッキングパラメータに対して、

前記次元の各々において前記デブロッキングベクトルを進め、
前記進められた次元に従って前記復号化ピクチャをデブロッキングし、
前記進められた次元に従ってデブロッキングされた前記復号化ピクチャからデブロッキング誤差を推定するステップと、

前記ベクトルの次元それぞれに対して算出された前記推定されたデブロッキング誤差を表す傾斜ベクトルに従って、次の反復における前記デブロッキングベクトルを変更するステップと

を反復することを特徴とする方法。

【請求項 2】

前記推定された誤差が前記閾値を上回らない場合、符号化ピクチャデータ及び前記デブ

ロックンクベクトルの識別子を復号器へ送信するステップを更に備えることを特徴とする請求項 1 に記載の方法。

【請求項 3】

前記デブロックンク誤差が所定の閾値を下回るまで前記方法を反復するステップを更に備えることを特徴とする請求項 1 に記載の方法。

【請求項 4】

前記方法の 2 回目の反復において、

最初の反復と前記 2 回目の反復との間の推定された誤差の変化率を推定するステップと

、
前記変化率が第 2 の閾値を下回る場合に、符号化ピクチャデータ及び前記デブロックンクベクトルの識別子を復号器へ送信するステップと、

を更に備えることを特徴とする請求項 1 に記載の方法。

【請求項 5】

前記傾斜の大きさが第 2 の閾値を下回る場合に、符号化ピクチャデータ及び前記デブロックンクベクトルの識別子を復号器へ送信するステップを更に備えることを特徴とする請求項 1 に記載の方法。

【請求項 6】

前記方法の後続の反復において、

前記変更されたデブロックンクベクトルが前記方法の前の反復において使用されたデブロックンクベクトルと一致するかを判定するステップと、

前記変更されたデブロックンクベクトルが前記方法の前の反復において使用されたデブロックンクベクトルと一致する場合、符号化ピクチャデータ及び前記デブロックンクベクトルの識別子を復号器へ送信するステップと、

を更に備えることを特徴とする請求項 1 に記載の方法。

【請求項 7】

前記進めるステップは、前記次元の各々において前記デブロックンクベクトルを増分するステップを備えることを特徴とする請求項 1 に記載の方法。

【請求項 8】

前記進めるステップは、前記次元の各々において前記デブロックンクベクトルを先行の反復の前記傾斜により判定された方向に調整するステップを備えることを特徴とする請求項 1 に記載の方法。

【請求項 9】

最初の反復の前記デブロックンクベクトルではデフォルト値が設定されることを特徴とする請求項 1 に記載の方法。

【請求項 10】

最初の反復の前記デブロックンクベクトルは、先行ピクチャに対して前記方法を実行することにより最終的に取得されたデブロックンクベクトルの値に設定されることを特徴とする請求項 1 に記載の方法。

【請求項 11】

最初の反復の前記デブロックンクベクトルは、現在のピクチャと共通の符号化の割り当てを有する先行ピクチャに対して前記方法を実行することにより最終的に取得されたデブロックンクベクトルの値に設定されることを特徴とする請求項 1 に記載の方法。

【請求項 12】

最初の反復の前記デブロックンクベクトルは、現在のピクチャと同様の動き特性を有する先行ピクチャに対して前記方法を実行することにより最終的に取得されたデブロックンクベクトルの値に設定されることを特徴とする請求項 1 に記載の方法。

【請求項 13】

前記デブロックンクピクチャの前記推定される誤差は、前記デブロックンクされた復号化ピクチャとフィルタリングされていないソースピクチャデータとの比較から判定されることを特徴とする請求項 1 に記載の方法。

10

20

30

40

50

【請求項 14】

前記デブロッキングピクチャの前記推定される誤差は、前記デブロッキングされた復号化ピクチャとフィルタリングされたソースピクチャデータとの比較から判定されることを特徴とする請求項 1 に記載の方法。

【請求項 15】

ビデオ符号化方法であって、
動き補償によりソースピクチャのブロックを符号化するステップと、
参照ピクチャのブロックを復号するステップと、ここで、当該ステップは、
反復傾斜利用探索処理によりデブロッキングフィルタリング動作のパラメータを推定するステップと、

最終的に推定されたデブロッキングパラメータに従って、復号化された参照ピクチャをデブロッキングフィルタリングするステップとを含む；

参照ピクチャ記憶装置に前記デブロッキングピクチャを格納するステップと、
前記符号化ピクチャのデータ及び前記最終的に推定されたデブロッキングパラメータをチャンネルへ送信するステップとを備え、

前記反復傾斜利用探索処理は、
デブロッキングベクトルに従って復号化ピクチャのデブロッキングを実行するステップと、ここで、前記デブロッキングベクトルは、それぞれが当該デブロッキングベクトルの各次元となるデブロッキングパラメータで表わされる；

前記デブロッキングされた復号化ピクチャからデブロッキング誤差を推定するステップと、

前記推定された誤差が所定の閾値を上回る場合には、ベクトルの次元によって表される各デブロッキングパラメータに対して、

前記次元の各々において前記デブロッキングベクトルを進め、
前記進められた次元に従って前記復号化ピクチャをデブロッキングし、
前記進められた次元に従ってデブロッキングされた前記復号化ピクチャからデブロッキング誤差を推定するステップと、

前記ベクトルの次元それぞれに対して算出された前記推定されたデブロッキング誤差を表す傾斜ベクトルに従って、次の反復における前記デブロッキングベクトルを変更するステップと

を有することを特徴とする方法。

【請求項 16】

前記推定された誤差が前記閾値を上回らない場合、符号化ピクチャデータ及び前記デブロッキングベクトルの識別子を復号器へ送信するステップを更に備えることを特徴とする請求項 15 に記載の方法。

【請求項 17】

前記デブロッグ誤差が所定の閾値を下回るまで前記方法を反復させるステップを更に備えることを特徴とする請求項 15 に記載の方法。

【請求項 18】

前記方法の 2 回目の反復において、
最初の反復と前記 2 回目の反復との間の推定された誤差の変化率を推定するステップと、

前記変化率が第 2 の閾値を下回る場合、符号化ピクチャデータ及び前記デブロッキングベクトルの識別子を復号器へ送信するステップと、
を更に備えることを特徴とする請求項 15 に記載の方法。

【請求項 19】

前記傾斜の大きさが第 2 の閾値を下回る場合、符号化ピクチャデータ及び前記デブロッキングベクトルの識別子を復号器へ送信するステップを更に備えることを特徴とする請求項 15 に記載の方法。

【請求項 20】

10

20

30

40

50

前記方法の後続の反復において、

前記変更されたデブロッキングベクトルが前記方法の前の反復において使用されたデブロッキングベクトルと一致するかを判定するステップと、

前記変更されたデブロッキングベクトルが前記方法の前の反復において使用されたデブロッキングベクトルと一致する場合、符号化ピクチャデータ及び前記デブロッキングベクトルの識別子を復号器へ送信するステップと、

を更に備えることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 1】

前記進めるステップは、前記次元の各々において前記デブロッキングベクトルを増分するステップを備えることを特徴とする請求項 1 5 に記載の方法。

10

【請求項 2 2】

前記進めることは、前記次元の各々において前記デブロッキングベクトルを先行の反復の前記傾斜により判定された方向に調整することを備えることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 3】

最初の反復の前記デブロッキングベクトルにはデフォルト値が設定されることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 4】

最初の反復の前記デブロッキングベクトルには、先行ピクチャに対して前記方法を実行することにより最終的に取得されたデブロッキングベクトルの値が設定されることを特徴とする請求項 1 5 に記載の方法。

20

【請求項 2 5】

最初の反復の前記デブロッキングベクトルには、現在のピクチャと共通の符号化の割り当てを有する先行ピクチャに対して前記方法を実行することにより最終的に取得されたデブロッキングベクトルの値が設定されることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 6】

最初の反復の前記デブロッキングベクトルには、現在のピクチャと同様の動き特性を有する先行ピクチャに対して前記方法を実行することにより最終的に取得されたデブロッキングベクトルの値が設定されることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 7】

30

前記デブロッキングピクチャの前記推定される誤差は、前記デブロッキングされた復号化ピクチャとフィルタリングされていないソースピクチャデータとの比較から判定されることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 8】

前記デブロッキングピクチャの前記推定される誤差は、前記デブロッキングされた復号化ピクチャとフィルタリングされたソースピクチャデータとの比較から判定されることを特徴とする請求項 1 5 に記載の方法。

【請求項 2 9】

ビデオ符号化装置であって、

ソースピクチャ及び予測ブロックに対する入力を有するブロックベース符号器と、

前記ブロックベース符号器に接続される予測器と、

ここで前記予測器は、

デブロッキングフィルタと反復傾斜利用探索処理によりデブロッキングパラメータを推定する制御論理とを含み、参照ピクチャの符号化ブロックを復号化するブロックベース復号器と、

デブロッキングピクチャに対する参照ピクチャ記憶装置とを含む；

前記符号化ピクチャのデータ及び前記最終的に推定されたデブロッキングパラメータを格納する送信バッファとを備え、

前記反復傾斜利用探索処理では、

デブロッキングベクトルに従って復号化ピクチャのデブロッキングを実行し、

50

ここで、前記デブロッキングベクトルは、それぞれが当該デブロッキングベクトルの各次元となるデブロッキングパラメータで表わされる；

前記デブロッキングされた復号化ピクチャからデブロッキング誤差を推定し、前記推定された誤差が所定の閾値を上回る場合には、ベクトルの次元によって表される各デブロッキングパラメータに対して、

前記次元の各々において前記デブロッキングベクトルを進め、前記進められた次元に従って前記復号化ピクチャをデブロッキングし、前記進められた次元に従ってデブロッキングされた前記復号化ピクチャからデブロッキング誤差を推定し、

前記ベクトルの次元それぞれに対して算出された前記推定されたデブロッキング誤差を表す傾斜ベクトルに従って、次の反復における前記デブロッキングベクトルを変更することをを行うことを特徴とするビデオ符号化装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、ビデオ符号化に関し、特に、動き補償符号化の一部として補間フィルタを使用するビデオ符号化システムに関する。

【背景技術】

【0002】

通常、ビデオコーデックは、元の静止画像用JPEG符号器に使用されるのと同様に、本明細書中で「画素ブロック」と呼ばれる画素のブロックに対して離散コサイン変換（「DCT」）を使用してビデオフレームを符号化する。最初のフレーム（「フレーム内（イントラ）」フレームと呼ばれる）は、独立フレームとして符号化、及び送信される。後続フレームは、シーン内のオブジェクトの小さい動きのためゆっくり変化するものとしてモデル化される、動き補償（「MC」）と呼ばれる技術を使用するフレーム間モードで効率的に符号化される。この場合、先に符号化されたフレームにおける位置からの画素ブロックの変位は、予測画素ブロックとソース画像からの画素ブロックとの間の差分の符号化表現と共に動きベクトルとして送信される。

【0003】

動き補償の簡単な説明を以下に示す。図1及び図2は、動き補償画像符号器/復号器システムを示すブロック図である。システムは、圧縮画像の記憶容量及び計算を減少すると同時に高度な圧縮及び適応性を付与するために、変換符号化（画素の画素ブロックのDCTの形態である）と予測符号化（差分パルス符号変調（「PCM」）の形態である）とを組み合わせる。変換領域において動き補償を実行することが困難であるため、フレーム間符号器における第1のステップは動き補償予測誤差を作成することである。この計算は、符号器及び復号器の双方において1つ以上のフレーム記憶装置を必要とする。結果として得られた誤差信号は、DCTを使用して変換され、適応量子化器により量子化され、可変長符号器（「VLC」）を使用してエントロピ符号化され、チャンネルを介する送信のためにバッファに入れられる。

【0004】

動き推定器は、図3に示すように動作する。最も単純な形態において、現在のフレームは、本明細書中で「mcブロック」と呼ばれる例えば16×16又は8×8の一定のサイズの動き補償ブロックに分割される。しかし、特に高度なビデオ符号化であるITU-T勧告H.264のH.264等の新規のコーデックにおいて、可変サイズのmcブロックが使用されることが多い。実際、非矩形のmcブロックも研究及び提案されてきた。一般に、mcブロックのサイズは画素ブロックのサイズ以上である。

【0005】

更に、動き補償の最も単純な形態において、図3に示すように、先行の復号化フレームが参照フレームとして使用される。しかし、特にH.264等の新規のコーデックにおいて、多くの可能な参照フレームのうちの1つが同様に使用されてもよい。実際、適切な信

10

20

30

40

50

号伝送を用いる場合、異なる参照フレームが各m cブロックに対して使用されてもよい。

【0006】

現在のフレーム内の各m cブロックは、参照フレーム内の変位m cブロックの集合と比較されることで、現在のm cブロックを最適に予測するものが決定される。最良のマッチングm cブロックが見つけれられた場合、参照m cブロックの変位を指定する動きベクトルが決定される。

【0007】

空間的冗長の利用

ビデオは連続する静止画像でもあるので、J P E Gと同様の技術を使用して何らかの圧縮を達成できる。そのような圧縮方法はフレーム内符号化技術と呼ばれ、ビデオの各フレームは個別に且つ独立して圧縮又は符号化される。フレーム内符号化は、フレームにおける隣接画素間に存在する空間的冗長を利用する。フレーム内符号化のみを使用して符号化されたフレームを「Iフレーム」と呼ばれる。

10

【0008】

時間的冗長の利用

「順方向予測」と呼ばれる上述の単方向の動き推定においては、符号化対象フレーム内の目標m cブロックは、「参照フレーム」と呼ばれる先行フレーム内の同一サイズのm cブロックの集合とマッチングされる。目標m cブロックと「最良のマッチング」参照フレーム内のm cブロックが参照m cブロックとして使用される。その後、予測誤差が目標m cブロックと参照m cブロックとの間の差分として計算される。一般に、予測m cブロックは参照フレーム内の符号化されたm cブロックの境界と一致しない。この最良のマッチング参照m cブロックの位置は、参照m cブロックと目標m cブロックとの間の変位を記述する動きベクトルにより示される。動きベクトル情報は同様に符号化され、予測誤差と共に送信される。順方向予測を使用して符号化されたフレームは「Pフレーム」と呼ばれる。

20

【0009】

予測誤差自体は、上述した概要におけるD C Tを用いるフレーム内符号化技術を使用して送信される。

【0010】

双方向時間的予測

「動き補償補間」とも呼ばれる双方向時間的予測は、現在のビデオコーデックの重要な特徴である。双方向予測を用いて符号化されるフレームは、通常は先行フレーム及び後続フレームである2つの参照フレームを使用する。しかし、特にH . 2 6 4等の新規のコーデックにおいて、多くの可能な参照フレームのうちの2つが同様に使用されてもよい。実際、適切な信号伝送を用いる場合、異なる参照フレームが各m cブロックに対して使用できる。

30

【0011】

双方向符号化されるフレーム内の目標m cブロックは、先行の参照フレームからのm cブロックにより予測されるか（順方向予測）、後続の参照フレームからのm cブロックにより予測されるか（逆方向予測）、あるいは各参照フレームから1つずつの2つのm cブロックの平均により予測される（補間）。いずれの場合も、参照フレームからの予測m cブロックが動きベクトルと関連付けられるため、各m cブロックに対して最大2つの動きベクトルが双方向予測の場合に使用される。双方向予測フレーム内のm cブロックに対する動き補償補間を図4に示す。双方向予測を使用して符号化されたフレームは「Bフレーム」と呼ばれる。

40

【0012】

双方向予測は多くの利点を提供する。主要な利点は、得られる圧縮が順方向（単方向）予測のみから得られる圧縮より通常は高いことである。同一のピクチャ品質を取得するために、双方向予測フレームは順方向予測のみを使用するフレームより少ないビットで符号化可能である。

50

【 0 0 1 3 】

しかし、フレームがシーケンス外で符号化される必要があるため、双方向予測では符号化処理において余分な遅延が発生する。更に、m c ブロックのマッチング（最も多くの計算を必要とする符号化手順）が目標 m c ブロック毎に先行の参照フレームに対して 1 回及び後続の参照フレームに対して 1 回の、2 回実行される必要があるため、符号化が更に複雑になる。

【 0 0 1 4 】

双方向予測に対する一般的な符号器アーキテクチャ

図 5 は、一般的な双方向ビデオ符号器を示す。フレームの再順序付けは符号化前に実行されると仮定する。すなわち、B フレームの予測に使用される I 又は P フレームは、対応する B フレームのいずれかの前に符号化及び送信される必要がある。このコーデックにおいて、B フレームは参照フレームとして使用されない。アーキテクチャを変更することにより、B フレームは H . 2 6 4 のように参照フレームとして使用可能である。

10

【 0 0 1 5 】

入力ビデオは、動き補償推定器 / 予測器に送られ、予測が減算器のマイナス入力に送られる。各 m c ブロックに対して、フレーム間 / フレーム内分類器は、入力画素と減算器の予測誤差出力とを比較する。通常、平均二乗予測誤差が平均二乗画素値を上回る場合、フレーム内 m c ブロックが決定される。画素及び予測誤差の双方の D C T を含む更に複雑な比較により性能が若干向上するが、これは通常はコストに見合うと考えられない。

【 0 0 1 6 】

フレーム内 m c ブロックの場合、予測は 0 に設定される。フレーム内 m c ブロック以外の場合、予測は上述のように予測器から入力される。予測誤差は D C T 及び量子化器を通過した後、符号化及び多重化されてバッファに送出される。

20

【 0 0 1 7 】

量子化されたレベルは、逆量子化器により再構成 D C T 係数に変換され、逆数は符号化された予測誤差を生成するために逆 D C T 部（「 I D C T 」）により変換される。加算器は、予測誤差に予測を加算し、符号化された画素値を生成するために結果を例えば 0 ~ 2 5 5 の範囲にクリップする。

【 0 0 1 8 】

B フレームの場合、動き補償推定器 / 予測器はピクチャ記憶装置に保持される先行フレーム及び後続フレームの双方を使用する。

30

【 0 0 1 9 】

I フレーム及び P フレームの場合、加算器により出力された符号化画素は後続ピクチャ記憶装置に書き込まれ、それと同時に、古い画素は後続ピクチャ記憶装置から先行ピクチャ記憶装置にコピーされる。実際は、これはメモリアドレスの単純な変更により通常は達成される。

【 0 0 2 0 】

また、実際は、符号化画素はピクチャ記憶装置に入力される前に適応デブロッキングフィルタによりフィルタリングされてもよい。これにより、特に符号化アーティファクトが可視になる低ビットレートに対して動き補償予測が向上される。

40

【 0 0 2 1 】

量子化アダプタと関連する符号化統計プロセッサは出力ビットレートを制御し、ピクチャの品質を可能な限り最適化する。

【 0 0 2 2 】

双方向予測に対する一般的な復号器アーキテクチャ

図 6 は、一般的な双方向ビデオ復号器を示す。これは、逆処理を使用する符号器の画素再構成部分に対応する構造を有する。フレームの再順序付けは復号化及びビデオ出力の後に行われると仮定する。補間フィルタは、符号器と同様に動き補償予測器の出力に対して配置される。

【 0 0 2 3 】

50

小数動きベクトルの変位

図3及び図4は、現在のフレーム内の復号化中の現在のmcブロックの位置に対して垂直及び水平方向に変位している参照フレーム内の参照mcブロックを示す。変位量は、動きベクトルと呼ばれる2次元ベクトル $[d_x, d_y]$ により表される。動きベクトルは符号化及び送信されてもよく、あるいは復号器内に既存の情報から推定されてもよい。その場合、動きベクトルは送信されない。双方向予測の場合、送信される各mcブロックは2つの動きベクトルを必要とする。

【0024】

最も単純な形態において、 d_x 及び d_y は、参照mcブロックを変位させるための水平方向の画素数及び垂直方向の行数を表す符号付き整数である。この場合、参照mcブロックは参照記憶装置から適切な画素を単に読み出すことにより取得される。

10

【0025】

しかし、新規のビデオコーデックにおいて、 d_x 及び d_y が小数値をとれるようにすることが有益であることが分かっている。一般に、そのようなビデオコーデックでは $1/4$ 画素、すなわち整数 ± 0.25 、 0.5 又は 0.75 までの変位精度が可能である。

【0026】

小数動きベクトルは、参照記憶装置からの画素の単純な読み出し以上の動作を必要とする。参照記憶装置の画素間の場所に対する参照mcブロック値を取得するために、それらの間を補間する必要がある。

【0027】

単純な双線形補間はかなり適切に動作する。しかし、実際は、特にこの目的のために設計された2次元補間フィルタを使用するのが有益であることが分かっている。実際、性能及び実用上の理由で、フィルタはシフト不変フィルタではない場合が多い。その代わり、小数動きベクトルの異なる値が異なる補間フィルタを利用してよい。

20

【0028】

適応補間フィルタを使用する動き補償

最適な動き補償補間フィルタは多くの要因に依存する。例えば、シーン内のオブジェクトは純粋な平行移動をしない場合がある。オブジェクトは2次元及び3次元の双方において回転する場合がある。他の要因は、ズーム、カメラの動き、並びに影又は種々の照明により生じる光の変動を含む。

30

【0029】

カメラの特徴は、それらのセンサの特定の特性により変化する。例えば、多くのコンシューマ向けカメラは本来インターレースを施され、それらの出力はインターレース解除及びフィルタリングされ、インターレースアーティファクトが存在しない高画質のピクチャが提供される。低輝度状態によりフレーム毎の露光時間が増加した場合、移動オブジェクトの動きに依存するばけが生じる。画素は正方形でなくてもよい。

【0030】

従って、多くの例において、動き補償補間フィルタが上記の外的要因及び他の外的要因に適応する場合、性能が向上される。そのようなシステムにおいて、補間フィルタは、各フレームにわたり現在のmcブロックと対応する参照mcブロックとの間の平均二乗誤差を最小化することにより設計される。これらは、いわゆるウィナーフィルタ(Wiener filter)である。その場合、フィルタ係数は、実際の動き補償符号化において使用される各フレームの最初に量子化され且つ送信される。H.264及び同様のコーデックにおいて、少数のデブロッキングパラメータのみが定期的に調整可能である。更に、フィルタ動作が非線形であるため、通常のウィナーフィルタ設計を適用できない。

40

【0031】

従って、符号化中にデブロッキングフィルタのパラメータを選択する効率的な方式が当分野において必要とされる。

【図面の簡単な説明】

【0032】

50

【図 1】図 1 は、従来のビデオ符号器を示すブロック図である。

【図 2】図 2 は、従来のビデオ復号器を示すブロック図である。

【図 3】図 3 は、動き補償予測の原理を示す図である。

【図 4】図 4 は、双方向時間的予測の原理を示す図である。

【図 5】図 5 は、従来の双方向ビデオ符号器を示すブロック図である。

【図 6】図 6 は、従来の双方向ビデオ復号器を示すブロック図である。

【図 7】図 7 は、本発明の実施形態と共に使用するのに適した符号器 / 復号器システムを示す図である。

【図 8】図 8 は、本発明の一実施形態に係るビデオ符号器を概略的に示すブロック図である。

10

【図 9】図 9 は、本発明の一実施形態に係る方法を示す図である。

【図 10】図 10 は、本発明の一実施形態に係る傾斜探索方法の動作例を示す図である。

【図 11】図 11 は、本発明の別の実施形態に係る方法を示す図である。

【図 12】図 12 は、本発明の別の実施形態に係る傾斜探索方法の動作例を示す図である。

【図 13】図 13 は、本発明の一実施形態に係るビデオ復号器を概略的に示すブロック図である。

【発明を実施するための形態】

【0033】

本発明の実施形態は、符号化ビデオに対するデブロッキングパラメータを選択するための反復方法を提供する。この方法によると、復号化ピクチャは多次元デブロッキングベクトルに関連するパラメータに従ってデブロッキングされ、「デブロッキング誤差」と呼ばれる誤差がそれから推定される。推定された誤差が所定の閾値を上回る場合、各々が各次元において現在のデブロッキングベクトルから進められた別のデブロッキングベクトルが作成される。方法は、進められた各ベクトルに従って復号化ピクチャをデブロッキングし、進められた各ベクトルのデブロッキングの各々から誤差を推定する。最後に、デブロッキングベクトルは、ベクトルの次元の推定されたデブロッキング誤差から導出される傾斜に従って、次の反復に対して変更される。この傾斜利用探索方法は、効率的な方法でデブロッキングパラメータの最終的な集合に収束する。

20

【0034】

図 7 は、本発明と共に使用するのに適した符号器 / 復号器システム 100 を示す。図中、ネットワーク 130 を介して復号器 120 と通信する符号器 110 が提供される。符号器 110 は、カメラ装置を介して符号器においてローカルに取り込まれるか又は記憶装置（不図示）から検索されるソースビデオのデータストリームに対して符号化動作を実行する。符号化動作はソースビデオデータの帯域幅を減少し、そこから符号化ビデオを生成する。符号器 110 は、ネットワーク 130 を介して復号器 120 へ符号化ビデオを送信する。復号器 120 は、符号器 110 により実行された符号化動作を反転して、符号化ビデオデータから回復ビデオデータストリームを生成する。通常、符号器 110 により実行される符号化動作は不可逆処理であるため、回復ビデオデータはソースビデオデータの不正確な複製となる。復号器 120 は、表示装置上に回復ビデオデータをレンダリングするか、あるいは後で使用するために回復ビデオデータを格納する。

30

40

【0035】

図示するように、ネットワーク 130 は符号器 110 から復号器 120 に符号化ビデオデータを転送する。ネットワーク 130 は、1 つ又は複数の有線又は無線通信ネットワーク、コンピュータネットワーク、あるいはそれらの組み合わせとして提供される。更に、ネットワーク 130 は、電気記憶装置、光学記憶装置又は磁気記憶装置等の記憶ユニットとして提供されてもよい。

【0036】

図 8 は、本発明と共に使用するのに適した符号器を概略的に示すブロック図である。符号器 200 は、ブロックベース符号化チェーン 210 及び予測部 220 を含む。

50

【 0 0 3 7 】

ブロックベース符号化チェーン 2 1 0 は、減算器 2 1 2、変換部 2 1 4、量子化器 2 1 6 及び可変長符号器 2 1 8 を含む。減算器 2 1 2 は、ソース画像から入力 m c ブロックを受信し、予測部 2 2 0 から予測 m c ブロックを受信する。減算器 2 1 2 は、入力 m c ブロックから予測 m c ブロックを減算して残余画素のブロックを生成する。変換部 2 1 4 は、通常は離散コサイン変換（「DCT」）又はウェーブレット変換である空間変換に従って、m c ブロックの残余データを変換係数のアレイに変換する。量子化器 2 1 6 は、量子化パラメータ（「QP」）に従って各ブロックの変換係数を切り縮める（turncate）。切り縮めに使用された QP 値は、チャンネルで復号器へ送信される。可変長符号器 2 1 8 は、例えば可変長符号化アルゴリズムであるエントロピ符号化アルゴリズムに従って、量子化された係数を符号化する。可変長符号化の後、各 m c ブロックの符号化データはバッファ 2 4 0 に格納され、チャンネルを介する復号器への送信を待機する。

10

【 0 0 3 8 】

予測部 2 2 0 は、逆量子化部 2 2 2、逆変換部 2 2 4、加算器 2 2 6、デブロッキングフィルタ 2 2 8、参照ピクチャキャッシュ 2 3 0、動き補償予測器 2 3 2 及び動き推定器 2 3 4 を含む。逆量子化部 2 2 2 は、量子化器 2 1 6 により使用された QP に従って符号化ビデオデータを量子化する。逆変換部 2 2 4 は、再量子化された係数を画素領域に変換する。加算器 2 2 6 は、逆変換部 2 2 4 から出力された残余画素と動き補償予測器 2 3 2 からの予測動きデータとを加算する。デブロッキングフィルタ 2 2 8 は、同一フレームの回復 m c ブロックと他の回復 m c ブロックとの間の継ぎ目の回復画像データをフィルタリングする。参照ピクチャキャッシュ 2 3 0 は、後で受信される m c ブロックの符号化中に参照フレームとして使用するために回復フレームを格納する。

20

【 0 0 3 9 】

動き補償予測器 2 3 2 は、ブロック符号器 2 1 0 が使用する予測 m c ブロックを生成する。動き推定器 2 3 4 は、符号化中のソース画像と参照ピクチャキャッシュ 2 3 0 に格納された参照フレームとの間の画像の動きを推定する。動き推定器 2 3 4 は使用される推定モード（例えば、単方向 P 符号化又は双方向 B 符号化）を選択し、そのような予測符号化で使用する動きベクトルを生成する。動きベクトルは動き補償予測器 2 3 2 及びチャンネルに出力される。それに応答して、動き補償予測器 2 3 2 は参照ピクチャキャッシュ 2 3 0 から予測 m c ブロックを検索し、ブロック符号器 2 1 0 に予測ブロックを出力する。オプションとして、動き補償予測器 2 3 2 は、ブロック符号器 2 1 0 に出力する前に、検索された m c ブロックに対して補間フィルタリングを実行する（不図示）。

30

【 0 0 4 0 】

デブロッキングフィルタ 2 2 8 の動作は制御パラメータに基づいて変動する。符号器において、制御論理（不図示）は、予測部により生成された符号化画像データを再検討し、符号化画像データとソース画像のデータとを比較して、符号化誤差を最小化するパラメータを判定する。一実施形態において、選択処理は、パラメータの適切な集合に迅速に収束するための候補パラメータ値を介する傾斜探索処理により強化される。適切なパラメータが選択されると、パラメータ識別子は復号化において使用するために復号器へ送信される。例えばデブロッキングパラメータは、ピクチャ内の m c ブロック又はより大きいエンティティ（例えば、マクロブロック又はスライス）毎に送信される。デブロッキングパラメータは、符号化 m c ブロックの他のデータ（例えば、動きベクトル、量子化パラメータ及び符号化された残余）と共に送信される。

40

【 0 0 4 1 】

図 9 は、本発明の一実施形態に係る方法 3 0 0 を示す。この方法 3 0 0 は、ソースピクチャの符号化及び復号化から開始し（ブロック 3 1 0）、現在のデブロッキングベクトルに関連するデブロッキングパラメータを使用してピクチャに対してデブロッキングを実行する（ブロック 3 2 0）。方法 3 0 0 の最初の反復において、ベクトルはデフォルト値に設定される。方法 3 0 0 は、デブロッキングピクチャに関連する誤差を計算し（ブロック 3 3 0）、誤差と所定の閾値とを比較する（ブロック 3 4 0）。誤差が所定の閾値を下回

50

る場合、方法300は符号化ピクチャデータ及び現在のデブロッキングベクトルを復号器へ送信し(ブロック350)、現在のピクチャに対する処理を終了する。

【0042】

デブロッキングパラメータは、複数の値を有するN次元ベクトルを表すと考えられる。例えばH.264は、画像符号化において符号器により定義される2つのパラメータ、すなわちslice_alpha_c0_offset_div2及びslice_beta_offset_div2(本明細書中で「 α 」及び「 β 」と呼ぶ)を定義する。H.264の実現例に適用される場合、デブロッキングパラメータは2次元ベクトルであると考えられる。図9の方法300において考えられる場合、将来の符号化プロトコルは、3次元空間、4次元空間又は5次元以上の空間として表される更なる種類のデブロッキングパラメータを定義してもよい。

10

【0043】

ブロック340の比較において誤差が閾値を上回ると判定される場合、方法300は新規のデブロッキングベクトルに対する多次元探索を実行する。詳細には、方法300は各次元iにおいて個別に現在のベクトルを増分する(ブロック360)。方法300は、増分された各ベクトルに関連するパラメータを使用して復号化ピクチャのデブロッキングを実行し(ブロック370)、元のピクチャとデブロッキングピクチャとの間の誤差を計算する(ブロック380)。N次元ベクトル空間において、N個のデブロッキングピクチャ及びそれらから取得されるN個の誤差値が存在する。方法300は、ブロック330で取得された誤差値とブロック380で各次元の計算から取得された誤差とを比較することにより傾斜値(gradient value)を計算する(ブロック390)。方法300は、傾斜から新規のベクトルを生成し(ブロック400)、ブロック320に戻って解析の別の反復を実行する。

20

【0044】

傾斜ベクトル(gradient vector)はN次元ベクトルであり、各次元iの成分はブロック330からのデブロッキング誤差とブロック380から取得された次元誤差との比較により取得される。例えば傾斜ベクトルGは、傾斜ベクトルの各成分G_iに対して以下のように計算される：

$$G = [G_1, G_2, \dots, G_N]$$

式中、 $G_i = \text{err}_i - \text{err}_{\text{DBLK}}$

30

方法300は、以下のようにして新規のデブロッキングベクトルVを更に生成する：

【0045】

【数1】

$$V_{\text{new}} = V_{\text{old}} - \frac{w * G}{|G|}$$

式中、wはアルゴリズムのステップサイズを表し、各次元成分V_iは整数値を有するように丸められる。ステップサイズwはプログラム可能値であって、通常は各次元におけるデブロッキングベクトル空間のサイズに基づいて設定される。

40

【0046】

H.264の実現例に適用される場合、例えば方法300は $G = [\text{err}_{\text{LK}} - \text{err}_{\text{DBLK}}, \text{err}_{\text{LK}} - \text{err}_{\text{DBLK}}]$ として傾斜ベクトルGを生成する。式中、 err_{LK} 及び err_{DBLK} はそれぞれ、ブロック380により取得される次元及び次元における誤差値である。方法300は、以下のようにして新規のデブロッキングベクトル $[\quad , \quad]_{\text{new}}$ を更に生成する：

【0047】

【数 2】

$$[\alpha, \beta]_{new} = [\alpha, \beta]_{old} - \frac{w * G}{|G|}$$

式中、 $[\alpha, \beta]_{new}$ は整数値を有するように同様に丸められる。実際は、 $w = 1 \sim 4$ の値の場合に満足な結果が得られる。

【0048】

動作中、反復を数回行うことにより、図9の傾斜探索方法300が閾値を下回る誤差を生成する(ブロック340)デブロッキングパラメータの集合に収束することが予想される。方法300の他の実施形態は、誤差が十分に小さい状態が達成される前に動作を終了するための他のテストを含む。

10

【0049】

例えば符号化ピクチャが復号化された後(ブロック310)、方法300は復号化誤差を計算し、それと別の閾値とを比較する(ブロック410)。復号化誤差が閾値を下回る場合、方法300はデブロッキングパラメータを伴わずに符号化ピクチャを復号器へ送信する(ブロック420)。そのようなテストは適切な画質を達成するためにデブロッキングが不要であると有利に判定するため、ブロック320～400の動作を実行するために必要な処理リソースが節約される。

20

【0050】

別の実施形態において、ブロック340で判定された誤差が閾値を上回る場合、方法300は現在の反復と先行の反復との間の誤差の変化(「誤差」と示す)を更に判定する。最初の反復の実行時、現在の誤差値及びブロック410で取得された復号化誤差値から誤差を導出してもよい。あるいは、最初の反復においてブロック420は省略される。方法300は、誤差の値と別の閾値とを比較する(ブロック430)。誤差の値が所定の閾値を上回る場合、これは、傾斜探索方法の収束率が方法300の更なる実行に値するほど十分に画像の改善を向上する可能性が低いことを示す。この場合、方法300はブロック350へ進み、符号化ピクチャのデータ及びデブロッキングベクトルを復号器へ送信する。

30

【0051】

更なる一実施形態において、方法300が新規のベクトルを設定した後(ブロック400)、方法300は新規のベクトルが方法の何らかの前の反復において使用されたかを判定する(ブロック440)。新規のベクトルが前の反復において使用された場合、これは、方法300の更なる実行が現在の反復までに取得された結果以上に画像の改善を向上する可能性が低いことを示す。この場合、方法300はブロック350へ進み、符号化ピクチャのデータ及びブロック400の動作の前に取得されたデブロッキングベクトルを復号器へ送信する。

【0052】

更に別の実施形態において、ブロック350へ進むほど十分に小さい誤差が生成されずにブロック320～400の反復が所定の回数実行された場合には、方法300は終了する。

40

【0053】

多くの実現例において、デブロッキングパラメータの各次元の可能な値の範囲は有限である。例えばH.264の実現例において、値及び値はそれぞれ-6～6の値を有する整数である必要がある。一実施形態において、特定の次元iにおいてベクトルを増分する処理(ブロック360)により増分された値が正当な値を超過する場合(例えば、=7)、増分された値は制限値に設定される必要があり(=6)、各次元に関連するブロック370及び380の処理は省略される。その場合、各次元に関連する誤差iの値はブロック330で導出される誤差値に等しいと推定される。

50

【 0 0 5 4 】

更なる一実施形態において、方法 3 0 0 が誤差値から傾斜を計算する場合（ブロック 3 9 0）、方法 3 0 0 はブロック 3 3 0 で取得されたデブロッキング誤差に対する傾斜の大きさの比率を更に計算し、比率と別の閾値とを比較する（ブロック 4 5 0）。比率が閾値を下回る場合、方法 3 0 0 はブロック 3 5 0 へ進み、符号化ピクチャのデータ及びブロック 3 2 0 で使用されたデブロッキングベクトルを復号器へ送信する。

【 0 0 5 5 】

実現例において、ブロック 3 4 0、4 1 0 及び 4 3 0 ~ 4 5 0 における比較に使用される閾値は個々の必要性に適するように調整されるプログラム可能値である。閾値は静的な事前プログラム値として設定される。あるいは、それらは動作中に動的に変化する。例えば閾値は、各ピクチャに割り当てられた符号化動作の種類（例えば、I、P 又は B 符号化）に基づいて変動してもよく、あるいはピクチャを符号化するために使用された量子化パラメータに基づいて変動してもよい。閾値は、符号器における符号化動作を支配した目標ビットレートの割り当てに基づいて変動してもよい。更に、閾値は、ビデオデータが符号化される対象である例えば大画面装置（ラップトップ、タブレットコンピュータ）又は小画面装置（ポータブルメディアプレーヤ、スマートフォン）である目標復号器を記述するデータに基づいて変動してもよい。

【 0 0 5 6 】

図 1 0 は、デブロッキングマトリクス of の例に実行される図 9 の傾斜探索方法の動作を示す。図示されるように、デブロッキングマトリクスは、値、により指標付けされる 13×13 の 2 次元マトリクスである。本例は、H. 2 6 4 の実現例において使用されるデブロッキングマトリクスに対応する。値及び値により指標付けされる各セルの場所は、復号器により適用されるデブロッキングパラメータの集合に関連する。動作中、符号器は及びの識別子を送信し、復号器は格納されたマトリクスからデブロッキングフィルタパラメータを検索するためにそれらを使用する。

【 0 0 5 7 】

方法の最初の反復において、デブロッキングベクトル 5 1 0 は例えば $[0, 0]$ であるデフォルト値に設定される。第 1 のデブロッキング動作は、デブロッキングベクトル 5 1 0 に関連するパラメータを使用して実行される。それから取得される誤差が方法を終了するのに不十分である場合、デブロッキングベクトルは各次元において増分される。2 次元ベクトル空間において、これにより一対の増分されたベクトル 5 2 0、5 3 0 が得られる。増分された各ベクトル 5 2 0、5 3 0 に関連するパラメータをそれぞれ使用する一対のデブロッキング動作が実行される。傾斜ベクトル G は、一対のデブロッキング動作に関連する誤差値及び元のデブロッキングベクトル 5 1 0 から取得された誤差値から導出される。その後、傾斜ベクトル G から導出される更新されたデブロッキングベクトル 5 4 0 を使用して別の反復が実行され、そこから取得された誤差が方法の動作を終了するのに不十分である場合、別の対の増分されたベクトル 5 5 0、5 6 0 が計算される。停止点に到達するまで動作は続行し、新規のデブロッキングベクトル、増分されたデブロッキングベクトル及び傾斜ベクトルを生成する（不図示）。

【 0 0 5 8 】

2 次元の例における図 9 の方法の動作は、以下の処理フローにより実行される。

現在のピクチャを符号化する。

現在のピクチャを復号化する。

E 0 デブロッキングされていない復号化ピクチャの M S E（現在のピクチャ - 復号化ピクチャ）。

I f E 0 < t 1 の場合：

送信する：

ピクチャの圧縮ビット及び

デブロッキングパラメータ（デブロッキングが行われていないことを示す）。

後続ピクチャの動き補償符号化における参照として、参照ピクチャキャッシュにデブ

10

20

30

40

50

ロッキングされていない復号化ピクチャを格納する，
 現在のピクチャに対する処理を終了する．

E l s e ，
 デブロッキングが行われていないことを示すために、デブロッキングパラメータベクトル $[A 0 , B 0] = [* , *]$ を設定する．
 デブロッキングパラメータベクトル $[A 1 , B 1] = [0 , 0]$ を設定する．
 ピクチャを保持するために一時格納 T S 0 及び T S 1 を確立する．
 T S 0 デブロッキングされていない復号化ピクチャ．

P T . 1 : ベクトル $[A 1 , B 1]$ から取得されたパラメータを用いて復号化ピクチャを
 デブロッキングする． 10
 T S 1 デブロッキングピクチャ
 元のピクチャとデブロッキングピクチャとの間の平均二乗誤差 E 1 を計算する．

I f E 1 < t e :
 送信する：
 ピクチャの圧縮ビット及び
 デブロッキングパラメータ A 1 及び B 1 ．
 参照ピクチャキャッシュに T S 1 を格納する．
 現在のピクチャに対する処理を終了する． 20

E l s e , i f E 1 / E 0 > t i :
 送信する：
 ピクチャの圧縮ビット及び
 デブロッキングパラメータ A 0 及び B 0 ．
 参照ピクチャキャッシュに T S 0 を格納する．
 現在のピクチャに対する処理を終了する．

I f $[A 1 + 1 , B 1] = [A 0 , B 0]$ の場合、 E 2 E 0 を設定する．
 E l s e , E 2 E 1 を設定する．

I f $[A 1 + 1 , B 1] \neq [A 0 , B 0]$ 、且つ、 $A 1 < 6$: 30
 ベクトル $[A 1 + 1 , B 1]$ から取得されたパラメータを使用して、復号化ピクチャ
 をデブロッキングする．
 元のピクチャとデブロッキングピクチャとの間の平均二乗誤差 E 2 を計算する．

I f $[A 1 , B 1 + 1] = [A 0 , B 0]$ E 3 E 0 を設定する．
 E l s e , E 3 E 1 を設定する．

I f $[A 1 , B 1 + 1] \neq [A 0 , B 0]$ 且つ $B 1 < 6$:
 ベクトル $[A 1 , B 1 + 1]$ から取得されたパラメータを使用して、復号化ピクチャ
 をデブロッキングする． 40
 元のピクチャとデブロッキングピクチャとの間の平均二乗誤差 E 3 を計算する．

$G = [E 2 - E 1 , E 3 - E 1]$ として傾斜ベクトル G を計算する．

I f $| G | / E 1 < t s$:
 送信する：
 ピクチャの圧縮ビット及び
 デブロッキングパラメータ A 1 及び B 1 ．
 基準ピクチャキャッシュに T S 1 を格納する．
 現在のピクチャに対する処理を終了する． 50

E l s e ,

デブロッキングパラメータベクトル $[A 0 , B 0] = [A 1 , B 1]$ を設定する.

$E 0 = E 1$ を設定する.

$T S 0 \quad T S 1$ を設定する.

履歴テーブルに $[A 1 , B 1]$ を格納する.

新規のデブロッキングパラメータベクトル $[A 1 , B 1] \quad [A 1 , B 1] - w \times G / | G |$ を計算する。ここで、新規の $A 1$ 及び $B 1$ は範囲 $[- 6 , 6]$ 内の整数に丸められる。

【 0 0 5 9 】

I f $[A 1 , B 1]$ が履歴テーブルと一致する :

送信する :

ピクチャの圧縮ビット及び

デブロッキングパラメータ $A 0$ 及び $B 0$.

基準ピクチャキャッシュに $T S 0$ を格納する.

現在のピクチャに対する処理を終了する.

E l s e

新規のベクトル $[A 1 , B 1]$ を用いて、 $P T . 1$ から開始する次の反復を実行する.

【 0 0 6 0 】

図 1 1 は、本発明の一実施形態に係る方法 6 0 0 を示す。方法 6 0 0 は、ソースピクチャの符号化及び復号化から開始し (ブロック 6 1 0) 、現在のデブロッキングベクトルに関連するデブロッキングパラメータを使用してピクチャに対してデブロッキングを実行する (ブロック 6 2 0) 。方法 6 0 0 の最初の反復において、ベクトルはデフォルト値に設定される。方法 6 0 0 は、デブロッキングピクチャに関連する誤差を計算し (ブロック 6 3 0) 、誤差と所定の閾値とを比較する (ブロック 6 4 0) 。誤差が所定の閾値を下回る場合、方法 6 0 0 は符号化ピクチャデータ及び現在のデブロッキングベクトルを復号器へ送信し (ブロック 6 5 0) 、現在のピクチャに対する処理を終了する。

【 0 0 6 1 】

デブロッキングベクトルは、複数の値を有する N 次元ベクトルを表すと考えられる。図 9 の実施形態と同様に、 $H . 2 6 4$ の実現例に適用される場合、パラメータ及びパラメータは 2 次元ベクトルであると考えられる。図 1 1 の方法 6 0 0 において考えられる場合、将来の符号化プロトコルは、3 次元空間、4 次元空間又は 5 次元以上の空間として表される更なる種類のデブロッキングパラメータを定義してもよい。

【 0 0 6 2 】

ブロック 6 4 0 の比較において誤差が閾値を上回ると判定される場合、方法 6 0 0 は新規のデブロッキングベクトルに対する多次元探索を実行する。詳細には、方法 6 0 0 は次元 i に対する支配的探索方向に従って各次元 i において個別に現在のベクトルを進める (ブロック 6 6 0) 。方法 6 0 0 は、進められた各ベクトルに関連するパラメータを使用して復号化ピクチャのデブロッキングを実行し (ブロック 6 7 0) 、元のピクチャとデブロッキングピクチャとの間の誤差を計算する (ブロック 6 8 0) 。 N 次元ベクトル空間において、 N 個のデブロッキングピクチャ及びそれらから取得される N 個の誤差値が存在する。方法 6 0 0 は、ブロック 6 3 0 で取得された誤差値とブロック 6 8 0 で各次元の計算から取得された誤差とを比較することにより傾斜値を計算する (ブロック 6 9 0) 。方法 6 0 0 は、傾斜から新規のベクトルを生成し (ブロック 7 0 0) 、ブロック 6 2 0 に戻って解析の別の反復を実行する。

【 0 0 6 3 】

傾斜ベクトルは N 次元ベクトルであり、各次元 i の成分はブロック 6 3 0 からのデブロッキング誤差とブロック 6 8 0 から取得された次元誤差との比較により取得される。例え

10

20

30

40

50

ば傾斜ベクトルGは、傾斜ベクトルの各成分G_iに対して以下のように計算される：

$$G = [G_1, G_2, \dots, G_N]$$

式中、G_i = (err_i - err_{DBLK}) · DIR_i。上記の式において、DIR_iは各次元に対する支配的探索方向を表し、値 - 1 又は 1 を有する。

【0064】

方法600は、以下のようにして新規のデブロッキングベクトルVを更に生成する：

【0065】

【数3】

$$V_{new} = V_{old} - \frac{w * G}{|G|} \quad 10$$

式中、wはアルゴリズムのステップサイズを表し、各次元成分V_iは整数値を有するように丸められる。ステップサイズwは、通常は各次元におけるデブロッキングベクトル空間のサイズに基づいて設定されるプログラブル値である。

【0066】

H.264の実現例に適用される場合、例えば方法600はG = [(err₁ - err_{DBLK}) · DIR₁, (err₂ - err_{DBLK}) · DIR₂]として傾斜ベクトルGを生成する。式中、err₁及びerr₂はそれぞれ、ブロック380により取得される次元及び次元における誤差値であり、DIR₁及びDIR₂はそれぞれ次元及び次元における探索方向の指標を表す。方法600は、以下のようにして新規のデブロッキングベクトル[α, β]_{new}を更に生成する：

【0067】

【数4】

$$[\alpha, \beta]_{new} = [\alpha, \beta]_{old} - \frac{w * G}{|G|} \quad 30$$

式中、[α, β]_{new}は整数値を有するように同様に丸められる。実際は、w = 1.4の値の場合に満足な結果が得られる。

【0068】

動作中、反復を数回行うことにより、図11の傾斜探索方法600が閾値を下回る誤差を生成する(ブロック640)デブロッキングパラメータの集合に収束することが予想される。方法600の他の実施形態は、誤差が十分に小さい状態が達成される前に動作を終了するための他のテストを含む。

【0069】

例えば符号化ピクチャが復号化された後(ブロック610)、方法600は復号化誤差を計算し、それと別の閾値とを比較する(ブロック710)。復号化誤差が閾値を下回る場合、方法600はデブロッキングパラメータを伴わずに符号化ピクチャを復号器へ送信する(ブロック720)。そのようなテストは適切な画質を達成するためにデブロッキングが不要であると有利に判定するため、ブロック620~700の動作を実行するために必要な処理リソースが節約される。

【0070】

別の実施形態において、ブロック640で判定された誤差が閾値を上回る場合、方法600は現在の反復と先行の反復との間の誤差の変化(「誤差」と示す)を更に判定する。最初の反復の実行時、現在の誤差値及びブロック710で取得された復号化誤差値から誤差を導出してよい。あるいは、最初の反復においてブロック720は省略される。方法600は、誤差の値と別の閾値とを比較する(ブロック730)。誤差の値が所

定の閾値を上回る場合、これは傾斜探索方法の収束率が方法 600 の更なる実行に値するほど十分に画像の改善を向上する可能性が低いことを示す。この場合、方法 600 はブロック 650 へ進み、符号化ピクチャのデータ及びデブロッキングベクトルを復号器へ送信する。

【0071】

更なる一実施形態において、方法 600 が新規のベクトルを設定した後（ブロック 700）、方法 600 は新規のベクトルが方法の何らかの前の反復において使用されたかを判定する（ブロック 740）。新規のベクトルが前の反復において使用された場合、これは、方法 600 の更なる実行が現在の反復までに取得された結果以上に画像の改善を向上する可能性が低いことを示す。この場合、方法 600 はブロック 650 へ進み、符号化ピクチャのデータ及びブロック 700 の動作の前に取得されたデブロッキングベクトルを復号器へ送信する。

10

【0072】

更に別の実施形態において、ブロック 650 へ進むほど十分に小さい誤差が生成されずにブロック 620 ~ 700 の反復が所定の回数実行された場合、方法 600 は終了する。

【0073】

多くの実現例において、デブロッキングパラメータの各次元の可能な値の範囲は有限である。例えば H.264 の実現例において、値及び値はそれぞれ -6 ~ 6 の値を有する整数である必要がある。一実施形態において、特定の次元 i においてベクトルを進める処理（ブロック 660）により進められた値が正当な値を超える場合（例えば、 $= -7$ ）、増分された値は制限値に設定される必要があり（ $= -6$ ）、各次元に関連するブロック 670 及び 680 の処理は省略される。その場合、各次元に関連する誤差 i の値はブロック 630 で導出される誤差値に等しいと推定される。

20

【0074】

更なる一実施形態において、方法 600 が誤差値から傾斜を計算する場合（ブロック 690）、方法 600 はブロック 630 で取得されたデブロッキング誤差に対する傾斜の大きさの比率を更に計算し、比率と別の閾値とを比較する（ブロック 750）。比率が閾値を下回る場合、方法 600 はブロック 650 へ進み、符号化ピクチャのデータ及びブロック 620 で使用されたデブロッキングベクトルを復号器へ送信する。

【0075】

実現例において、ブロック 640、710 及び 730 ~ 750 における比較に使用される閾値は個々の必要性に適するように調整されるプログラム可能な値である。閾値は静的な事前プログラム値として設定される。あるいは、それらは動作中に動的に変動する。例えば閾値は、各ピクチャに割り当てられた符号化動作の種類（例えば、I、P 又は B 符号化）に基づいて変動してもよく、あるいはピクチャを符号化するために使用された量子化パラメータに基づいて変動してもよい。閾値は、符号器における符号化動作を支配した目標ビットレートの割り当てに基づいて変動してもよい。更に、閾値は、ビデオデータが符号化される対象である例えば大画面装置（ラップトップ、タブレットコンピュータ）又は小画面装置（ポータブルメディアプレーヤ、スマートフォン）である目標復号器を記述するデータに基づいて変動してもよい。

30

40

【0076】

図 12 は、デブロッキングマトリクスの例に実行される図 11 の傾斜探索方法の動作を示す。図示されるように、デブロッキングマトリクスは、値、により指標付けされる 13×13 の 2 次元マトリクスである。本例は、H.264 の実現例において使用されるデブロッキングマトリクスに対応する。値及び値により指標付けされる各セルの場所は、復号器により適用されるデブロッキングパラメータの集合に関連する。動作中、符号器は及びの識別子を送信し、復号器は格納されたマトリクスからデブロッキングフィルタパラメータを検索するためにそれらを使用する。

【0077】

方法の最初の反復において、デブロッキングベクトル 810 は例えば $[0, 0]$ である

50

デフォルト値に設定される。第 1 のデブロッキング動作は、デブロッキングベクトル 8 1 0 に関連するパラメータを使用して実行される。それから取得される誤差が方法を終了するのに不十分である場合、デブロッキングベクトルはデフォルト方向に従って各次元において進められる。2 次元ベクトル空間において、これにより一対の増分されたベクトル 8 2 0、8 3 0 が得られる。増分された各ベクトル 8 2 0、8 3 0 に関連するパラメータをそれぞれ使用する一対のデブロッキング動作が実行される。傾斜ベクトル G は、一対のデブロッキング動作に関連する誤差値及び元のデブロッキングベクトル 8 1 0 から取得された誤差値から導出される。その後、傾斜ベクトル G から導出される更新されたデブロッキングベクトル 8 4 0 を使用して別の反復が実行され、そこから取得された誤差が方法の動作を終了するのに不十分である場合、別の対の進められたベクトル 8 5 0、8 6 0 が計算される。2 回目の反復において、次元ベクトル 8 5 0、8 6 0 は、最初の反復の傾斜ベクトル G の負数に対応する方向に進められる。停止点に到達するまで動作は続行し、新規のデブロッキングベクトル、増分されたデブロッキングベクトル及び傾斜ベクトルを生成する（不図示）。

10

【 0 0 7 8 】

2 次元の例における図 1 1 の方法の動作は、以下の処理フローにより実行される。

現在のピクチャを符号化する。

現在のピクチャを復号化する。

E 0 デブロッキングされていない復号化ピクチャの M S E (現在のピクチャ - 復号化ピクチャ)。

20

I f E 0 < t 1 :

送信する :

ピクチャの圧縮ビット及び

デブロッキングパラメータ (デブロッキングが行われていないことを示す)。

後続ピクチャの動き補償符号化における基準として、基準ピクチャキャッシュにデブロッキングされていない復号化ピクチャを格納する。

現在のピクチャに対する処理を終了する。

E l s e ,

デブロッキングが行われていないことを示すために、デブロッキングパラメータベクトル [A 0 , B 0] = [* , *] を設定する。

30

デブロッキングパラメータベクトル [A 1 , B 1] = [0 , 0] を設定する。

ピクチャを保持するために一時格納 T S 0 及び T S 1 を確立する。

傾斜係数 G A = 1 及び G B = 1 を設定する。

T S 0 デブロッキングされていない復号化ピクチャ。

P T . 1 : ベクトル [A 1 , B 1] から取得されたパラメータを用いて復号化ピクチャをデブロッキングする。

T S 1 デブロッキングピクチャ。

元のピクチャとデブロッキングピクチャとの間の平均二乗誤差 E 1 を計算する。

40

I f E 1 < t e :

送信する :

ピクチャの圧縮ビット及び

デブロッキングパラメータ A 1 及び B 1 .

参照ピクチャキャッシュに T S 1 を格納する。

現在のピクチャに対する処理を終了する。

E l s e , i f E 1 / E 0 > t i :

送信する :

ピクチャの圧縮ビット及び

デブロッキングパラメータ A 0 及び B 0 .

参照ピクチャキャッシュに T S 0 を格納する。

50

現在のピクチャに対する処理を終了する。

If $[A_1 + GA, B_1] = [A_0, B_0]$ の場合、 $E_2 = E_0$ を設定する。

Else, $E_2 = E_1$ を設定する。

If $[A_1 + GA, B_1] \neq [A_0, B_0]$ 且つ $|A_1 + GA| \leq 6$:

ベクトル $[A_1 + GA, B_1]$ から取得されたパラメータを使用して、復号化ピクチャをデブロッキングする。

元のピクチャとデブロッキングピクチャとの間の平均二乗誤差 E_2 を計算する、

If $[A_1, B_1 + GB] = [A_0, B_0]$ $E_3 = E_0$ を設定する。 10

Else, $E_3 = E_1$ を設定する。

If $[A_1, B_1 + GB] \neq [A_0, B_0]$ 且つ $|B_1 + GB| \leq 6$:

ベクトル $[A_1, B_1 + GB]$ から取得されたパラメータを使用して、復号化ピクチャをデブロッキングする。

元のピクチャとデブロッキングピクチャとの間の平均二乗誤差 E_3 を計算する。

$G = [(E_2 - E_1) \times GA, (E_3 - E_1) \times GB]$ として傾斜ベクトル G を計算する

.

20

If $|G| / E_1 < ts$:

送信する :

ピクチャの圧縮ビット及び

デブロッキングパラメータ A_1 及び B_1 .

参照ピクチャキャッシュに TS_1 を格納する。

現在のピクチャに対する処理を終了する。

Else,

デブロッキングパラメータベクトル $[A_0, B_0] = [A_1, B_1]$ を設定する。

$E_0 = E_1$ を設定する。

$TS_0 = TS_1$ を設定する。 30

$|E_1 - E_2| > ta$ の場合、 $GA = \text{符号}((E_1 - E_2) \times GA)$ を設定する。

$|E_1 - E_3| > tb$ の場合、 $GB = \text{符号}((E_1 - E_3) \times GB)$ を設定する。

履歴テーブルに $[A_1, B_1]$ を格納する。

新規のデブロッキングパラメータベクトル $[A_1, B_1] = [A_1, B_1] - w \times G / |G|$ を計算する。ここで、新規の A_1 及び B_1 は範囲 $[-6, 6]$ 内の整数に丸められる。

If $[A_1, B_1]$ が履歴テーブルと一致する :

送信する :

ピクチャの圧縮ビット及び

デブロッキングパラメータ A_0 及び B_0 .

基準ピクチャキャッシュに TS_0 を格納する。

現在のピクチャに対する処理を終了する。 40

Else, 新規のベクトル $[A_1, B_1]$ を用いて、 PT_1 から開始する次の反復を実行する。

【0079】

上記の方法において、値 w はデブロッキングマトリクス内の傾斜ベクトル G の計算中に使用されるストライド長を表す。上述のように、ストライド長 w は 1.4 等の所定の値に設定される。別の実施形態において、ストライド長 w は動的に設定される。例えばストラ 50

イド長 w はより大きい値 (3, 5 等) に設定され、方法の動作中に改善される。そのような一実施形態において、ストライド長 w は、 $w = a * w$ (式中、 $a < 1$) 又は $w = w - a$ に従う倍率により減少される。更に、ストライド長 w の減少はデブロッキングフィルタの性能の観測可能な誤差に基づいて動的に設定されてもよい。新規のデブロッキング動作の各々 (上記のブロック 320 又は 620) が実行される場合、現在の反復のデブロッキング誤差と先行の反復のデブロッキング誤差とが比較される。誤差値の比率が所定の閾値を下回る場合 (例えば、 $|e_{i+1} / e_i| > th$)、ストライド長は比率が閾値を上回る場合より積極的に修正される。

【0080】

上記の実施形態において、方法 300 及び 600 は最初の反復のデブロッキングベクトルをデブロッキングマトリクスを中心 (例えば、 $[0, 0]$) に初期化した。他の実施形態は、収束を高速化するためにデブロッキングベクトルを他の値に初期化してもよい。例えば一実施形態において、デブロッキングベクトルは先行ピクチャに対する方法の実行において最終的に取得されたベクトルに最初に設定される。あるいは、デブロッキングベクトルは、符号化の種類が共通する先行ピクチャ (例えば、B ピクチャに対して動作している場合は先行の B ピクチャ、あるいは P ピクチャに対して動作している場合は先行の P ピクチャ) の処理において最終的に取得されたベクトルに最初に設定される。更に別の実施形態において、初期デブロッキングベクトルは、処理中のピクチャと同様の動きベクトルを示すか又は共通の基準ピクチャを参照する先行ピクチャの処理において取得されたベクトルの値に設定される。あるいは、デブロッキングベクトルは、量子化パラメータ、画素のアスペクト比、符号化速度、ノイズレベル又は画像の複雑性等の共通の符号化パラメータを示す他のピクチャの処理から取得された値に最初に設定される。

【0081】

本発明の原理は、種々の誤差計算を用いて適用される。例えば、ブロック 330 及び 380 (図 9) 又はブロック 630 及び 680 (図 11) で実行される誤差計算は、

【0082】

【数 5】

$$err = \sum_{i,j} (p1_{i,j} - p2_{i,j})^2$$

の形態である対象ピクチャ間の平均二乗誤差を計算することにより計算される。式中、 $p1$ 、 $p2$ は一対のフレームからの画素を表し、 i 、 j はピクチャ内の画素の場所を表す。画素値は閾値と比較可能なスカラー値であってもよい。

【0083】

別の実施形態において、誤差計算は、

【0084】

【数 6】

$$err = \sum_{i,j} d_{i,j} \cdot (p1_{i,j} - p2_{i,j})^2$$

の形態の重み付け平均二乗誤差計算として実行される。式中、 $d_{i,j}$ は復号化ブロック間の最近接境界から各画素までの距離を表す。そのような一実施形態は、ブロック化アーチファクトが最も目立つ傾向にあるブロック境界で生じる誤差に相対的に大きい重みを割り当てる。

【0085】

更なる一実施形態において、誤差計算は、ブロック境界のエッジから所定の距離内に存在し且つ数学的に共通の符号を示す各画素の誤差を合計する。各画素の誤差は、エッジからの距離に従って重み付けされてもよい。結果は、ピクチャにわたり平均二乗される。本

10

20

30

40

50

実施形態において、誤差計算は、ブロックのエッジ付近の画素に大きい重みを割り当てるだけでなく、エッジに沿う共通の誤差にも大きい重みを付与する。共通の符号を有する誤差はランダムノイズ誤差より可視である傾向がある。

【0086】

別の実施形態において、平均二乗誤差の基準を使用する代わりに、誤差計算は主観的に有意義の誤差を考慮する誤差モデルを使用して実行される。例えば誤差計算は、元のビデオと符号化ビデオとの間の誤差が目立たない最短視距離(Minimum Viewing Distance) (「MVD」)を計算する。この尺度をゼロ最小可知歪み(zero Just Noticeable Distortion) (「JND」)に対する最短視距離とも呼ぶ。本実施形態において、JND値は、同時係属出願であるs.n.12/415,340号「Quality Metrics for Coded Video Using Just Noticeable Difference Models」において開示されるように計算される。ブロック340 (図9)又は640 (図11)において、計算されたMVDが所定の閾値を下回る場合、方法はそれぞれブロック350又は650へ進む。

【0087】

上記の説明において、誤差計算はソースピクチャと復号化されデブロッキングされたピクチャとの間の誤差を判定するものとして説明された。別の実施形態において、誤差計算は、フィルタリングされていないソースピクチャではなくフィルタリングされたソースピクチャを使用して実行される。このように、上述の方法は各ピクチャ(又はピクチャの一部)にわたり、フィルタリングされた符号化されていない現在のmcブロックと符号化され且つデブロッキングされた現在のmcブロックとの間の誤差を計算する。符号化されていない現在のmcブロックをフィルタリングするために使用されるフィルタは、標準化されるか又は復号器に通信される必要がない。それらは、上述のパラメータ等のパラメータ、あるいは入力ビデオにおけるノイズレベル等の復号器が認識していない他のパラメータに適用してもよい。鮮明なエッジに更なる重みを付与するために、それらは高い空間周波数を強調する。

【0088】

図13は、本発明の一実施形態に係る復号器900の簡略なブロック図を示す。復号器900は、可変長復号器910、逆量子化器920、逆変換部930、加算器940、フレームバッファ950及びデブロッキングフィルタ960を含む。復号器900は、参照ピクチャキャッシュ970及び動き補償予測器980を含む予測部を更に含む。

【0089】

可変長復号器910は、チャンネルバッファから受信したデータを復号化する。可変長復号器910は、逆量子化器920に符号化係数データを転送し、動き補償予測器980に動きベクトルを転送し、デブロッキングフィルタ960にデブロッキングベクトルデータを転送する。逆量子化器920は、逆可変長復号器910から受信した係数データに量子化パラメータを乗算する。逆変換部930は、逆量子化器920から受信した逆量子化された係数データを画素データに変換する。その名称が示す通り、逆変換部930は、符号器の変換部により実行された変換動作(例えば、DCT又はウェーブレット変換)の逆を実行する。加算器940は、各画素に対して、逆変換部930により取得された残余画素データと動き補償予測器980から取得した予測画素データとを加算する。加算器940は回復mcブロックデータを出力し、回復フレームがそれから構成され且つ表示装置(不図示)においてレンダリングされる。フレームバッファ950は復号化mcブロックを蓄積し、それらから再構成フレームを構築する。デブロッキングフィルタ960は、チャンネルにより識別されたフィルタリングパラメータに従って、回復フレームデータに対してデブロッキングフィルタリング動作を実行する。

【0090】

動き補償予測は、参照ピクチャキャッシュ970及び動き補償予測器980を介して行われる。参照ピクチャキャッシュ970は、参照フレームとして識別されたフレーム(例えば、復号化I又はPフレーム)に対するデブロッキングフィルタ960により出力され

た回復画像データを格納する。動き補償予測器 980 は、チャンネルから受信した mc ブロックの動きベクトルデータに 応答して、参照ピクチャキャッシュ 970 から参照 mc ブロックを検索する。動き補償予測器は、加算器 940 に参照 mc ブロックを出力する。

【0091】

上述の説明において、本発明の種々の実施形態に従って構成されるビデオ符号化システムにおいて使用される機能ブロックが識別された。実際は、これらのシステムは、内蔵ビデオカメラを備える移動装置（例えば、カメラ付き電話、娯楽システム及びコンピュータ）等の種々の装置、並びに / 又はテレビ会議機器及びカメラ利用可能デスクトップコンピュータ等の有線通信システムにおいて適用される。いくつかの応用例において、上述の機能ブロックは、ブロックがコンピュータプログラムの別個の要素として提供される一体型ソフトウェアシステムの要素として提供される。他の応用例において、機能ブロックは、デジタル信号プロセッサ又は特定用途向け集積回路内の機能単位等の処理システムの個別の回路構成要素として提供される。本発明の更に他の応用例は、専用ハードウェア及びソフトウェア構成要素の複合システムとして実現される。更に、本明細書中で説明される機能ブロックは、別個のユニットとして提供される必要がない。例えば、図 8 はブロックベース符号化チェーン 210 及び予測部 220 の構成要素を別個のユニットとして示すが、1 つ以上実施形態において、それらの一部又は全てが一体化されてもよく、それらは別個のユニットである必要がない。特に指示されない限り、そのような実現例の詳細は本発明の動作にとって重要ではない。

【0092】

本明細書において、本発明のいくつかの実施形態が特に示され且つ / 又は説明される。しかし、本発明の変更及び変形は、本発明の趣旨の範囲から逸脱せずに上記の教示及び添付の特許請求の範囲の範囲に含まれることが理解されるだろう。

【図 1】

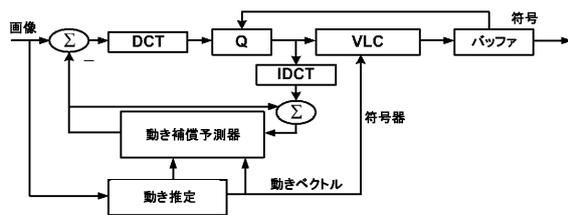


FIG. 1

【図 3】

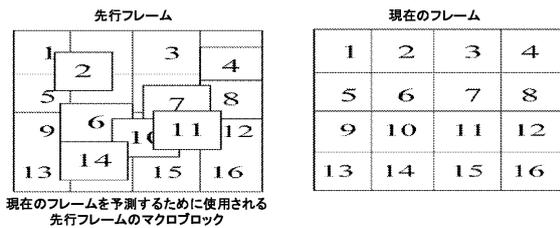


FIG. 3

【図 2】

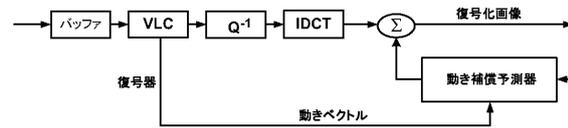


FIG. 2

【図 4】

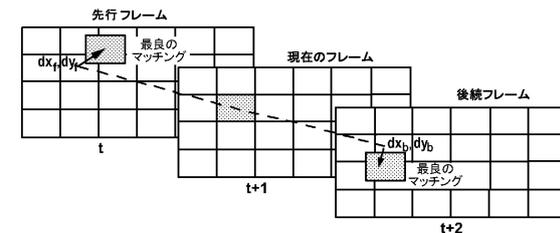


FIG. 4

【 図 1 1 】

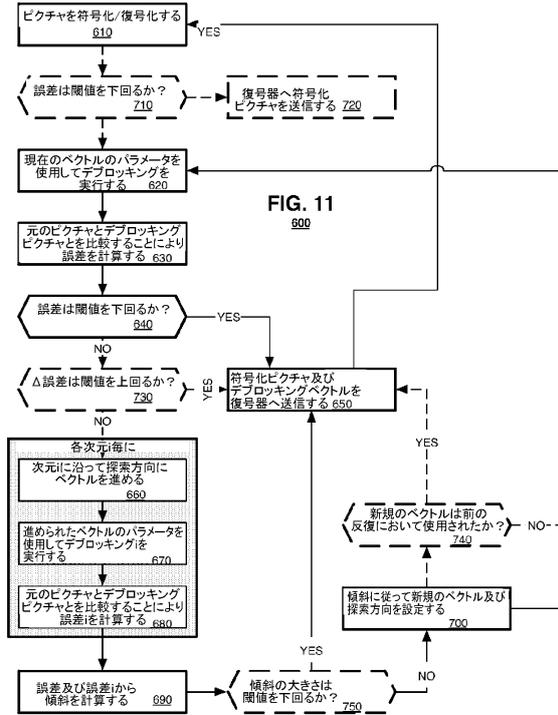


FIG. 11
900

【 図 1 2 】

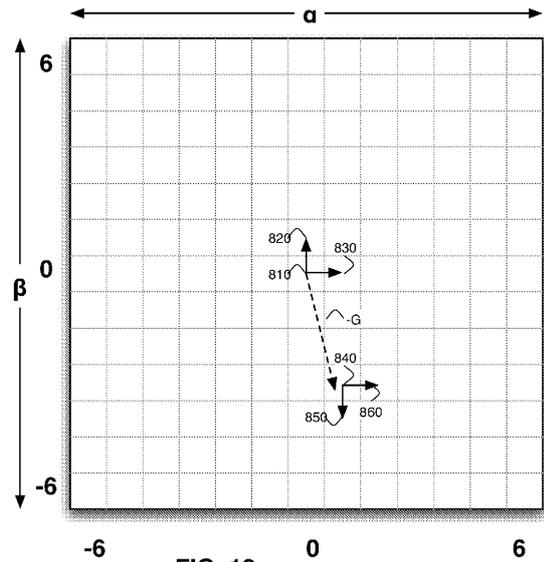


FIG. 12

【 図 1 3 】

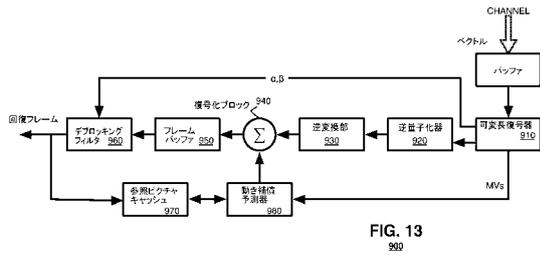


FIG. 13
900

【図7】

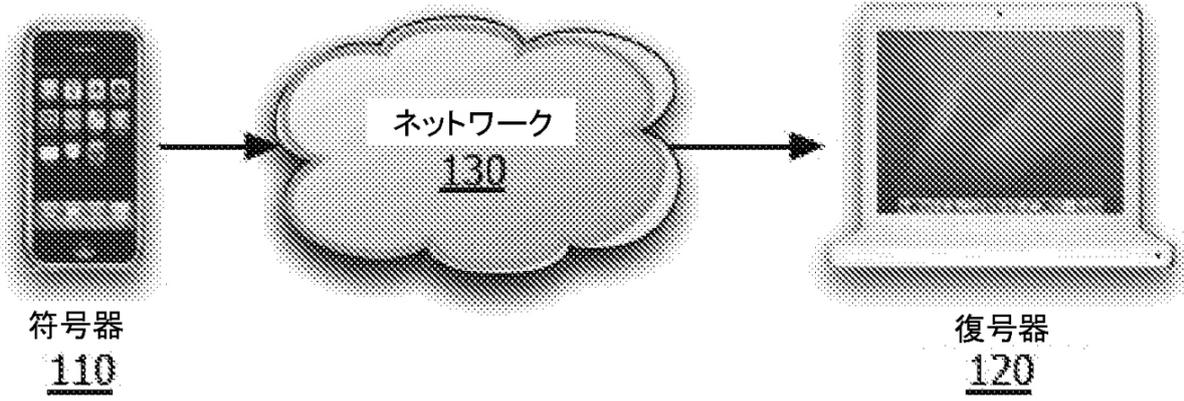


FIG. 7
100

フロントページの続き

(74)代理人 100134175

弁理士 永川 行光

(72)発明者 ハスケル, バーリン ジェフリー

アメリカ合衆国 カリフォルニア州 94040, マウンテン ビュー, フェアブルック 1
190

審査官 川崎 優

(56)参考文献 国際公開第2010/047104(WO, A1)

国際公開第2012/012582(WO, A1)

特開2008-022404(JP, A)

(58)調査した分野(Int.Cl., DB名)

H04N 19/00-98