

(19) 日本国特許庁(JP)

## (12) 特許公報(B2)

(11) 特許番号

特許第4988715号  
(P4988715)

(45) 発行日 平成24年8月1日(2012.8.1)

(24) 登録日 平成24年5月11日(2012.5.11)

(51) Int.Cl.

H04N 7/30 (2006.01)

F 1

H04N 7/133

Z

請求項の数 34 (全 37 頁)

(21) 出願番号	特願2008-512582 (P2008-512582)
(86) (22) 出願日	平成18年5月19日 (2006.5.19)
(65) 公表番号	特表2008-541672 (P2008-541672A)
(43) 公表日	平成20年11月20日 (2008.11.20)
(86) 國際出願番号	PCT/US2006/019641
(87) 國際公開番号	W02006/127546
(87) 國際公開日	平成18年11月30日 (2006.11.30)
審査請求日	平成21年2月24日 (2009.2.24)
(31) 優先権主張番号	60/683,240
(32) 優先日	平成17年5月20日 (2005.5.20)
(33) 優先権主張国	米国(US)
(31) 優先権主張番号	11/437,515
(32) 優先日	平成18年5月18日 (2006.5.18)
(33) 優先権主張国	米国(US)

(73) 特許権者	392026693
	株式会社エヌ・ティ・ティ・ドコモ
	東京都千代田区永田町二丁目11番1号
(74) 代理人	100088155
	弁理士 長谷川 芳樹
(74) 代理人	100092657
	弁理士 寺崎 史朗
(74) 代理人	100121980
	弁理士 沖山 隆
(74) 代理人	100128107
	弁理士 深石 賢治
(72) 発明者	レレスク, ダン
	アメリカ合衆国, カリフォルニア州, モーガン ヒル, セラ アヴェニュー 18325

最終頁に続く

(54) 【発明の名称】ビデオ符号化におけるノイズフィルタリング方法及び装置

## (57) 【特許請求の範囲】

## 【請求項1】

ビデオフレームにおけるデータを符号化するためのビデオ符号化器であって、変換されたフレームデータに対して動作するノイズフィルタリングモジュールであって、複数の変換係数それぞれの信号パワーを推定し、前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較し、前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することによって、前記複数の変換係数に対する信号推定を実行する、該ノイズフィルタリングモジュール、  
を備え、

10

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定する、ことを含むビデオ符号化器。

## 【請求項2】

ビデオフレームにおけるデータを符号化するためのビデオ符号化器であって、変換されたフレームデータに対して動作するノイズフィルタリングモジュールであって

20

、複数の変換係数それぞれの信号パワーを推定し、前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較し、前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することによつて、前記複数の変換係数に対する信号推定を実行する、該ノイズフィルタリングモジュール、  
を備え、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、  
係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によつて求められる値に設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更せずに残す、ことを含むビデオ符号化器。

#### 【請求項3】

前記少なくとも一つの閾値は、一対の閾値を含む請求項1又は2に記載のビデオ符号化器。

#### 【請求項4】

ノイズフィルタは、  
復号されたフレームデータに応じて複数の変換係数を生成する変換モジュールと、  
変換モジュールによって生成された変換係数に応じて、ノイズ低減が行われた変換係数を生成する推定器と、

推定器から出力されたノイズ低減が行われた変換係数からノイズ低減が行われたサンプルを生成する逆変換モジュールと、

逆変換モジュールによって生成されたノイズ低減が行われたサンプルにフィルタをかけて、再構築されたフレームサンプルを生成するフィルタと、  
を備える請求項1又は2に記載のビデオ符号化器。

#### 【請求項5】

変換モジュール、逆変換モジュール及びフィルタは、フレーム内の処理されている位置を特定するマスクに基づいて機能する請求項4に記載のビデオ符号化器。

#### 【請求項6】

マスクを生成するマスク位置生成器を更に備える請求項5に記載のビデオ符号化器。

#### 【請求項7】

複数の変換係数それぞれの信号パワーを推定するステップと、  
前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較するステップと、  
、  
前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップと、  
を含み、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、  
係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定するステップと、を含む方法。

#### 【請求項8】

複数の変換係数それぞれの信号パワーを推定するステップと、

10

20

30

40

50

前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較するステップと、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップと、  
を含み、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップは、前記変換係数それぞれに対して

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、  
係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によつて求められる値に設定するステップと、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更せずに残すステップと、を含む方法。

**【請求項9】**

前記少なくとも一つの閾値は、一対の閾値を含む請求項7又は8に記載の方法。

**【請求項10】**

復号されたフレームデータに応じて複数の変換係数を生成するステップと、  
 変換係数に応じて、ノイズ低減が行われた変換係数を生成するステップと、  
 推定器から出力されたノイズ低減が行われた変換係数からノイズ低減が行われたサンプルを生成するステップと、  
 逆変換モジュールによって生成されたノイズ低減が行われたサンプルにフィルタをかけて、再構築されたフレームサンプルを生成するステップと、  
 を更に含む請求項7又は8に記載の方法。

**【請求項11】**

復号されたフレームデータに応じて複数の変換係数を生成するステップと、  
 変換係数に応じて、ノイズ低減が行われた変換係数を生成するステップと、  
 推定器から出力されたノイズ低減が行われた変換係数からノイズ低減が行われたサンプルを生成するステップと、  
 逆変換モジュールによって生成されたノイズ低減が行われたサンプルにフィルタをかけて、フレーム内の処理されている位置を特定するマスクに基づいて、再構築されたフレームサンプルを生成するステップと、  
 を含む請求項10に記載の方法。

**【請求項12】**

命令を格納する一つ以上のコンピュータ可読記録媒体であつて、該命令がシステムによる実行時に、該システムに、

複数の変換係数それぞれの信号パワーを推定するステップと、  
 前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較するステップと、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップと、  
 を含み、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップは、前記変換係数それぞれに対して

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、  
 係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスによつて生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によつて求められる値に設定

10

20

30

40

50

するステップと、を含む方法を実行させるコンピュータ可読記録媒体。

**【請求項 1 3】**

命令を格納する一つ以上のコンピュータ可読記録媒体であって、該命令がシステムによる実行時に、該システムに、

複数の変換係数それぞれの信号パワーを推定するステップと、

前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較するステップと、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップと、

を含み、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定するステップは、前記変換係数それぞれに対して

係数の信号パワーが第 1 の閾値未満であれば、係数をゼロに設定するステップと、

係数の信号パワーが第 1 の閾値未満でなくかつ第 2 の閾値未満であれば、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第 1 の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第 1 の閾値の間の関係を定めた式によって求められる値に設定するステップと、

係数の信号パワーが第 1 の閾値未満でなくかつ第 2 の閾値未満でなければ、係数を変更せずに残すステップと、を含む方法を実行させるコンピュータ可読記録媒体。

**【請求項 1 4】**

ビデオフレームデータに対してノイズ低減を行うための方法であって、

メモリからフレームデータを取得するステップと、

前記フレーム内のマスク下のサンプルに変換を適用し、変換係数を生成するステップと、

前記変換係数に対してノイズ低減を行うステップと、

前記ノイズ低減が行われた変換係数に逆変換を適用するステップと、

前記ノイズ低減が行われたフレームサンプルであってフレーム内の位置に対応する該フレームサンプルをフィルタリングして、該位置におけるフィルタリングされたデータサンプルを得るステップと、

を含み、

前記変換係数に対してノイズ低減を行うステップは、

変換係数それぞれの信号パワーを推定するステップと、

信号パワーを閾値と比較するステップと、

信号パワーが閾値未満であれば、前記係数それぞれをゼロに設定するステップと、

信号パワーが閾値未満でなければ、信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び閾値に基づいて前記係数それぞれの値を調節するステップと、  
を含む方法。

**【請求項 1 5】**

ビデオフレームデータに対してノイズ低減を行うための方法であって、

メモリからフレームデータを取得するステップと、

前記フレーム内のマスク下のサンプルに変換を適用し、変換係数を生成するステップと、

前記変換係数に対してノイズ低減を行うステップと、

前記ノイズ低減が行われた変換係数に逆変換を適用するステップと、

前記ノイズ低減が行われたフレームサンプルであってフレーム内の位置に対応する該フレームサンプルをフィルタリングして、該位置におけるフィルタリングされたデータサンプルを得るステップと、

を含み、

前記変換係数に対してノイズ低減を行うステップは、

10

20

30

40

50

変換係数それぞれの信号パワーを推定するステップと、  
信号パワーを第1の閾値と比較するステップと、  
信号パワーが第1の閾値未満であれば、前記変換係数それぞれをゼロに設定するステップと、  
信号パワーを第2の閾値と比較するステップと、  
信号パワーが第2の閾値未満でありかつ第1の閾値未満でなければ、信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づいて前記変換係数それぞれの値を調節するステップと、  
信号パワーが第2の閾値未満でなくかつ第1の閾値未満でなければ、前記変換係数それぞれの値を変更せずに残すステップと、  
を含む方法。

**【請求項16】**

命令を格納する一つ以上のコンピュータ可読記録媒体であって、該命令がシステムによる実行時に、該システムに、ビデオフレームデータに対するノイズ低減を行う方法であつて、

メモリからフレームデータを取得するステップと、  
前記フレーム内のサンプルに変換を適用して、変換係数を生成するステップと、  
前記変換係数に対してノイズ低減を行うステップと、  
前記ノイズ低減が行われた変換係数に逆変換を適用するステップと、  
前記ノイズ低減が行われたフレームサンプルであつてフレーム内の位置に対応する該フレームサンプルをフィルタリングし、該位置におけるフィルタリングされたデータサンプルを得るステップと、  
を含み、  
前記変換係数に対してノイズ低減を行うステップは、  
変換係数それぞれの信号パワーを推定するステップと、  
信号パワーを閾値と比較するステップと、  
信号パワーが閾値未満であれば、前記係数それぞれをゼロに設定するステップと、  
信号パワーが閾値未満でなければ、信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び閾値に基づいて前記係数それぞれの値を調節するステップと、  
を含む該方法を実行させるコンピュータ可読記録媒体。

**【請求項17】**

命令を格納する一つ以上のコンピュータ可読記録媒体であつて、該命令がシステムによる実行時に、該システムに、ビデオフレームデータに対するノイズ低減を行う方法であつて、

メモリからフレームデータを取得するステップと、  
前記フレーム内のサンプルに変換を適用して、変換係数を生成するステップと、  
前記変換係数に対してノイズ低減を行うステップと、  
前記ノイズ低減が行われた変換係数に逆変換を適用するステップと、  
前記ノイズ低減が行われたフレームサンプルであつてフレーム内の位置に対応する該フレームサンプルをフィルタリングし、該位置におけるフィルタリングされたデータサンプルを得るステップと、  
を含み、  
前記変換係数に対してノイズ低減を行うステップは、  
変換係数それぞれの信号パワーを推定するステップと、  
信号パワーを第1の閾値と比較するステップと、  
信号パワーが第1の閾値未満であれば、前記変換係数それぞれをゼロに設定するステップと、  
信号パワーを第2の閾値と比較するステップと、  
信号パワーが第2の閾値未満でありかつ第1の閾値未満でなければ、信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づいて前記変換係数

10

20

30

40

50

それぞれの値を調節するステップと、

信号パワーが第2の閾値未満でなくかつ第1の閾値未満でなければ、前記変換係数それぞれの値を変更せずに残すステップと、

を含む該方法を実行させるコンピュータ可読記録媒体。

**【請求項18】**

フレームの空間データを、信号処理に適する別の領域に変換する変換モジュールと、  
フレームデータに対応する変換係数のノイズ低減された推定値を、信号パワーを推定し  
、前記推定された信号パワーを一つ以上の閾値と比較し、前記変換係数を、前記比較の結果、符号化プロセスによって生じるノイズのノイズパワー、及び閾値に従って調整することによって、生成する推定器モジュールと、

10

前記ノイズ低減された変換係数に逆変換を適用し、前記マスク下の前記データをその元の領域に戻す逆変換モジュールと、

各フレーム位置におけるフィルタリングされたサンプルを、前記推定器による処理によって得られるその推定値を組み合わせることによって取得する、フィルタモジュールと、を備え、

前記変換係数を、前記比較の結果、ノイズパワー、及び閾値に従って調整することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、ノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定する、ことを含むノイズフィルタリングモジュール。

20

**【請求項19】**

フレームの空間データを、信号処理に適する別の領域に変換する変換モジュールと、  
フレームデータに対応する変換係数のノイズ低減された推定値を、信号パワーを推定し  
、前記推定された信号パワーを一つ以上の閾値と比較し、前記変換係数を、前記比較の結果、符号化プロセスによって生じるノイズのノイズパワー、及び閾値に従って調整することによって、生成する推定器モジュールと、

前記ノイズ低減された変換係数に逆変換を適用し、前記マスク下の前記データをその元の領域に戻す逆変換モジュールと、

30

各フレーム位置におけるフィルタリングされたサンプルを、前記推定器による処理によって得られるその推定値を組み合わせることによって取得する、フィルタモジュールと、を備え、

前記変換係数を、前記比較の結果、ノイズパワー、及び閾値に従って調整することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パワー、ノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更せずに残す、ことを含むノイズフィルタリングモジュール。

40

**【請求項20】**

空間的なマスクが位置するフレーム内の位置を生成して、該位置におけるマスク下で更に処理されるデータサンプルを決定するマスク生成器モジュールを更に備える請求項18又は19に記載のノイズフィルタリングモジュール。

**【請求項21】**

フレームデータ推定値を格納するメモリモジュールを更に備える請求項18又は19に記載のノイズフィルタリングモジュール。

**【請求項22】**

信号パワーを推定する手段と、

50

前記推定された信号パワーを一つ以上の閾値と比較する手段と、  
 信号の変換係数値を、前記信号の変換係数値と一つ以上の閾値との比較の結果に従って  
 、且つ、前記推定された信号パワーの値と、ノイズのパワーと、前記一つ以上の閾値とに  
 基づいて、調整する手段と、  
 を備え、

信号の変換係数値を、前記信号の変換係数値と一つ以上の閾値との比較の結果に従って  
、且つ、前記推定された信号パワーの値と、ノイズのパワーと、前記一つ以上の閾値とに  
基づいて、調整する手段は、

前記変換係数値それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

10

係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスに  
 よって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー  
 、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定  
 する、推定器モジュール。

#### 【請求項23】

信号パワーを推定する手段と、

前記推定された信号パワーを一つ以上の閾値と比較する手段と、

信号の変換係数値を、前記信号の変換係数値と一つ以上の閾値との比較の結果に従って  
 、且つ、前記推定された信号パワーの値と、ノイズのパワーと、前記一つ以上の閾値とに  
 基づいて、調整する手段と、

20

を備え、

信号の変換係数値を、前記信号の変換係数値と一つ以上の閾値との比較の結果に従って  
 、且つ、前記推定された信号パワーの値と、ノイズのパワーと、前記一つ以上の閾値とに  
 基づいて、調整する手段は、

前記変換係数値それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パ  
 ワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当  
 該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によ  
 って求められる値に設定し、

30

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更  
 せずに残す、推定器モジュール。

#### 【請求項24】

ビデオフレームにおけるデータを復号するビデオ復号器であって、

変換されたフレームデータに対して動作するノイズfiltrリングモジュールであって  
 、複数の変換係数それぞれの信号パワーを推定し、前記変換係数それぞれの前記信号パ  
 ワーを少なくとも一つの閾値と比較し、前記変換係数それぞれの値を、前記信号パワーと前  
 記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することによ  
 って、複数の変換係数に対する信号推定を実行する、該ノイズfiltrリングモジュール  
 、

40

を備え、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の  
 結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスに  
 よって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー  
 、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定  
 する、ことを含むビデオ復号器。

#### 【請求項25】

ビデオフレームにおけるデータを復号するビデオ復号器であって、

50

変換されたフレームデータに対して動作するノイズフィルタリングモジュールであって、複数の変換係数それぞれの信号パワーを推定し、前記変換係数それぞれの前記信号パワーを少なくとも一つの閾値と比較し、前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することによつて、複数の変換係数に対する信号推定を実行する、該ノイズフィルタリングモジュールを備え、

前記変換係数それぞれの値を、前記信号パワーと前記少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によつて求められる値に設定し、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更せずに残す、ことを含むビデオ復号器。

#### 【請求項 2 6】

前記少なくとも一つの閾値は、一対の閾値を含む請求項 2 4 又は 2 5 に記載のビデオ復号器。

#### 【請求項 2 7】

ノイズフィルタは、

復号されたフレームデータに応じて複数の変換係数を生成する変換モジュールと、変換モジュールによって生成された変換係数に応じて、ノイズ低減が行われた変換係数を生成する推定器と、

推定器から出力されたノイズ低減が行われた変換係数からノイズ低減が行われたサンプルを生成する逆変換モジュールと、

逆変換モジュールによって生成されたノイズ低減が行われたサンプルにフィルタをかけて、再構築されたフレームサンプルを生成するフィルタと、  
を備える請求項 2 4 又は 2 5 に記載のビデオ復号器。

#### 【請求項 2 8】

変換モジュール、逆変換モジュール及びフィルタは、フレーム内の処理されている位置を特定するマスクに基づいて機能する請求項 2 7 に記載のビデオ復号器。

#### 【請求項 2 9】

マスクを生成するマスク位置生成器を更に備える請求項 2 8 に記載のビデオ復号器。

#### 【請求項 3 0】

フレームデータのブロックの予測誤差を復号するステップと、  
前記ブロックに関連付けられている動きベクトルを復号するステップと、  
前記ブロックに対する動き補償を、前記動きベクトルと基準フレームとを使用し実行し、予測ブロックを求めるステップと、

前記予測ブロックと前記予測誤差とを使用して前記ブロックを再構築するステップと、  
前記フレームデータから複数の変換係数を生成し、前記変換係数に応じて前記変換係数の信号パワーの推定値を一つ以上の閾値と比較して、変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定して、ノイズ低減された変換係数を生成し、ノイズ低減されたサンプルをフィルタリングして再構築されたフレームサンプルを生成することによって、再構築されたブロックサンプルをフィルタリングするステップと、

を含み、

変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、

10

20

30

40

50

係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定するステップと、を含む復号方法。

【請求項31】

フレームデータのブロックの予測誤差を復号するステップと、

前記ブロックに関連付けられている動きベクトルを復号するステップと、

前記ブロックに対する動き補償を、前記動きベクトルと基準フレームとを使用し実行し、予測ブロックを求めるステップと、

前記予測ブロックと前記予測誤差とを使用して前記ブロックを再構築するステップと、

前記フレームデータから複数の変換係数を生成し、前記変換係数に応じて前記変換係数の信号パワーの推定値を一つ以上の閾値と比較して、変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定して、ノイズ低減された変換係数を生成し、ノイズ低減されたサンプルをフィルタリングして再構築されたフレームサンプルを生成することによって、再構築されたブロックサンプルをフィルタリングするステップと、

を含み、

変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定するステップと、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更せずに残すステップと、を含む復号方法。

【請求項32】

少なくとも一つの閾値は、一対の閾値を含む請求項30又は31に記載の復号方法。

【請求項33】

命令を格納する一つ以上のコンピュータ可読記録媒体であって、該命令がシステムによる実行時に、該システムに、

フレームデータのブロックの予測誤差を復号するステップと、

前記ブロックに関連付けられている動きベクトルを復号するステップと、

前記ブロックに対する動き補償を、前記動きベクトルと基準フレームとを使用して実行し、予測ブロックを求めるステップと、

前記予測ブロックと前記予測誤差とを使用して前記ブロックを再構築するステップと、

前記フレームデータから複数の変換係数を生成し、前記変換係数に応じて前記変換係数の信号パワーの推定値を一つ以上の閾値と比較して、変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定して、ノイズ低減された変換係数を生成し、ノイズ低減されたサンプルをフィルタリングして再構築されたフレームサンプルを生成することによって、再構築されたブロックサンプルをフィルタリングするステップと、

を含み、

変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、

係数の信号パワーが第1の閾値未満でなければ、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定するステップと、を含む復号方法を実行させるコンピュータ可読記録媒体。

10

20

30

40

50

**【請求項 3 4】**

命令を格納する一つ以上のコンピュータ可読記録媒体であって、該命令がシステムによる実行時に、該システムに、

フレームデータのブロックの予測誤差を復号するステップと、

前記ブロックに関連付けられている動きベクトルを復号するステップと、

前記ブロックに対する動き補償を、前記動きベクトルと基準フレームとを使用して実行し、予測ブロックを求めるステップと、

前記予測ブロックと前記予測誤差とを使用して前記ブロックを再構築するステップと、

前記フレームデータから複数の変換係数を生成し、前記変換係数に応じて前記変換係数の信号パワーの推定値を一つ以上の閾値と比較して、変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定して、ノイズ低減された変換係数を生成し、ノイズ低減されたサンプルをフィルタリングして再構築されたフレームサンプルを生成することによって、再構築されたブロックサンプルをフィルタリングするステップと、

を含み、

変換係数それぞれの値を、信号パワーと一つ以上の閾値との比較の結果に少なくとも部分的に基づいて設定することは、前記変換係数それぞれに対して、

係数の信号パワーが第1の閾値未満であれば、係数をゼロに設定するステップと、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満であれば、係数を信号パワー、符号化プロセスによって生じるノイズのノイズパワー及び第1の閾値に基づき、当該係数、当該信号パワー、当該ノイズパワー、当該第1の閾値の間の関係を定めた式によって求められる値に設定するステップと、

係数の信号パワーが第1の閾値未満でなくかつ第2の閾値未満でなければ、係数を変更せずに残すステップと、を含む復号方法を実行させるコンピュータ可読記録媒体。

**【発明の詳細な説明】****【関連出願】****【0001】**

[0001]本特許出願は、「ビデオ符号化におけるノイズフィルタリングの方法及び装置 (Method And Apparatus For Noise Filtering in Video Coding)」と題して2005年5月20日に出願された対応の仮特許出願第60/683,240号の優先権を主張するものであり、当該仮特許出願を参照することによって援用するものである。

**【発明の分野】****【0002】**

[0002]本発明は、ビデオ符号化の分野に関するものであり、より詳細には、本発明は、ビデオ符号化におけるノイズフィルタリングを、係数の信号パワーの推定と、それら推定値と一つ以上の閾値との比較とに基づいて、実行することに関するものである。

**【発明の背景】****【0003】**

[0003]ハイブリッドビデオ符号化の従来方法では、ビデオフレームを、シーケンス中の他のフレームを参照することなく単独で符号化するか、或いは、ビデオフレームをシーケンス中の既に符号化されて再構築されている他のフレームから予測することによって符号化する。このプロセスは、通常、画像のブロックレベルで動作する。何れの場合にも、符号化プロセス後に再構築されたフレームには、符号化プロセスによって生じる歪みが含まれており、歪みはノイズとしてモデル化することができる。このノイズは、ブロッキングアーチファクト及びレンギングアーチファクトといった符号化アーチファクトとして視覚的に現れることがある。ビデオの符号化器及び復号器によって一般に使用されるフレームストアに格納されている再構築されたフレームの幾つかは、シーケンス中の他のフレームを予測するための基準フレームとして更に使用され、それ以外のフレームは使用されない。前者の場合、フレームストアに格納されている再構築されたフレームの品質は、それを

10

20

30

40

50

表示する上で重要であるのみならず、その予測プロセスにおける使用が、参照のために当該フレームに依存するフレームであって以降に符号化されるフレームの品質にも影響する。後者の場合、基準フレームとして更には使用されないフレームの品質は、それを表示する上で重要である。従って、何れの場合にも、符号化ノイズを除去する手順は、再構築されたビデオシーケンスの品質、及び、客観的な符号化効率の向上のために有益である。

#### 【0004】

[0004] 最近のコーデックの殆どには、符号化の結果としてのノイズのレベルを低減する何らかのタイプのフィルタリングが含まれている。これらのフィルタリングは、コーデックにおける符号化ループの外側又は内側で動作することができる。符号化ループの内側で動作するループフィルタの場合には、一般的に、符号化器と復号器とにおいて同じ利用可能データを使用して同じオペレーションを実行し、従って、そのフィルタリングに固有の副情報を符号化器から復号器に明示的に送信する必要がない。一般に使用されるノイズフィルタリング方法としては、再構築されたフレームデータの単純なローパスフィルタリングであって画像を平滑化して符号化アーチファクトの出現を低減させるローパスフィルタや、画像の局所的な特性（例：ブロックの縁部）に基づいて自身の強度を適合させるループフィルタが挙げられる。10

#### 【0005】

[0005] 符号化ループの外側でのフィルタリング手法、例えば、再構築された（ノイズが含まれる）フレームの復号器におけるローパスフィルタリングは、通常、復号後のビデオシーケンスの全体的な画質を十分に向上させることができない。「画像符号化におけるブロッキング効果の低減（Reduction of Blocking Effects in Image Coding）」（H. C. Reeve, J. S. Lim）（Opt. Eng., Vol. 23, No. 1, p. 34~37, 1984年1/2月）を参照されたい。符号化器における符号化ループと復号器における復号ループの内側でフィルタリングを実行するループ内フィルタは、優れた性能を有している詳しくは、「H. 264 / AVCビデオ符号化規格の概要（Overview of the H. 264 / AVC Video Coding Standard）」（Wiegandら）（IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, Issue: 7, p. 560~576, 2003年7月）を参照されたい。これらのフィルタは、通常、ビデオコーデックが使用する量子化方式と、フィルタリングする信号の特性に基づいて、フィルタリングの強度を発見的方式で適合させる。更に、これらのフィルタは符号化ループの中で動作するため、予測に使用する基準フレームの品質が向上し、従って、符号化プロセスの効率が向上する。2030

#### 【0006】

[0006] 発見的に処理を行なうものとは対照的に、ノイズからの信号推定であって特定の信号モデルを使用する信号推定に基づくフィルタリング手法が存在する。詳しくは、「確率、確率変数、及び確率過程（Probability, Random Variables, and Stochastic Processes）」（A. Papoulis）（3<sup>rd</sup> edition, New York, McGraw-Hill, 1991年）を参照されたい。40

#### 【0007】

[0007] 変換領域において動作する過完備のノイズ低減手法は、フレーム内の同じサンプル位置に対して複数の再構築されたインスタンス（推定値）を求め、これらを組み合わせて（例：平均する）推定の品質を高めることができるという利点を更に提供する。「推移不变ノイズ低減・除去（Translation Invariant Denoising）」（R. R. Coifman, D. L. Donoho）（Wavelets and Statistics, Springer Lecture Notes in Statistics 103, p. 125~150, New York, Springer Verlag）を参照されたい。既存のノイズ低減方法は、符号化ノイズから信号50

を取り出すことを試みるものであり、そのコアに信号推定器を有する。通常、問題の統計的情報は正確には既知ではないので、未知条件 (unknowns) について何らかの想定を行い、信号を推定する方式を選択する必要がある。最も有名な方式の一つは、未知条件の予想され得る最悪の状況時の信号推定値を最適化し、結果としてロバストな推定器をもたらすものである。詳しくは、「有界なデータ不確実性における決定論的パラメータの線形ミニマックスリグレット推定 (Linear Minimax Regret Estimation of Deterministic Parameters with Bounded Data Uncertainties)」(Y. C. Eldar、A. B. - Tal、A. Nemirovski) (IEEE Transactions on Signal Processing, Vol. 52, No. 8, 2004年8月) を参照されたい。この設計制約下では、最悪のシナリオに対してパフォーマンスが最適化されるため、結果としての推定器が控えめすぎる傾向にあることが周知である。10

#### 【0008】

[0008] 符号化ノイズフィルタリングの従来方法のパフォーマンスは、これらフィルタリング方法に固有の幾つかの特性と、ノイズの性質に関して成された仮定とによって、制限されている。符号化アーチファクトを低減するために適用される単純なローパスフィルタリング手法は、ビデオフレームにおける多様な視覚情報を扱うには有効ではなく、ビデオシーケンスに対するパフォーマンスが大きく変動する傾向にある（パフォーマンス制御の欠如）。ビデオ符号化における適応的なループ境界内フィルタリングでは、発見的手法を使用し、場合によっては最適なソリューションが保証しない。また、信号モデルの使用に基づく既存のノイズ低減・除去手法では、信号及びノイズの性質に関する非現実的な仮定の下に、具体的には、信号とノイズとが相關していないと仮定することによって、推定器を決定する。この仮定は、画像及びビデオの符号化において発生する信号ノイズ及び符号化ノイズにはあてはまらない。これらのフィルタリング手法は既存の問題の条件に適合していないので、そのパフォーマンスは高くない。20

#### 【発明の概要】

#### 【0009】

[0009] 本明細書では、ビデオフレームデータを符号化及び／又は復号する方法及び装置を開示する。一実施形態においては、ビデオ符号化器は、ノイズフィルタリングモジュールを備えており、このモジュールは、変換されたフレームデータに対して動作し、複数の変換係数それぞれの信号パワーを推定し、各係数の信号パワーを少なくとも一つの閾値と比較し、各変換係数の値を、信号パワーと少なくとも一つの閾値との比較の結果に少なくとも部分的に基づいて設定することによって、複数の変換係数に対して信号推定を実行する。30

#### 【0010】

[0010] 本発明は、以下の詳細な説明と、本発明の様々な実施形態の添付の図面とから、更に完全に理解されるであろう。しかしながら、これらの実施形態は、本発明をそれら特定の実施形態に限定するものではなく、説明及び理解のみを目的としているものと理解されたい。

#### 【本発明の詳細な説明】

#### 【0011】

[0022] ビデオ符号化及び復号用の信号ノイズ低減手法を実施する装置及び方法を開示する。一実施形態においては、フィルタリング装置は、ビデオコーデックによって生成される再構築されたフレームデータに対して動作し、そのフィルタリングプロセスは、フレームデータに対応する過完備の変換領域において行われる。符号化プロセスによって生じるノイズからの信号の推定は、この変換領域において実行される。一実施形態においては、この推定プロセスは、変換係数に対応する信号パワーを推定するステップと、推定された信号パワーを一つ以上の閾値と比較するステップと、比較の結果に基づいて変換係数を調整するステップと、を含んでいる。従って、推定プロセスでは、推定された信号パワーと、ノイズ出力と、（一つ以上の）閾値の値とを使用する。4050

## 【 0 0 1 2 】

[0023]一実施形態においては、過完備表現の下では、処理するフレーム内のデータサンプル位置のそれぞれについて、推定プロセスによって生成されるときに、その位置におけるサンプルの複数の推定値が対応する。各データサンプルのこれら初期推定値を更にフィルタリングし、処理するビデオフレーム内の位置におけるデータサンプルの最終的な推定値を得る。そのようなフィルタリングは、平均化であってもよい。

## 【 0 0 1 3 】

[0024]このように、過完備の変換領域における適応的な推定を使用して、符号化プロセス中に生じるノイズから信号を取り出す。一実施形態においては、信号パワーを推定するためのノイズフィルタにおける推定器は、符号化プロセスにおける信号及びノイズに関する仮定に基づいて動作する。また、一実施形態において、信号推定器は、例えば上述した従来技術の推定器の欠点が回避されるように、ロバスト性とパフォーマンスとの間のバランスが達成されるように設計されている。10

## 【 0 0 1 4 】

[0025]この信号ノイズ低減手法を使用する結果として、ビデオフレームフィルタリングの従来技術の手法と比較して、客観的なレート・歪みパフォーマンスが向上し、更に、復号後のビデオフレームの主観的な（視覚的な）品質が高まる。

## 【 0 0 1 5 】

[0026]以下の説明では、本発明をより完全に説明するために、数多くの詳細部分を記載してある。しかしながら、当業者には、これらの特定の詳細部分を使用せずに本発明を実施できることが明らかであろう。周知の構造及び装置については、本発明が曖昧になることがないように、詳細に示すのではなくブロック図形式で示してある。20

## 【 0 0 1 6 】

[0027]以下の詳細な説明の一部は、コンピュータメモリ内のデータビットに対する操作のアルゴリズム及び記号表現の形式で提示してある。このようなアルゴリズムによる説明及び表現は、データ処理の技術分野における当業者が、自身の開発内容の要旨を他の当業者に最も効果的に伝えるために使用する手段である。アルゴリズムは、本明細書において、及び一般的には、目的の結果に達する首尾一貫した一連のステップとして認識される。これらのステップは、物理量の物理的操作を必要とするステップである。通常（ただし絶対的ではないが）、これらの数量は、格納、伝送、結合、比較、その他の操作を行うことのできる電気信号又は磁気信号の形式をとる。これらの信号を、例えばビット、値、要素、記号、文字、術語、数字として表現することは、主として、一般的に使用されているという理由から、場合によっては便利であることが実証されている。30

## 【 0 0 1 7 】

[0028]しかしながら、上記及び同様の表現の何れも、該当する物理量が関連付けられるものであり、物理量に付された便利なラベルにすぎないことを認識しておく必要がある。以下の説明から明らかであるように、特に明記しない限りは、説明全体を通じて、「処理する」、「計算する」、「決定する」、「表示する」等の用語を用いての説明は、コンピュータシステムのレジスタ及びメモリ内で物理的（電子的）数量として表されているデータを操作して、コンピュータシステムのメモリ又はレジスタ、或いはそれ以外の、どのような情報の記憶装置、伝送装置、又は表示装置において同様に物理量として表される他のデータに変換する、コンピュータシステム、又は類似する電子計算装置の動作及び処理を意味しているものと理解されたい。40

## 【 0 0 1 8 】

[0029]本発明は、本明細書における動作を実行する装置にも関する。この装置は、必要な目的のために専用に構築することができ、或いは、格納されているコンピュータプログラムによって選択的に起動又は再構成される汎用コンピュータを備えていてもよい。そのようなコンピュータプログラムは、コンピュータ可読記憶媒体、例えば、以下に限定されないが、任意のタイプのディスク（例：フロッピーディスク、光ディスク、C D - R O M、光磁気ディスク）、R O M（読み取り専用メモリ）、R A M（ランダムアクセスメモリ）50

、E P R O M 、E E P R O M 、磁気カード又は光カード、或いは、電子的命令を格納するのに適しており、それぞれがコンピュータシステムのバスに結合されている任意のタイプの媒体、に格納することができる。

#### 【 0 0 1 9 】

[0030]本明細書に提示したアルゴリズム及び表示は、特定のコンピュータ或いは他の装置に本質的に関連するものではない。本明細書における教示内容によるプログラムは、様々な汎用システムにおいて使用することができ、或いは、場合によっては、必要な方法ステップを実行するための専用の装置を構築することが好都合であることもわかっている。これらの様々なシステムの必要な構造は、以下の説明から明らかになるであろう。更に、本発明は、特定のプログラミング言語に関連して説明していない。本明細書に説明した本発明の教示内容は、様々なプログラミング言語を使用して実施できることが理解されるであろう。10

#### 【 0 0 2 0 】

[0031]機械可読媒体には、機械（例：コンピュータ）が読むことのできる形式で情報を格納又は伝送する任意のメカニズムが含まれる。例えば、機械可読媒体としては、読み取り専用メモリ（「R O M」）、ランダムアクセスメモリ（「R A M」）、磁気ディスク記憶媒体、光記憶媒体、フラッシュメモリデバイスのほか、電気、光、音響、又はその他の形式の伝搬信号（例：搬送波、赤外線信号、デジタル信号）等が挙げられる。

#### 【 0 0 2 1 】

##### <ビデオ符号化器の実施形態の概要>

[0032]一実施形態においては、ビデオ符号化器はフィルタリングモジュールを含んでおり、このフィルタリングモジュールは、符号化ノイズを低減又は排除する目的でフレームデータを処理し、従って、符号化プロセスの客観的な品質と主観的な品質の双方を向上させる。ノイズフィルタは、過完備ベースでフレームデータを表現することによって構築される領域において、ノイズが含まれるフレームデータからビデオ信号を取り出すように動作する適応的な推定手法を使用する。本明細書に説明した装置及び方法は、フレームデータの過完備ベースの表現において、既存の従来技術とは異なる方式で推定オペレーション及びフィルタリングオペレーションを実行し、これらの装置及び方法を使用する結果として、従来技術と比較してパフォーマンスを向上させる。従来技術との差異としては、以下に限定されないが、ノイズから信号を取り出すために使用される推定器の構造、推定器が使用する他の変数の推定と、上述したように、表示を目的として、或いは参照及び表示の双方を目的として使用するフィルタリングされた最終的なデータサンプルを求めるために、推定器が生成する信号推定値を処理するために使用されるフィルタの構造、が挙げられる。2030

#### 【 0 0 2 2 】

[0033]本明細書に説明した推定手法及び／又はフィルタリングプロセスを使用する結果として、データ符号化の効率が高まり、これが符号化プロセスの優れたレート・歪み特性に反映され、更に、符号化アーチファクトが低減又は排除されることにより、従来技術の方法と比較して復号後のビデオシーケンスの視覚的な品質が向上する。

#### 【 0 0 2 3 】

[0034]図1は、ビデオ符号化器の一実施形態のブロック図である。以下、図1を参照する。ビデオ符号化器は、動き推定モジュール（M E M）1 2 9と、動き補償予測モジュール（M C P M）1 3 1と、メモリ1 2 3と、変換符号化モジュール（T C M）1 1 4と、動きデータ処理モジュール（M D P M）1 2 7と、ノイズフィルタモジュール（N F M）1 2 5と、フレームストア（F S）（例：メモリ）1 2 6と、スイッチ1 1 8，1 2 4，1 2 8，1 4 1，1 4 2と、を備えている。変換符号化器1 1 4は、変換モジュール1 1 1と、量子化器モジュール1 1 2と、エントロピ符号化モジュール1 1 3と、を含んでいる。変換符号化器1 1 4に入力されるフレームデータは、ビデオフレーム、又はD F D（d i s p l a c e d f r a m e d i f f e r e n c e）フレームによって構成することができる。D F Dフレームは、ビデオ符号化器において、ビデオフレームのデータと、4050

M C P M 1 3 1 が生成して出力するその予測との間の差をとることによって得られる。この差は、ビデオフレーム 1 0 1 と、I フレーム又はフレーム内ブロックとを受け取る減算器 1 0 3 から出力される（スイッチ 1 4 1 が閉じているとき）。M C P M 1 3 1 は、フレームストア 1 2 6 に格納されている動き補償後の再構築されたビデオフレームから成るデータに基づいて、予測を生成する。動き補償は、動き推定モジュール 1 2 9 が生成する動き情報を使用して行われる。このプロセスは、図 1 におけるスイッチの構成（後述する）によって有効になる。或いは、ビデオ符号化器（V C）が、予測を使用せずにビデオフレームを直接的に符号化するときには（従来の符号化器のフレーム内符号化モードに相当する）、M E M 1 2 9 、M C P M 1 3 1 、及び M D P M 1 2 7 は使用されない。

## 【0024】

10

[0035]ビデオ符号化器は、一般には、ブロックに分割されたフレームデータ（ブロック内の各位置における輝度値及び色差値から成る）に対して連続的に動作する。

## 【0025】

[0036]図 2 は、ビデオ符号化プロセスの一実施形態のフローチャートである。このプロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で実行されるもの等）、又は双方の組合せを備えることができる。

## 【0026】

20

[0037]図 2 において、フレームからのブロックが、以下に説明するプロセスによって、現在のフレームの全てのブロックが処理されるまで連続的に読み取られて処理される。図 2 を参照すると、このプロセスでは、処理ロジックは、最初に、現在のフレーム内のブロックにインデックスを付けるために使用するインデックス変数 i を 1 に設定する（処理ブロック 2 0 1）。次いで、処理ロジックは、現在のフレームからブロック c ( i ) を読み取る（処理ブロック 2 0 2）。

## 【0027】

[0038]ビデオ符号化器の動作の最初の段階では、図 1 におけるスイッチは以下のように構成されている。すなわち、スイッチ 1 4 1 、 1 4 2 、及び 1 2 8 は、処理するブロックのタイプに応じて、開いているか又は閉じている（フレーム内符号化するブロックの場合は開いており、従来のビデオプロセスのように予測するブロックの場合には閉じている）。スイッチ 1 2 4 は開いている。

30

## 【0028】

[0039]更に、処理ロジックは、インデックス k を 1 に初期化し（インデックス k は、現在のブロック c ( i ) に対応する探索領域内の位置）（処理ブロック 2 0 3）、動き推定探索領域内の位置 k に対応する基準フレーム内の基準ブロック b ( k ) を取得する（処理ブロック 2 0 4）。動き推定プロセスを実行し、このプロセスによって、フレームストアからの基準フレーム内の探索領域内の候補の位置を連続的に探索し、現在のブロック c ( i ) に最も整合する基準ブロックを識別する。整合の程度の量を使用して、ブロック c ( i ) とブロック b ( k ) との間の差を定量化する。動き推定プロセスは、現在のブロック c ( i ) と基準ブロック b ( k ) との間の差の量 d ( i , k ) が最小になるように、基準フレーム内の探索領域内のブロック b ( k ) を見つけることを試みる。この位置は、現在のブロック c ( i ) を起点とし基準フレーム内の基準ブロック b ( k ) を指す動きベクトル m v ( i , k ) によって指定される。

40

## 【0029】

[0040]処理ロジックは、ブロック c ( i ) とブロック b ( k ) との間の差の測度 d ( i , k ) を計算し、それをメモリに格納する（処理ブロック 2 0 5）。一実施形態においては、差 d ( i , k ) は、例えば、c ( i ) と b ( k ) のサンプルの間の差の絶対値の合計によって定量化する。このように、現在のブロック c ( i ) について、探索領域の内側の位置 k を、対応する基準ブロック b ( k ) を取得して d ( i , k ) を計算することによって調べ、誤差 d ( i , k ) をメモリに格納する。

## 【0030】

50

[0041]次いで、処理ロジックは、変数 k を 1だけインクリメントし(処理ブロック 206)、変数 k が K (探索領域内の位置の総数)より小さいか又は等しいかを調べる(処理ブロック 207)。現在のブロック c ( i )に対応する、基準フレームの探索領域位置の全てをまだ調べていない(すなわち  $k < K$ )場合、処理ロジックは処理ブロック 204 に戻り、プロセスがそこから続行する。全てを調べた(すなわち k が K より小さくない)場合、処理ロジックは処理ブロック 208 に進み、最良の、すなわち c ( i )と b ( k )との間の  $d ( i , k )$  が最小である整合ブロック b ( n )を見つける。一実施形態においては、現在のブロック c ( i )に対応する基準フレームからの K 個の探索領域位置の全てを処理した後、MEM129 の処理ロジックは、メモリに格納されている一連の値  $d ( i , k )$  のうち最小の誤差  $d_{min}$  を見つける。この誤差  $d_{min}$  は、座標が動きベクトル  $m v ( i )$  によって指定される特定の基準ブロック ( $b_{min}$  と表す) に対応する。  
10

### 【0031】

[0042]その後、処理ロジックは、動きベクトル  $m v ( i , n )$  をメモリに格納し(処理ブロック 209)、ブロック c ( i )の符号化モード  $m ( i )$  を決定して格納する(処理ブロック 210)。ブロック c ( i )の符号化モード  $m ( i )$  は、フレーム内符号化又は予測符号化である。一実施形態においては、MEM129 の処理ロジックは、ブロック c ( i )のサンプルをフレーム内符号化するときの符号化計算量と、ブロック c ( i )をブロック  $b_{min}$  を基準に予測的に符号化する計算量とを比較することによって、現在のブロック c ( i )の符号化モード  $c mode ( i )$  を決定する。前者の計算量の方が小さい場合、ブロック c ( i )のモードを「フレーム内 (INTRA)」とマークし、そうでない場合、現在のブロックを予測ブロックとしてマークする。  
20

### 【0032】

[0043]ブロック c ( i )の符号化モードが「フレーム内」でない場合、動きベクトル  $m v ( i )$  をメモリに格納する。また、現在のブロックのモード  $c mode ( i )$  もメモリに書き込む。処理ロジックがインデックス変数  $i$  を 1だけインクリメントすることにより、プロセスは現在のフレーム内の次のブロック c ( i )に進み(処理ブロック 211)、 $i$  が N (現在のフレーム内のブロックの総数)より小さいかを調べる(処理ブロック 212)。インデックス  $i$  が N より小さい場合、処理ロジックは処理ブロック 202 に移行し、プロセスはそこから続行する。インデックス  $i$  が N より小さくない場合、処理ロジックは処理ブロック 213 に移行し、インデックス変数  $i$  を 1 に設定する。  
30

### 【0033】

[0044]現在のフレーム内の全てのブロックを上述したように処理した時点で、動き補償プロセスを開始する。動き補償プロセスでは、最初に、処理ロジックが、現在のフレームから元の現在のブロック c ( i )を読み取る(処理ブロック 214)。一実施形態においては、ブロックを連続的に読み取る。処理ロジックは、ブロック c ( i )の符号化モードが「フレーム内」であるかを調べる(処理ブロック 215)。一実施形態においては、これは、MCPM131 の処理ロジックが行う。ブロック c ( i )の符号化モードが「フレーム内」である場合、処理ロジックは処理ブロック 219 に移行する。「フレーム内」ではない場合、処理ロジックは、 $m v ( i , n )$  を使用して動き補償を実行し、基準フレームからブロック予測子( $block predictor$ )  $b ( i )$  をフェッチし(処理ブロック 216)、次式に従って予測誤差  $e ( i )$  を計算する(処理ブロック 217)。  
40

### 【数1】

$$e(i) = c(i) - p(i)$$

この式において、予測ブロック  $p ( i )$  は、MCPM131 がブロック c ( i )用にメモリに格納されている動きベクトル  $m v ( i )$  を使用することによって、生成される。処理ロジックは、動き情報をエントロピ符号化し(処理ブロック 218)、処理ブロック 21  
50

9に移行する。一実施形態においては、現在のブロックに対応する動きベクトルデータをMDPM127に送って処理し(例:近傍のブロックの動きベクトルに対してDPCM符号化する)、エントロピ符号化器モジュール113に送って符号化する。

#### 【0034】

[0045]処理ブロック219において、処理ロジックは、c(i)の符号化モードをエントロピ符号化する。次いで、処理ロジックは、モードm(i)に応じてブロックデータc(i)又はe(i)を変換符号化し(処理ブロック220)、現在のブロックを再構築し(処理ブロック221)、それをメモリ(例:メモリ123)に書き込む(処理ブロック222)。

#### 【0035】

[0046]現在のフレーム内の全てのブロックに対して、上述したプロセスが繰り返される。そのために、次いで、処理ロジックは、インデックスiを1だけインクリメントし(処理ブロック223)、インデックスiがN(現在のフレーム内のブロックの総数)より小さいかを調べる(処理ブロック224)。Nより小さい場合、処理は処理ブロック214に移行し、プロセスはそこから続行する。Nより小さくない場合、処理ロジックは、メモリ(例:メモリ123)内のノイズが含まれるフレームデータをフィルタリングし(処理ブロック225)、フィルタリングしたデータをフレームストア(例:フレームストア126)に格納する(処理ブロック226)。その後、プロセスは終了する。

#### 【0036】

[0047]現在のフレーム内の全てのブロックを処理した後、スイッチ124を閉じる。メモリ123に格納されている再構築されたフレームデータを、以下に説明するようにノイズフィルタモジュール125によって処理する。ノイズフィルタモジュール125は、自身が処理したデータをフレームストア126に格納する。

#### 【0037】

##### <ノイズフィルタリング>

[0048]図3は、ノイズフィルタリングモジュールの一実施形態(例:図1のノイズフィルタリングモジュール125)のブロック図である。以下、図3を参照する。このノイズフィルタリングモジュールは、メモリ301と、空間マスク位置生成モジュール(MPGM)302と、変換モジュール(TM)303と、推定器モジュール(EM)304と、メモリ305と、逆変換モジュール(ITM)306と、メモリ307と、フィルタモジュール(FM)308と、を備えている。メモリ301は、マスク位置生成器302にマスク構造310を提供する。マスク位置生成器302は、これに応答して、フレーム内位置311を示し、これらの位置を、変換モジュール303と、逆変換モジュール306と、フィルタ308とに提供する。変換モジュール303は、復号されたフレームデータ312に変換を適用し、マスク位置生成器302によって示されたマスク下のサンプルの変換係数313を生成する。推定器305は、変換係数313を受け取り、これに応じて、ノイズ低減された変換係数314を生成する。ノイズ低減された変換係数314をメモリ305に格納する。次いで、逆変換モジュール306が、ノイズ低減された変換係数に、マスク位置生成器302からのマスクに基づいて逆変換を適用し、マスク下のノイズ低減されたサンプル315を生成する。これらサンプルはメモリ307に格納される。フィルタモジュール308は、ノイズ低減されたサンプル315に、マスク位置生成器302からのマスクに基づいてフィルタリングを実行し、再構築されたフレームサンプル316を生成する。

#### 【0038】

[0049]ノイズフィルタモジュールへの入力は、メモリモジュール123に格納されている再構築されたフレームデータ312(例:復号されたフレームデータ)から成る。ノイズフィルタモジュールは、フィルタリングされたフレームデータ316(例:再構築されたフレームサンプル)をフレームストア126に出力する。ノイズフィルタモジュールのオペレーションは、図4、図5、図6に記載したフローチャートによって示したように進行する。

## 【0039】

[0050]図4は、ノイズフィルタリングプロセスの実施形態のフローチャートである。このプロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で実行されるもの等）、又は双方の組合せを備えることができる。一実施形態においては、処理ロジックは、図3のノイズフィルタモジュールの推定器304の一部である。

## 【0040】

[0051]ノイズフィルタモジュールが実行するプロセスは、最初のステップとして、メモリ（例：123）に格納されている再構築されたフレームデータを取得する。処理ロジックは、再構築されたフレームデータを使用して、変数Vのセットを計算する（処理プロック401）。これら補助変数Vのセットは、推定プロセスにおいて更に使用される。推定器は、符号化プロセスによって生じる量子化ノイズの量を、コーデックが使用する量子化方式に基づいて計算する。本明細書においては、符号化プロセスにおいて使用する量子化ステップサイズをqによって表す。一実施形態においては、処理ロジックは、補助変数を次式によって計算する

## 【数2】

$$v^2 = \alpha \frac{q^2}{4}$$

10

20

この式において、 $\alpha$ は定数、 $v^2$ はノイズ出力の推定値を表す。推定器の処理ロジックは、符号化器において使用する符号化方式の特性に基づいて、変数bの値も決定する。この場合、bは、 $b = v^2$ として計算する。ここで、 $v$ は定数である。別の実施形態においては、処理ロジックは、追加の補助変数を以下のように計算する。

## 【数3】

$$t_1 = \mu v^2 - (\mu - 1)b$$

$$x_1 = \sqrt{b}$$

$$x_2 = \sqrt{t_1}$$

$$z_1 = \frac{1}{x_2 - x_1}$$

$$z_2 = x_1 z_1$$

30

これ、 $\mu$ は選択可能な定数である。

## 【0041】

[0052]更に別の実施形態においては、処理ロジックは、以下の補助変数を計算する。

40

## 【数4】

$$t_1 = \mu v^2 - (\mu - 1)b$$

$$x_1 = \sqrt{b}$$

$$x_2 = \sqrt{t_1}$$

$$y_2 = \frac{\mu}{\mu + 1} x_2$$

$$h_1 = \frac{y_2}{x_2 - x_1}$$

$$h_2 = \frac{y_2 x_1}{x_2 - x_1}$$

10

ここで、 $\mu$ は選択可能な定数である。

## 【0042】

[0053]次いで、処理ロジックは、サポート領域Rを有するマスクSを設定する（処理ブロック402）。マスクのサイズ（サポート領域R）は、例えば、符号化プロセスにおいて使用するブロックのサイズに等しくすることができる。 20

## 【0043】

[0054]その後、処理ロジックは、(i, j)をフレーム内のブロックの位置に初期化する（処理ブロック403）。このブロック位置(i, j)は、ビデオコーデックがフレームを符号化／復号するのに慣習的に使用されるブロックのグリッド上に位置しており、メモリ（例：メモリ123）からの現在フィルタリングしているフレーム内で選択する。

## 【0044】

[0055]更に、処理ロジックは、(m, n)を、(i, j)を含んだその周囲の位置に初期化する（処理ブロック404）。位置(m, n)は、マスクSを(i, j)に対してシフトさせる位置を表す。次いで、処理ロジックは、マスクSをフレーム内の位置(m, n)に移動させる（処理ブロック405）。 30

## 【0045】

[0056]処理ロジックは、マスクS下のサンプルに変換T（例：離散コサイン変換（DCT））を適用し（処理ブロック406）、変換後の係数に対し、推定器の処理ロジックを使用してノイズ低減を行う（処理ブロック407）。この実施形態では、マスクSの現在の位置によってカバーされるデータサンプルに対応するN個の変換係数を、s(k)（k=1, ..., N）によって表す。結果としての変換係数を、推定器モジュールの処理ロジックによって処理する。図7及び図8のフローチャートは、この動作の相異なる実施形態を示しており、以下ではこれらの実施形態について更に詳しく説明する。

## 【0046】

40

[0057]図7のフローチャートでは、推定器モジュールの処理ロジックは、変換係数s(k)を取得し、値s<sup>2</sup>(k)を計算することによって信号パワーを推定する。次いで、処理ロジックは、この信号パワーを閾値と比較し、係数s(k)に対応する推定値a(k)を以下のように求める。

## 【数5】

```

if       $s^2(k) \leq b$ 
       $a(k) = 0;$ 
else

$$a(k) = \frac{s^2(k) - b}{s^2(k) + v^2 - 2b}$$


```

10

## 【0047】

[0058]次いで、推定器の処理ロジックは、次式に従って、処理後の変換係数  $p(k)$  を求める。

## 【数6】

$$p(k) = a(k)s(k)$$

## 【0048】

[0059]代替の実施形態においては、推定器の処理ロジックは、図8のフローチャートに示したプロセスを使用し、前に定義した補助変数  $V$  を利用し、係数  $s(k)$  のそれぞれの推定値を、以下のように計算する。 20

## 【数7】

```

if       $s^2(k) \leq b$ 
       $a(k) = 0;$ 
else if  $b < s^2(k) \leq t_1$ 
       $a(k) = z_1 s(k) - z_2$ 
else
       $a(k) = 1$ 

```

20

30

## 【0049】

[0060]次いで、推定器の処理ロジックは、処理後の変換係数  $p(k)$  を以下のように求める。

## 【数8】

$$p(k) = a(k)s(k)$$

40

## 【0050】

[0061]代替の実施形態においては、推定器の処理ロジックは、同じく図8のフローチャートに示したプロセスを使用し、係数  $s(k)$  のそれれに対して、処理後の係数  $p(k)$  の値を以下のように計算する。

## 【数9】

```

if       $s^2(k) \leq b$ 
       $p(k) = 0;$ 

else if  $b < s^2(k) \leq t_1$ 
       $p(k) = h_1 s(k) - h_2;$ 

else
       $p(k) = s(k)$ 

```

10

図8については、後に更に詳しく説明する。

## 【0051】

[0062]図7又は図8のどちらのプロセスを使用した場合にも、処理ロジックは、係数 $p(k)$ をメモリ（例：メモリ305）に格納する。全ての係数を処理していなければ、上述したプロセスを繰り返す。

## 【0052】

[0063]次いで、処理ロジックは、マスクS下の処理済み変換係数 $p(k)$ に逆変換 $T^{-1}$ （例：逆DCT変換）を適用し（処理ブロック408）、フレームにおけるマスクSの現在の位置 $(m, n)$ に対応するサンプル $r(k)$ を得る。処理ロジックは、マスクS下のこれらの処理済みサンプルをメモリ（例：メモリ307）に格納する（処理ブロック409）。

20

## 【0053】

[0064]次いで、処理ロジックは、 $(m, n)$ を次の位置に設定し（処理ブロック410）、サポート領域R内の全ての位置 $(m, n)$ を処理したかを調べる（処理ブロック411）。ブロック位置 $(i, j)$ の周囲のマスクSの位置の全てを処理していない場合、処理は処理ブロック405に移行し、 $(i, j)$ に対するマスクSの次のシフト位置 $(m, n)$ を選択し、上述したプロセスを繰り返す。

## 【0054】

30

[0065]ブロック位置 $(i, j)$ の周囲のマスクSの位置の全てを処理した場合、処理ブロックは処理ブロック412に移行し、フレーム内のブロックグリッド上の次のブロック位置 $(i, j)$ を選択し、フレーム内の処理中のブロックのブロック位置 $(i, j)$ の全てを処理したかを調べる（処理ブロック413）。全てを処理していない場合、処理ロジックは処理ブロック404に移行し、プロセスはそこから続行する。全てを処理した場合、処理は処理ブロック414に移行する。このようにして、ブロック位置 $(i, j)$ の全てを処理するまで、上記のプロセスを繰り返す。

## 【0055】

[0066]処理ブロック414において、処理ロジックは、フレーム内のサンプル位置のそれぞれに対応する再構築バージョンを、フィルタ（例：フィルタ308）を使用してフィルタリングし（処理ブロック414）、処理している現在のフレームのフィルタリングされたデータサンプル $f(x, y)$ を、フレームストア（例：フレームストア126）に書き込む（処理ブロック415）。一実施形態においては、処理ロジックは、現在のフレーム内のサンプルのそれぞれに対応する、メモリ（例：メモリ307）に格納されている処理済みサンプルをフィルタリングする。一実施形態においては、このフィルタリングは、ノイズフィルタモジュール125のフィルタ308が実行する。従って、ノイズフィルタモジュール125によって処理している現在のフレーム内の位置 $(x, y)$ のそれぞれに対して、推定器304によってメモリ307に格納されている、複数の処理済みの再構築されたサンプル $r^l(x, y)$ が存在する。ここで、 $l = 1, \dots, L$ は、マスクSがフレーム内のブロックグリッド位置の周囲を占有できる位置をインデックス化しており、

40

50

そのような位置のそれぞれにおいて、マスク S のサポート領域 R にサンプル位置 (x, y) が含まれる。一実施形態においては、フィルタは、フレーム位置 (x, y) に対応する入力データ  $r^l$  (x, y) に対して、以下のように加重平均演算を実行する。

【数 1 0】

$$f(x, y) = \sum_l w_l r^l(x, y)$$

この式において、 $w_1$  は、推定器の処理ロジックによって生成される統計情報に基づいて選択する重みである。一実施形態においては、全ての重みは、 $w_1 = 1 / L$  として選択され、平均化演算に相当するものとする。代替の実施形態においては、重みは、マスク S の位置 1 において推定器を適用することで得られる処理後の係数  $p(k)$  のうち値がゼロである係数の数に比例するように選択する。

10

【0 0 5 6】

[0067] 次いで、プロセスが終了する。

【0 0 5 7】

[0068] 図 5 は、ノイズフィルタリングモジュールによって実行されるノイズフィルタリングプロセスの代替の実施形態のフローチャートである。このプロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で実行されるもの等）、又は双方の組合せを備えることができる。一実施形態においては、処理ロジックは、図 3 のノイズフィルタモジュールの推定器 304 の一部である。

20

【0 0 5 8】

[0069] 以下、図 5 を参照する。このプロセスでは、最初に、処理ロジックが、サポート領域 R を有するマスク S を生成する（処理ブロック 501）。マスクのサイズ（サポート領域 R）は、例えば、符号化プロセスにおいて使用するブロックのサイズに等しくすることができる。

【0 0 5 9】

[0070] 処理ロジックは、変数  $V(k)$  のセットを計算する（処理ブロック 502）。変換（例：DCT）をマスク S 下のサンプルに適用する（処理ブロック 303 において）ので、そのような変換の場合に慣習的に行われるよう、変換係数を順位付けることができる。この順位を  $k$  によって表す。この実施形態においては、変数  $v(k)$  は、マスク S 下で生成される変換係数の順位  $k$  に依存する。この補助変数  $v(k)$  のセットは、推定プロセスにおいて更に使用される。処理ロジックは、コーデックが使用する量子化方式に基づいて、符号化プロセスによって生じる量子化ノイズの量を計算する。本明細書においては、符号化プロセスにおいて使用する量子化ステップサイズを  $q$  によって表す。この場合、処理ロジックは、変数を次式によって計算する。

30

【数 1 1】

$$v^2(k) = \alpha(k) \frac{q^2}{4}$$

40

この式において、 $\alpha(k)$  は、様々な要因（例：マスク S の内側の係数  $s(k)$  の順位  $k$ ）に基づいて適合させることができる（この場合は  $\alpha(k) = s(k)$ ）。例えば、順位の高い係数に対しては  $s(k)$  を大きくする。 $s(k)$  が  $k$  の関数であるならば、 $b$  も係数の順位  $k$  の関数となり、 $b(k) = s(k)^2$  である。別の実施形態においては、処理ロジックは、マスク S (S の中の位置) 下の変換係数の順位  $k$  に依存する追加の補助変数を以下のように計算する。

50

## 【数12】

$$\begin{aligned}t_1(k) &= \mu v^2(k) - (\mu - 1)b(k) \\x_1(k) &= \sqrt{b(k)} \\x_2(k) &= \sqrt{t_1(k)} \\z_1(k) &= \frac{1}{x_2(k) - x_1(k)} \\z_2(k) &= x_1(k)z_1(k)\end{aligned}$$

10

上式において、 $\mu$  は選択可能な定数である。

## 【0060】

[0071]或いは、別の実施形態においては、処理ロジックは以下の補助変数を計算する。

## 【数13】

$$\begin{aligned}t_1(k) &= \mu v^2(k) - (\mu - 1)b(k) \\x_1(k) &= \sqrt{b(k)} \\x_2(k) &= \sqrt{t_1(k)} \\y_2(k) &= \frac{\mu}{\mu + 1} x_2(k) \\h_1(k) &= \frac{y_2(k)}{x_2(k) - x_1(k)} \\h_2(k) &= \frac{y_2(k)x_1(k)}{x_2(k) - x_1(k)}\end{aligned}$$

20

上式において、 $\mu$  は選択可能な定数である。

## 【0061】

[0072]次いで、処理ロジックは、(i, j)を、フレーム内のブロックの位置に初期化する（処理ブロック503）。一実施形態においては、ビデオコーデックがフレームを符号化 / 復号するのに慣習的に使用するブロックのグリッド上に位置しているブロック位置(i, j)を、メモリ（例：メモリ123）からの現在フィルタリングしているフレーム内で選択する。

## 【0062】

[0073]更に、処理ロジックは、(m, n)を、(i, j)を含んだその周囲の位置に初期化する（処理ブロック504）。位置(m, n)は、マスクSを(i, j)に対してシフトさせる位置である。次いで、処理ロジックは、マスクSをフレーム内の位置(m, n)に移動させる（処理ブロック505）。

40

## 【0063】

[0074]処理ロジックは、マスクS下のサンプルに変換T（例：DCT）を適用し（処理ブロック506）、変換係数に対してノイズ低減を行う（処理ブロック507）。この実施形態では、マスクSの現在の位置によってカバーされるデータサンプルに対応するN個の変換係数を、s(k) (k = 1, ..., N)によって表す。結果としての変換係数を、推定器モジュールの処理ロジックによって処理する。図7及び図8のフローチャートは、このオペレーションの相異なる実施形態を示しており、以下ではこれらの実施形態について更に詳しく説明する。

50

## 【0064】

[0075]図7のフローチャートでは、推定器モジュールの処理ロジックは、変換係数  $s(k)$  を取得し、値  $s^2(k)$  を計算することによって信号パワーを推定する。次いで、処理ロジックは、この信号パワーを閾値と比較し、係数  $s(k)$  に対する推定値を以下のように求める。

## 【数14】

```

if       $s^2(k) \leq b(k)$ 
       $a(k) = 0;$ 
else
      
$$a(k) = \frac{s^2(k) - b(k)}{s^2(k) + v^2(k) - 2b(k)}$$


```

10

## 【0065】

[0076]次いで、推定器の処理ロジックは、次式に従って、処理後の変換係数  $p(k)$  を求める。

## 【数15】

$$p(k) = a(k)s(k)$$

20

## 【0066】

[0077]代替の実施形態においては、推定器の処理ロジックは、図8のフローチャートに示したプロセスを使用し、補助変数を利用して、係数  $s(k)$  のそれぞれの推定値を以下のように計算する。

## 【数16】

```

if       $s^2(k) \leq b(k)$ 
       $a(k) = 0;$ 
else if  $b(k) < s^2(k) \leq t_1(k)$ 
      
$$a(k) = z_1(k)s(k) - z_2(k)$$

else
       $a(k) = 1$ 

```

30

## 【0067】

[0078]次いで、推定器の処理ロジックは、処理後の変換係数  $p(k)$  を以下のように求める。

## 【数17】

$$p(k) = a(k)s(k)$$

40

## 【0068】

[0079]代替の実施形態においては、推定器の処理ロジックは、同じく図8のフローチャートに示したプロセスを使用し、処理後の係数  $p(k)$  の値を以下のように計算する。

50

## 【数18】

```

if      s2(k) <= b(k)
      p(k) = 0;

else if b(k) < s2(k) <= t1(k)
      p(k) = h1(k)s(k) - h2(k);

else
      p(k) = s(k)

```

10

## 【0069】

[0080]図8については、後に更に詳しく説明する。

## 【0070】

[0081]図7又は図8のどちらのプロセスを使用した場合にも、処理ロジックは、係数p(k)を(例えばメモリ305に)格納する。全ての係数を処理していなければ、上述したプロセスを繰り返す。

## 【0071】

[0082]ノイズ低減を行った後、処理ロジックは、マスクS下の変換係数p(k)に逆変換T<sup>-1</sup>(例:逆DCT変換)を適用し(処理ブロック508)、フレームにおけるマスクSの現在の位置(m, n)に対応するサンプルr(k)を得て、マスクS下の再構築されたサンプルをメモリ(例:メモリ307)に格納する(処理ブロック509)。

20

## 【0072】

[0083]次いで、処理ロジックは、(m, n)を次の位置に設定し(処理ブロック510)、サポート領域R内の全ての位置(m, n)を処理したかを調べる(処理ブロック511)。ブロック位置(i, j)の周囲のマスクSの位置の全てを処理していない場合、処理は処理ブロック505に移行し、(i, j)に対するマスクSの次のシフト位置(m, n)を選択し、上述したプロセスを繰り返す。

## 【0073】

30

[0084]ブロック位置(i, j)の周囲のマスクSの位置の全てを処理した場合、処理ブロックは処理ブロック512に移行し、フレーム内のブロックグリッド上の次のブロック位置(i, j)を選択し、フレーム内の処理中のブロックのブロック位置(i, j)の全てを処理したかを調べる(処理ブロック513)。全てを処理していない場合、処理ロジックは処理ブロック504に移行し、プロセスはそこから続行する。全てを処理した場合、処理は処理ブロック514に移行する。このようにして、ブロック位置(i, j)の全てを処理するまで、上記のプロセスを繰り返す。

## 【0074】

[0085]処理ブロック514において、処理ロジックは、フレーム内のサンプル位置のそれに対応する再構築されたバージョンを、フィルタを使用してフィルタリングする(処理ブロック514)。一実施形態においては、使用するフィルタはフィルタ308である。従って、ノイズフィルタモジュール125によって処理している現在のフレーム内の位置(x, y)のそれに対しても、推定器によってメモリ307に格納されている、複数の処理済みの再構築されたサンプルr<sup>1</sup>(x, y)が存在する。ここで、l = 1, . . . , Lは、マスクSがフレーム内のブロックグリッド位置の周囲を占有できる位置をインデックス化しており、そのような位置のそれにおいて、マスクSのサポート領域Rにサンプル位置(x, y)が含まれる。一実施形態においては、フィルタ308は、フレーム位置(x, y)に対応する入力データr<sup>1</sup>(x, y)に対して、以下のように加重平均演算を実行する。

40

## 【数19】

$$f(x, y) = \sum_l w_l r^l(x, y)$$

この式において、 $w_1$  は、推定器によって生成される統計情報に基づいて選択する重みである。一実施形態においては、全ての重みを、 $w_1 = 1 / L$  として選択する。代替の実施形態においては、重みは、マスク S の位置 1 において推定器を適用した結果としての処理後の係数 p(k) のうち値がゼロである係数の数に比例するように選択する。

10

## 【0075】

[0086] 処理ロジックは、処理している現在のフレームに用に得られたフィルタリング済みのデータサンプル  $f(x, y)$  を、フレームストア（例：フレームストア 126）に書き込む（処理ブロック 515）。

## 【0076】

[0087] その後、プロセスが終了する。

## 【0077】

[0088] 図 6 は、ノイズフィルタリングモジュールによって実行されるノイズフィルタリングプロセスの更に別の代替実施形態のフローチャートである。このプロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で実行されるもの等）、又は双方の組合せを備えることができる。

20

## 【0078】

[0089] 以下、図 6 を参照し、このプロセスでは、最初に、処理ロジックが、サポート領域 R を有するマスク S を生成する（処理ブロック 601）。一実施形態においては、マスクのサイズ（サポート領域 R）は、符号化プロセスにおいて使用するブロックのサイズに等しくすることができる。

## 【0079】

[0090] 次いで、処理ロジックは、(i, j) を、フレーム内のブロックの位置に初期化する（処理ブロック 602）。一実施形態においては、ビデオコーデックがフレームを符号化／復号するのに慣習的に使用するブロックのグリッド上に位置しているブロック位置 (i, j) は、メモリ（例：メモリ 123）からの現在フィルタリングしているフレーム内で選択する。

30

## 【0080】

[0091] 更に、処理ロジックは、(m, n) を、(i, j) を含んだその周囲の位置に初期化する（処理ブロック 603）。位置 (m, n) は、マスク S を (i, j) に対してシフトさせる位置である。処理ロジックは、マスク S をフレーム内の位置 (m, n) に移動させる（処理ブロック 604）。

## 【0081】

[0092] 推定器の処理ロジックは、現在のマスク位置において 303 によって生成される変換係数の順位 k と、フレーム内のマスク S の位置 (m, n) とに依存する変数 V(m, n, k) のセットを計算する（処理ブロック 605）。これら補助変数のセットは、推定プロセスにおいて更に使用される。処理ロジックは、符号化プロセスによって生じる量子化ノイズの量を、コーデックが使用する量子化方式に基づいて計算する。本明細書においては、符号化プロセスにおいて使用する量子化ステップサイズを q によって表す。この場合、処理ロジックは、変数を以下のように計算する。

40

## 【数20】

$$V^2(m, n, k) = \alpha(m, n, k) \frac{q^2}{4}$$

50

この式において、 $\nu$ は、様々な要因（例：マスク S の内側の係数  $s(k)$  の順位  $k$ 、 $(i, j)$  の周囲のマスク S の位置  $(m, n)$ ）に基づいて適合させることができる（この場合、 $\nu = (m, n, k)$ ）。例えば、順位の高い係数に対しては  $\nu$  を大きくする。 $\nu$  が  $(m, n, k)$  の関数であるならば、 $b$  も係数の順位  $(m, n, k)$  の関数となり、 $b(m, n, k) = \nu^2(m, n, k)$  である。別の実施形態においては、処理ロジックは、マスク S (S の中の位置) にマッピングされる変換係数の順位  $(m, n, k)$  に依存する追加の補助変数を、以下のように計算する。

【数 2 1】

10

$$\begin{aligned} t_1(m, n, k) &= \mu\nu^2(m, n, k) - (\mu-1)b(m, n, k) \\ x_1(m, n, k) &= \sqrt{b(m, n, k)} \\ x_2(m, n, k) &= \sqrt{t_1(m, n, k)} \\ z_1(m, n, k) &= \frac{1}{x_2(m, n, k) - x_1(m, n, k)} \\ z_2(m, n, k) &= x_1(m, n, k)z_1(m, n, k) \end{aligned}$$

20

上式において、 $\mu$  は選択可能な定数である。

【0 0 8 2】

[0093] 或いは、別の実施形態においては、処理ロジックは以下の補助変数を計算する。

【数 2 2】

$$\begin{aligned} t_1(m, n, k) &= \mu\nu^2(m, n, k) - (\mu-1)b(m, n, k) \\ x_1(m, n, k) &= \sqrt{b(m, n, k)} \\ x_2(m, n, k) &= \sqrt{t_1(m, n, k)} \\ y_2(m, n, k) &= \frac{\mu}{\mu+1}x_2(m, n, k) \\ h_1(m, n, k) &= \frac{y_2(m, n, k)}{x_2(m, n, k) - x_1(m, n, k)} \\ 3 \ h_2(m, n, k) &= \frac{y_2(m, n, k)x_1(m, n, k)}{x_2(m, n, k) - x_1(m, n, k)} \end{aligned}$$

30

上式において、 $\mu$  は選択可能な定数である。

40

【0 0 8 3】

[0094] 処理ロジックは、マスク S 下のサンプルに変換 T (例：DCT) を適用し (処理ブロック 606)、変換後の係数に対し、推定器の処理ロジックを使用してノイズ低減を行う (処理ブロック 607)。この実施形態では、マスク S の現在の位置によってカバーされるデータサンプルに対応する N 個の変換係数を、 $s(m, n, k)$  ( $k = 1, \dots, N$ ) によって表す。結果としての変換係数を、推定器モジュールの処理ロジックによって処理する。図 7 及び図 8 のフローチャートは、このオペレーションの相異なる実施形態を示しており、以下ではこれらの実施形態について更に詳しく説明する。

【0 0 8 4】

[0095] 図 7 のフローチャートでは、推定器モジュールの処理ロジックは、変換係数  $s$  (

50

$m, n, k$  ) を取得し、値  $s^2 (m, n, k)$  を計算することによって信号パワーを推定する。次いで、処理ロジックは、この信号パワーを閾値と比較し、係数  $s (m, n, k)$  に対する推定値を以下のように求める。

## 【数23】

if       $s^2(m, n, k) \leq b(m, n, k)$

$a(m, n, k) = 0;$

else

$$a(m, n, k) = \frac{s^2(m, n, k) - b(m, n, k)}{s^2(m, n, k) + v^2(m, n, k) - 2b(m, n, k)}$$

10

## 【0085】

[0096] 次いで、推定器の処理ロジックは、次式に従って、処理後の変換係数  $p (m, n, k)$  を求める。

## 【数24】

$$p(m, n, k) = a(m, n, k)s(m, n, k)$$

20

## 【0086】

[0097] 代替の実施形態においては、推定器の処理ロジックは、図8のフローチャートに示したプロセスを使用し、補助変数を利用し、係数  $s (m, n, k)$  のそれぞれの推定値を、以下のように計算する。

## 【数25】

if       $s^2(m, n, k) \leq b(m, n, k)$

$a(m, n, k) = 0;$

else if     $b(m, n, k) < s^2(m, n, k) \leq t_1(m, n, k)$

$a(m, n, k) = z_1(m, n, k)s(m, n, k) - z_2(m, n, k)$

else

$a(m, n, k) = 1$

30

## 【0087】

[0098] 次いで、推定器の処理ロジックは、処理後の変換係数  $p (m, n, k)$  を以下のように求める。

40

## 【数26】

$$p(m, n, k) = a(m, n, k)s(m, n, k)$$

## 【0088】

[0099] 代替の実施形態においては、推定器の処理ロジックは、同じく図8のフローチャートに示したプロセスを使用し、処理後の係数  $p (m, n, k)$  の値を以下のように計算する。

## 【数27】

```

if       $s^2(m, n, k) \leq b(m, n, k)$ 
       $p(m, n, k) = 0;$ 

else if  $b(m, n, k) < s^2(m, n, k) \leq t_1(m, n, k)$ 
       $p(m, n, k) = h_1(m, n, k)s(m, n, k) - h_2(m, n, k)$ 

else
       $p(m, n, k) = s(m, n, k)$ 

```

10

## 【0089】

[00100]図8については、後に更に詳しく説明する。

## 【0090】

[00101]図7又は図8のどちらのプロセスを使用した場合にも、処理ロジックは、係数 $p(k)$ をメモリ（例：メモリ305）に格納する。全ての係数が処理されていなければ、上述したプロセスを繰り返す。

## 【0091】

[00102]次いで、処理ロジックは、マスクS下の変換係数に逆変換 $T^{-1}$ を適用し（処理ブロック608）、フレームにおけるマスクSの現在の位置（m, n）に対応するサンプル $r(m, n, k)$ を得る。処理ロジックは、マスクS下のこれらの再構築されたサンプルをメモリ（例：メモリ307）に格納する（処理ブロック609）。

20

## 【0092】

[00103]次いで、処理ロジックは、（m, n）を次の位置に設定し（処理ブロック610）、サポート領域R内の全ての位置（m, n）を処理したかを調べる（処理ブロック611）。ブロック位置（i, j）の周囲のマスクSの位置の全てを処理していない場合、処理は処理ブロック605に移行し、（i, j）に対するマスクSの次のシフト位置（m, n）を選択し、上述したプロセスを繰り返す。

## 【0093】

30

[00104]ブロック位置（i, j）の周囲のマスクSの位置の全てを処理した場合、処理ブロックは処理ブロック612に移行し、フレーム内のブロックグリッド上の次のブロック位置（i, j）を選択し、フレーム内の処理中のブロックのブロック位置（i, j）の全てを処理したかを調べる（処理ブロック613）。全てを処理していない場合、処理ロジックは処理ブロック604に移行し、プロセスはそこから続行する。全てを処理した場合、処理は処理ブロック614に移行する。このようにして、ブロック位置（i, j）の全てを処理するまで、上記のプロセスを繰り返す。

## 【0094】

[00105]処理ブロック614において、処理ロジックは、フレーム内のサンプル位置のそれに対応する再構築されたバージョンを、フィルタ（例：フィルタ308）を使用してフィルタリングし（処理ブロック614）、処理している現在のフレームのフィルタリングされたデータサンプル $f(x, y)$ を、フレームストア（例：フレームストア126）に書き込む（処理ブロック615）。一実施形態においては、処理ロジックは、現在のフレーム内の各サンプルに対応する、メモリ（例：メモリ307）に格納されている推定されたサンプルをフィルタリングする。一実施形態においては、このフィルタリングは、ノイズフィルタモジュール125のフィルタ308が実行する。従って、ノイズフィルタモジュール125によって処理している現在のフレーム内の位置（x, y）のそれに対して、推定器304によってメモリに格納されている、複数の処理済みの再構築されたサンプル $r^1(x, y)$ が存在する。ここで、 $l = 1, \dots, L$ は、マスクSがフレーム内のブロックグリッド位置の周囲を占有できる位置をインデックス化しており、その

40

50

のような位置のそれぞれにおいて、サンプル位置 ( $x, y$ ) が含まれる。一実施形態においては、フィルタは、フレーム位置 ( $x, y$ ) に対応する入力データ  $r^l(x, y)$  に対して、以下のように加重平均演算を実行する。

**【数 2 8】**

$$f(x, y) = \sum_l w_l r^l(x, y)$$

この式において、 $w_1$  は、推定器の処理ロジックによって生成される統計情報に基づいて選択する重みである。一実施形態においては、全ての重みを、 $w_1 = 1 / L$  として選択する。代替実施形態においては、重みは、マスク S の位置 1 において推定器を適用した結果としての処理後の係数  $p(k)$  のうち値がゼロである係数の数に比例するように選択する。

**【0 0 9 5】**

[00106] 次いで、プロセスが終了する。

**【0 0 9 6】**

<推定プロセスの例>

[00107] 図 7 は、推定プロセスの一実施形態のフローチャートである。推定プロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で実行されるもの等）、又は双方の組合せを備えることができる。

**【0 0 9 7】**

[00108] 以下、図 7 を参照する。このプロセスでは、処理ロジックは、最初に変換係数を取得し（処理ブロック 701）、変換係数の信号パワーを推定する（処理ブロック 702）。一実施形態においては、変換係数の二乗を計算することによって信号パワーを推定する。

**【0 0 9 8】**

[00109] 次いで、処理ロジックは、信号パワーが閾値よりも小さいかを調べる（処理ブロック 703）。閾値の一実施形態は、上の段落[0052]に説明したものである。閾値よりも小さい場合に、処理ロジックは、係数を 0 に設定し（処理ブロック 704）、処理が処理ブロック 706 に移行し、係数を格納する。閾値よりも小さくない場合、処理ロジックは、信号パワー、ノイズ出力、及び閾値に基づいて、上述したロジックを実行することによって係数の値を調節し（処理ブロック 705）、処理が処理ブロック 706 に移行し、係数を格納する。

**【0 0 9 9】**

[00110] 処理ロジックは、係数を格納した後、全ての係数を処理したかを調べる（処理ブロック 707）。処理していない場合、処理ロジックは処理ブロック 701 に移行し、プロセスはこのポイントから続行する。全ての係数を処理した場合、プロセスは終了する。

**【0 1 0 0】**

[00111] 図 8 は、推定プロセスの代替の実施形態のフローチャートである。このプロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア（回路、専用ロジック等）、ソフトウェア（汎用コンピュータシステム又は専用マシン上で実行されるもの等）、又は双方の組合せを備えることができる。

**【0 1 0 1】**

[00112] 以下、図 8 を参照する。このプロセスでは、処理ロジックは、最初に変換係数を取得し（処理ブロック 801）、変換係数の信号パワーを推定する（処理ブロック 802）。一実施形態においては、変換係数の二乗を計算することによって信号パワーを推定する。次いで、処理ロジックは、信号パワーが閾値よりも小さいかを調べる（処理ブロック 803）。

10

20

30

40

50

ク 8 0 3 )。閾値の一実施形態は、上の段落[0071]に説明したものである。閾値よりも小さい場合、処理ロジックは、係数を 0 に設定し(処理ブロック 8 0 4 )、処理が処理ブロック 8 0 6 に移行し、係数を格納する。閾値よりも小さくない場合、処理ロジックは、信号パワーが第 2 の閾値よりも小さいかを調べる(処理ブロック 8 0 5 )。この閾値の実施形態は、上の段落[0093]に説明したものである。第 2 の閾値よりも小さい場合、処理ロジックは、信号パワー、ノイズ出力、及び閾値に基づいて、上述したロジックを実行することによって係数の値を調節し(処理ブロック 8 0 6 )、処理が処理ブロック 8 0 8 に移行し、係数を格納する。第 2 の閾値よりも小さくない場合、処理ロジックは、係数を変更せずそのままにし(処理ブロック 8 0 7 )、処理ブロック 8 0 8 に移行して係数を格納する。

#### 【 0 1 0 2 】

[00113]処理ロジックは、係数を格納した後、全ての係数を処理したかを調べる(処理ブロック 8 0 9 )。処理していない場合、処理ロジックは処理ブロック 7 0 1 に移行し、プロセスはこのポイントから続行する。全ての係数を処理した場合、プロセスは終了する。

#### 【 0 1 0 3 】

<ビデオ復号器の例>

[00114]図 9 は、ビデオ復号器( V D )の一実施形態のブロック図である。以下、図 9 を参照する。ビデオ復号器は、変換復号器 9 1 3 と、加算器 9 0 4 と、メモリモジュール 9 0 5 と、動き補償・予測モジュール( M C P M ) 9 0 7 と、動きデータ処理モジュール( M D P M ) 9 0 8 と、フレームストア 9 0 9 と、ノイズフィルタモジュール( N F M ) 9 0 6 と、スイッチ 9 4 1 ~ 9 4 3 と、を備えている。一実施形態においては、変換復号器 9 1 3 は、エントロピ復号器 9 0 1 と、逆量子化器 9 0 2 と、逆変換モジュール 9 0 3 と、を含んでいる。

#### 【 0 1 0 4 】

[00115]ビデオ復号器は、図 1 0 のフローチャートを参照しながら以下に説明するプロセスに従って、ビデオフレームデータを再構築する。スイッチ 9 4 2 及びスイッチ 9 4 3 は、従来の符号化プロセスの動作と同様に、復号するブロックのタイプ(フレーム内符号化されている、又は予測符号化されている)に応じて開いているか、又は閉じている。スイッチ 9 4 1 は、最初は開いている。

#### 【 0 1 0 5 】

[00116]図 1 0 は、ビデオ復号プロセスの一実施形態のフローチャートである。このプロセスは処理ロジックによって実行される。処理ロジックは、ハードウェア(回路、専用ロジック等)、ソフトウェア(汎用コンピュータシステム又は専用マシン上で実行されるもの等)、又は双方の組合せを備えることができる。

#### 【 0 1 0 6 】

[00117]以下、図 1 0 を参照する。このプロセスは、最初に、処理ロジックが、インデックス変数  $i$  を 1 に初期化する(処理ブロック 1 0 0 1 )。

#### 【 0 1 0 7 】

[00118]次いで、処理ロジックは、復号するビデオフレーム内の現在のブロック  $c(i)$  の符号化モードを復号し(処理ブロック 1 0 0 2 )、ブロック  $c(i)$  の符号化モードが「フレーム内」であるかを調べる(処理ブロック 1 0 0 3 )。「フレーム内」である場合、処理ロジックは、従来のプロセスに従って現在のブロック  $c(i)$  をフレーム内復号し(処理ブロック 1 0 0 4 )、処理ブロック 1 0 0 9 に移行する。「フレーム内」ではない場合、処理ロジックは、現在のブロック  $c(i)$  のブロック予測誤差  $e(i)$  を変換復号器 9 1 3 を使用して復号し(処理ブロック 1 0 0 5 )、現在のブロック  $c(i)$  の動きベクトル  $m v(i)$  を変換復号器 9 1 3 と M D P M 9 0 8 とを使用して復号し(処理ブロック 1 0 0 6 )、 $m v(i)$  とフレームストア内の基準フレームとを使用して現在のブロック  $c(i)$  の動き補償を行って予測ブロック  $u(i)$  を求める(処理ブロック 1 0 0 7 )。一実施形態においては、復号された動きベクトル  $m v(i)$  を M C P M 9 0 8 が使用して、フレームストア 9 0 9 に格納されている基準フレームから基準ブロック(予測子)

10

20

30

40

50

$u(i)$  をフェッチする。

【0108】

[00119] 次いで、処理ロジックは、次式に従って現在のブロックを再構築する（処理ブロック1008）。

【数29】

$$rc(i) = u(i) + e(i)$$

即ち、現在のブロック  $rc(i)$  は、予測誤差  $e(i)$  を予測サンプル  $u(i)$  に加算することによって再構築される。 10

【0109】

[00120] その後、処理ロジックは、再構築されたデータをメモリ（例：メモリ905）に書き込む（処理ブロック1009）。すなわち、再構築されたブロック  $rc(i)$  に対応するデータをメモリに書き込む。次いで、処理ロジックは、 $i$  を 1 だけインクリメントすることによって、復号している現在のフレーム内の次のブロック位置に進む（処理ブロック1010）。

【0110】

[00121] 処理ロジックは、インデックス変数  $i$  が  $N$ （処理している現在のフレーム内のブロックの総数）より小さいかを調べる（処理ブロック1011）。小さい場合、処理ロジックは処理ブロック1002に移行し、プロセスはそのポイントから続行する。このようにして、復号している現在のフレーム内の全てのブロックについて、上述したプロセスを繰り返す。小さくない場合、スイッチ941を閉じ、処理ロジックは、メモリ（例：メモリ905）内のフレームデータをノイズフィルタモジュール（例：ノイズフィルタモジュール906）を使用してフィルタリングする（処理ブロック1012）。一実施形態においては、ループフィルタリングモジュールが、メモリモジュール905に格納されている復号されたフレームデータを読み取り、ビデオ符号化器におけるループフィルタモジュール125について説明したプロセスと同じプロセスを実行する。次いで、処理ロジックは、フィルタリングされたデータをフレームストア（例：フレームストア909）に書き込む（処理ブロック1013）。 30

【0111】

[00122] その後、プロセスが終了する。

【0112】

[00123] したがって、本明細書に説明した方法及び装置を使用して求められるビデオフレームデータのフィルタリングでは、従来技術のソリューションと比較して、客観的及び主観的な（画質の）パフォーマンスが向上する。

【0113】

<コンピュータシステムの例>

[00124] 図11は、本明細書に説明した一つ以上のオペレーションを実行することのできる例示的なコンピュータシステムのブロック図である。以下、図11を参照する。コンピュータシステム1100は、例示的なクライアントコンピュータシステム又はサーバコンピュータシステムを備えることができる。コンピュータシステム1100は、情報を伝えるための通信メカニズム又はバス1111と、バス1111に結合されており情報を処理するプロセッサ1112と、を備えている。プロセッサ1112はマイクロプロセッサを含んでいるが、マイクロプロセッサ（例：Pentium™、Power PC™、Alphah™）に限定されない。 40

【0114】

[00125] システム1100は、バス1111に結合されているおり、情報と、プロセッサ1112によって実行される命令とを格納するRAM（ランダムアクセスメモリ）又はその他の動的記憶装置1104（メインメモリと称する）を更に備えている。メインメモ

10

20

30

40

50

リ 1 1 0 4 は、プロセッサ 1 1 1 2 による命令の実行時に一時変数又はその他の中間情報を格納する目的にも使用することができる。

#### 【 0 1 1 5 】

[00126]更に、コンピュータシステム 1 1 0 0 は、バス 1 1 1 1 に結合されており、静的な情報と、プロセッサ 1 1 1 2 への命令とを格納する R O M ( 読取り専用メモリ ) 及び / 又はその他の静的記憶装置 1 1 0 6 と、データ記憶装置 1 1 0 7 ( 例 : 磁気ディスク又は光ディスクと、それに対応するディスクドライブ ) とを備えている。データ記憶装置 1 1 0 7 は、バス 1 1 1 1 に結合されており、情報及び命令を格納する。

#### 【 0 1 1 6 】

[00127]更に、コンピュータシステム 1 1 0 0 には、バス 1 1 1 1 に結合されており、  
10  
コンピュータの使用者に情報を表示するディスプレイ装置 1 1 2 1 ( 例 : C R T ( 陰極線管 ) 、液晶ディスプレイ ( L C D ) ) を結合することができる。英数字キー及びその他のキーを有しており、情報及びコマンド選択をプロセッサ 1 1 1 2 に伝える英数字入力装置 1 1 2 2 も、バス 1 1 1 1 に結合することができる。更なるユーザ入力装置は、バス 1 1 1 1 に結合されるカーソル制御装置 1 1 2 3 ( 例 : マウス、トラックボール、トラックパッド、スタイルス、カーソル方向キー ) であり、方向情報及びコマンド選択をプロセッサ 1 1 1 2 に伝え、ディスプレイ 1 1 2 1 上のカーソルの動きを制御する。

#### 【 0 1 1 7 】

[00128]バス 1 1 1 1 に結合することのできる別の装置は、媒体 ( 例 : 紙、フィルム、  
20  
又は類似するタイプの媒体 ) 上に情報を記録するために使用することのできるハードコピー装置 1 1 2 4 である。バス 1 1 1 1 に結合することのできる別の装置は、電話又はハンドヘルド型パームデバイス ( h a n d h e l d p a l m d e v i c e ) と通信するための有線 / 無線通信機能 1 1 2 5 である。

#### 【 0 1 1 8 】

[00129]なお、本発明においては、システム 1 1 0 0 及び関連のハードウェアの一部又は全てを使用することができる。しかしながら、コンピュータシステムの別の構成では、これらデバイスの一部又は全てを備えてもよいことを理解されたい。

#### 【 0 1 1 9 】

[00130]この技術分野における当業者には、上記の説明を読み終えた後、本発明の数多くの変更及び修正が明らかになるであろうが、説明を目的として図示及び記載した実施形態の何れも、本発明を制限することを意図したものではないことを理解されたい。従って、様々な実施形態の細部を示したが、それらは、本発明の本質的な特徴のみを記載している特許請求の範囲を制限することを意図したものではない。  
30

#### 【 図面の簡単な説明 】

#### 【 0 1 2 0 】

【 図 1 】ビデオ符号化器の一実施形態のブロック図である。

【 図 2 】ビデオ符号化プロセスの一実施形態のフローチャートである。

【 図 3 】ノイズフィルタモジュールの一実施形態のブロック図である。

【 図 4 】ノイズフィルタリングプロセスの一実施形態のフローチャートである。

【 図 5 】ノイズフィルタリングプロセスの別の実施形態のフローチャートである。  
40

【 図 6 】ノイズフィルタリングプロセスの更に別の実施形態のフローチャートである。

【 図 7 】推定プロセスの一実施形態のフローチャートである。

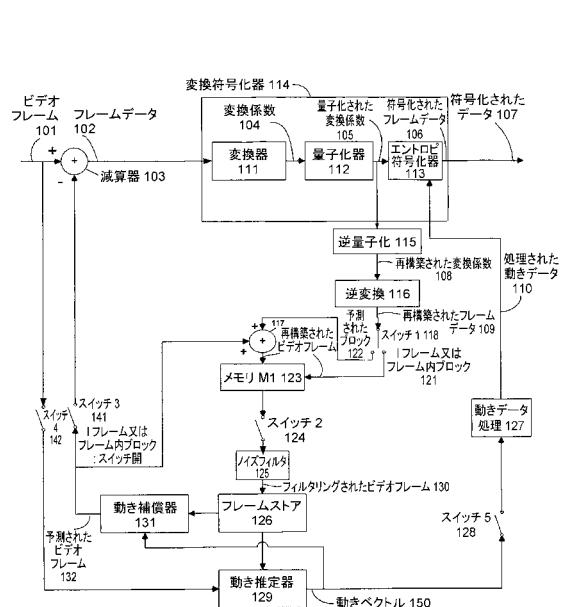
【 図 8 】推定プロセスの別の実施形態のフローチャートである。

【 図 9 】ビデオ復号器の一実施形態のブロック図である。

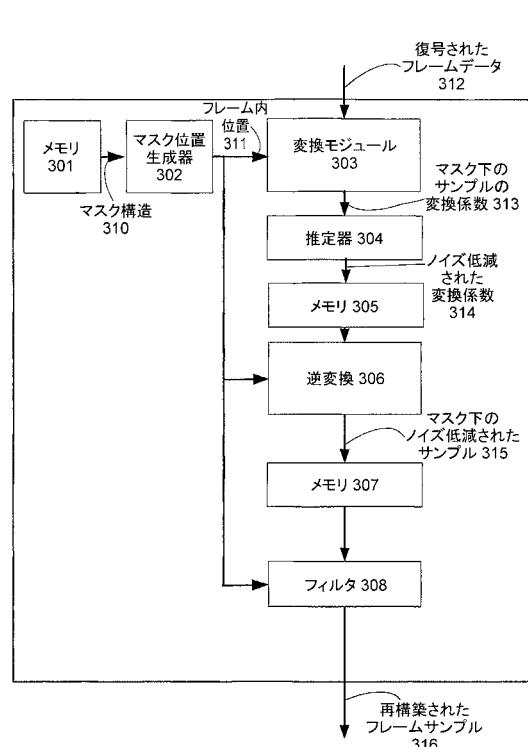
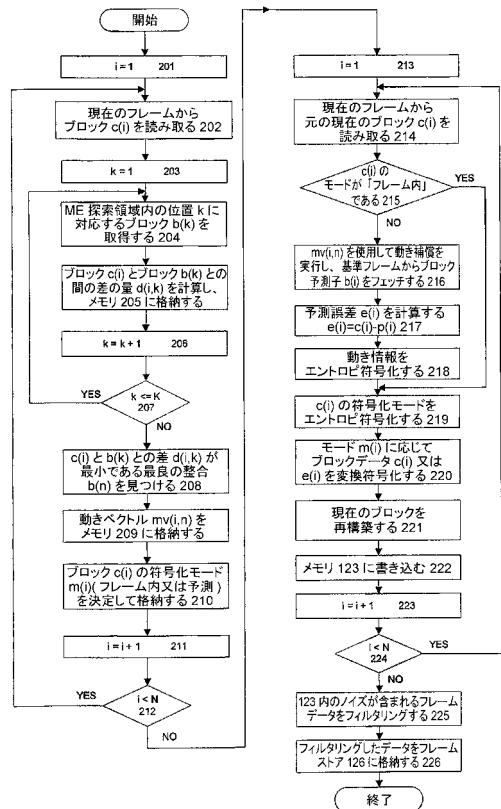
【 図 1 0 】ビデオ復号プロセスの一実施形態のフローチャートである。

【 図 1 1 】コンピュータシステムの一実施形態のブロック図である。

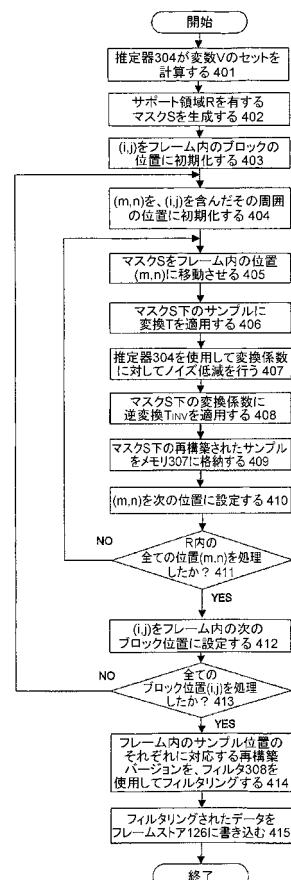
【図1】



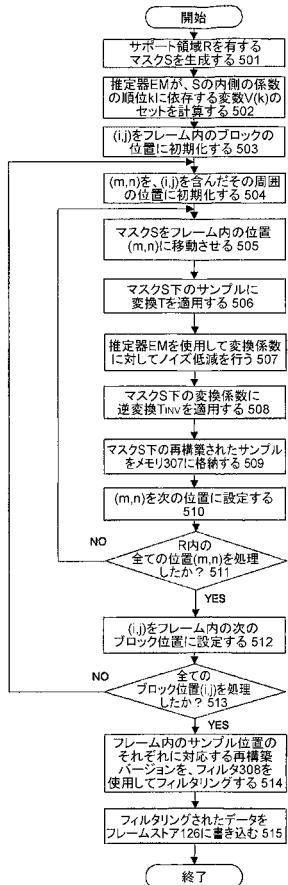
【図2】



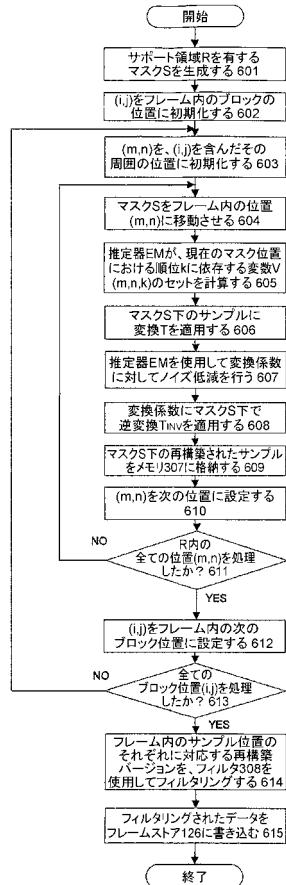
【図4】



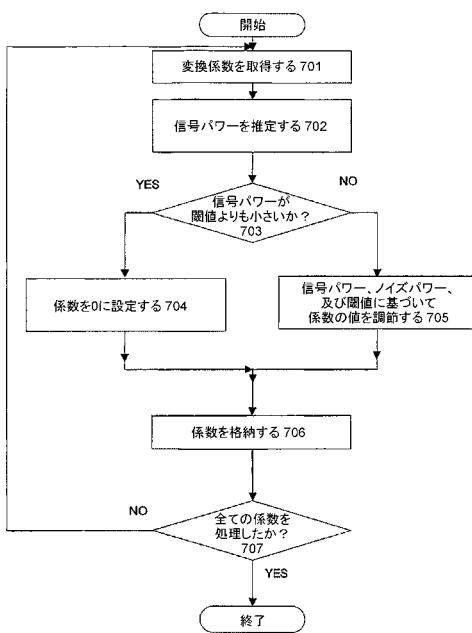
【図5】



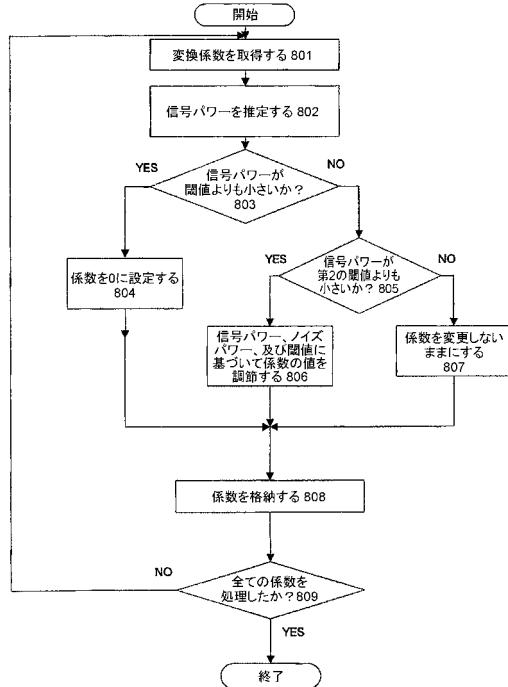
【図6】



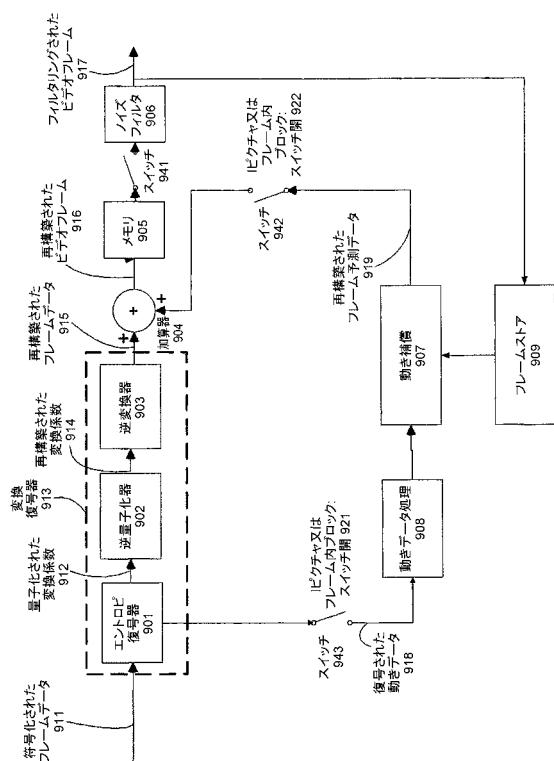
【図7】



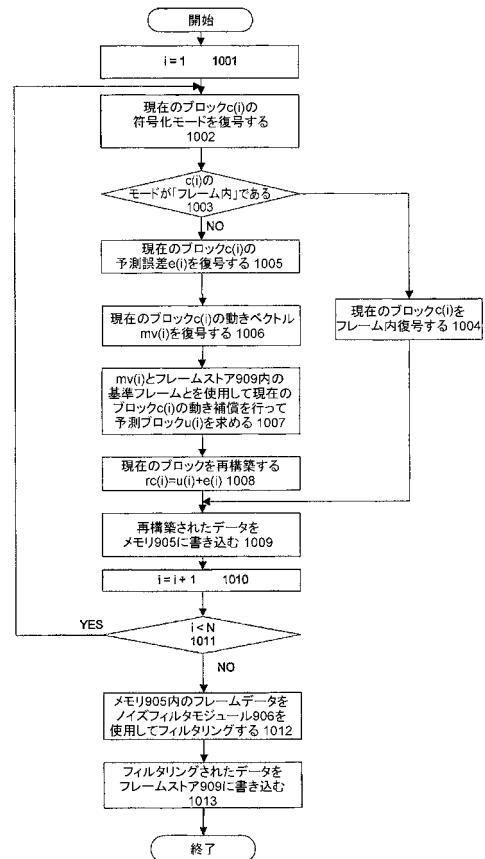
【図8】



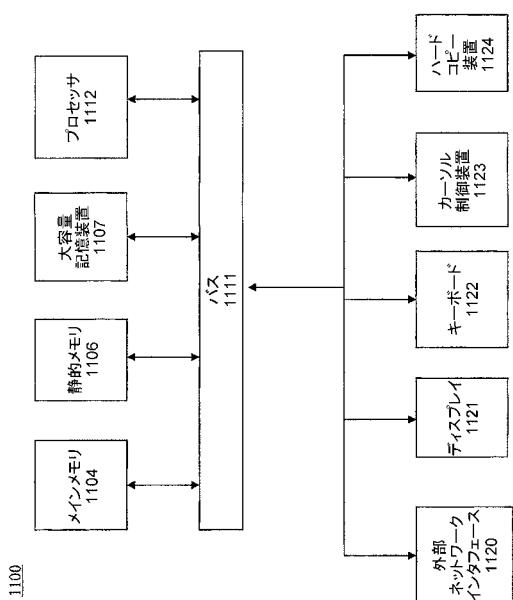
【図 9】



【図 10】



【図 11】



---

フロントページの続き

(72)発明者 ポッセン , フランク

スイス , シーエイチ - 1228 ブラン - イエス クアテス エックスエックス , シーエイチ  
. デス ペティーツ - ボイス 71

審査官 岩井 健二

(56)参考文献 特開2002-216126 (JP, A)

特開2002-218466 (JP, A)

特開平09-212623 (JP, A)

特開平07-212627 (JP, A)

特開平07-015631 (JP, A)

特開平06-078296 (JP, A)

国際公開第2005/039187 (WO, A1)

(58)調査した分野(Int.Cl., DB名)

H04N 7/24 - 7/68