

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4444790号
(P4444790)

(45) 発行日 平成22年3月31日(2010.3.31)

(24) 登録日 平成22年1月22日(2010.1.22)

(51) Int.Cl. F I
HO4N 7/30 (2006.01) HO4N 7/133 Z

請求項の数 8 (全 52 頁)

(21) 出願番号	特願2004-318271 (P2004-318271)	(73) 特許権者	000001007
(22) 出願日	平成16年11月1日(2004.11.1)		キヤノン株式会社
(65) 公開番号	特開2005-143105 (P2005-143105A)		東京都大田区下丸子3丁目30番2号
(43) 公開日	平成17年6月2日(2005.6.2)	(74) 代理人	100076428
審査請求日	平成16年11月1日(2004.11.1)		弁理士 大塚 康德
(31) 優先権主張番号	2003906071	(74) 代理人	100112508
(32) 優先日	平成15年10月31日(2003.10.31)		弁理士 高柳 司郎
(33) 優先権主張国	オーストラリア(AU)	(74) 代理人	100115071
(31) 優先権主張番号	2003906072		弁理士 大塚 康弘
(32) 優先日	平成15年10月31日(2003.10.31)	(74) 代理人	100116894
(33) 優先権主張国	オーストラリア(AU)		弁理士 木村 秀二
(31) 優先権主張番号	2003906073		
(32) 優先日	平成15年10月31日(2003.10.31)		
(33) 優先権主張国	オーストラリア(AU)		

最終頁に続く

(54) 【発明の名称】 画像圧縮方法、画像圧縮装置、画像圧縮システム、プログラム

(57) 【特許請求の範囲】

【請求項1】

画像のシーケンスのうち現在の画像を圧縮する装置が行う画像圧縮方法であって、

(a) 前記装置が有する処理手段が、現在の画像をウェーブレット変換によって変換して、変換係数の集合を提供する工程と、

(b) 前記装置が有する処理手段が、変換係数の集合を複数のブロックに分割する工程と、

(c) 前記装置が有する処理手段が、少なくとも1つのブロックに関して符号化終了条件を判定する工程と、

(d) 前記装置が有する処理手段が、前記少なくとも1つのブロックを一連のコーディングパスで符号化する工程と、

(e) 前記装置が有する処理手段が、現在のコーディングパスにおける圧縮処理の速度値と、現在のコーディングパスを最終コーディングパスとした場合に生じるブロックの劣化を示すひずみ値と、を計算し、当該計算の結果に基づいて後続するコーディングパスに係る速度とひずみに関する性能尺度を予測する工程と、

(f) 前記装置が有する処理手段が、前記予測された性能尺度が前記判定する工程で符号化終了と判定されるための前記符号化終了条件を満たす場合、前記後続するコーディングパスで符号化する前に前記少なくとも1つのブロックの符号化を終了する工程と、

(g) 前記ブロックの劣化が画像間で連続することによって発生するフリッカを減少させるために、前記シーケンスにおいて先行する画像に対するウェーブレット変換係数に

10

20

において係数の切捨てビットプレーンまでの下位 2 ビットの値と、前記現在の画像に対するウェーブレット変換係数において前記下位 2 ビットに対応する 2 ビットの値と、に応じて前記現在の画像に対するウェーブレット変換係数を修正する平滑化処理を行う工程とを備え、

前記平滑化処理を行う工程では、前記先行する画像における前記下位 2 ビットのうちの
下位 1 ビットの値と、前記現在の画像における前記 2 ビットのうちの
下位 1 ビットの値と、
が異なる場合に、前記下位 2 ビットの上位ビットと下位ビット及び前記 2 ビットの上位
ビットと下位ビットのそれぞれが取りうるビット値の組み合わせ毎に設定されている量
に基づいて、前記現在の画像に対するウェーブレット変換係数の切捨てビットプレーンを修
正する

10

ことを特徴とする画像圧縮方法。

【請求項 2】

前記性能尺度は速度 - ひずみ傾きであり、前記予測された速度 - ひずみ傾きが速度 - ひずみカットオフ閾値未満である場合、前記符号化終了条件を満たすことを特徴とする請求項 1 に記載の画像圧縮方法。

【請求項 3】

1 つ以上のブロックを前記シーケンス中の先行する画像の対応するブロックと比較する工程と、

前記比較する工程において前記 1 つ以上のブロックと前記対応する先行するブロックとの間で動きが検出されなかった場合、前記 1 つ以上のブロックを静的ブロックとして分類する工程とを更に含み、

20

前記判定する工程において、前記静的ブロックとして分類されたブロックとその他のブロックとで前記符号化終了条件を変えることを特徴とする請求項 1 又は 2 に記載の画像圧縮方法。

【請求項 4】

前記比較する工程は前記 1 つ以上の前記ブロックに関するモーション検出パラメータを計算することを含むことを特徴とする請求項 3 に記載の画像圧縮方法。

【請求項 5】

前記モーション検出パラメータは、ブロックの 2 乗の和及びそれに対応する先行するブロックの 2 乗の和の積の平方根で除算された前記ブロックとそれに対応する先行するブロックとの内積から成ることを特徴とする請求項 4 に記載の画像圧縮方法。

30

【請求項 6】

画像のシーケンスのうちの現在の画像を圧縮する画像圧縮装置であって、

(a) 現在の画像をウェーブレット変換によって変換して、変換係数の集合を提供する手段と、

(b) 変換係数の集合を複数のブロックに分割する手段と、

(c) 少なくとも 1 つのブロックに関して符号化終了条件を判定する手段と、

(d) 前記少なくとも 1 つのブロックを一連のコーディングパスで符号化する手段と

、
(e) 現在のコーディングパスにおける圧縮処理の速度値と、現在のコーディングパスを最終コーディングパスとした場合に生じるブロックの劣化を示すひずみ値と、を計算し、当該計算の結果に基づいて後続するコーディングパスに係る速度とひずみに関する性能尺度を予測する手段と、

40

(f) 前記予測された性能尺度が前記判定する手段で符号化終了と判定されるための前記符号化終了条件を満たす場合、前記後続するコーディングパスで符号化する前に前記少なくとも 1 つのブロックの符号化を終了する手段と、

(g) 前記ブロックの劣化が画像間で連続することによって発生するフリッカを減少させるために、前記シーケンスにおいて先行する画像に対するウェーブレット変換係数に
おいて係数の切捨てビットプレーンまでの下位 2 ビットの値と、前記現在の画像に対する
ウェーブレット変換係数において前記下位 2 ビットに対応する 2 ビットの値と、に応じて

50

前記現在の画像に対するウェーブレット変換係数を修正する平滑化処理を行う手段とを備え、

前記平滑化処理を行う手段は、前記先行する画像における前記下位 2 ビットのうちの低位 1 ビットの値と、前記現在の画像における前記 2 ビットのうちの低位 1 ビットの値と、が異なる場合に、前記下位 2 ビットの上位ビットと下位ビット及び前記 2 ビットの上位ビットと下位ビットのそれぞれが取りうるビット値の組み合わせ毎に設定されている量に基づいて、前記現在の画像に対するウェーブレット変換係数の切捨てビットプレーンを修正する

ことを特徴とする画像圧縮装置。

【請求項 7】

画像のシーケンスのうちの現在の画像を圧縮する画像圧縮システムであって、シーケンス中の先行する画像の圧縮について記述する情報を格納する記憶装置と、前記記憶装置に接続され、

現在の画像をウェーブレット変換によって変換して、変換係数の集合を提供し、前記変換係数の集合を複数のブロックに分割し、

少なくとも 1 つのブロックに関して、シーケンス中の 1 つ以上の先行する画像に関する格納情報に従属する符号化終了条件を判定し、

前記少なくとも 1 つのブロックを一連のコーディングパスで符号化し、

現在のコーディングパスにおける圧縮処理の速度値と、現在のコーディングパスを最終コーディングパスとした場合に生じるブロックの劣化を示すひずみ値と、を計算し、当該計算の結果に基づいて後続するコーディングパスに係る速度とひずみに関する性能尺度を予測し、

前記予測された性能尺度が前記判定で符号化終了と判定されるための前記符号化終了条件を満たす場合、前記後続するコーディングパスで符号化する前に前記少なくとも 1 つのブロックの符号化を終了し、

前記ブロックの劣化が画像間で連続することによって発生するフリッカを減少させるために、前記シーケンスにおいて先行する画像に対するウェーブレット変換係数において係数の切捨てビットプレーンまでの下位 2 ビットの値と、前記現在の画像に対するウェーブレット変換係数において前記下位 2 ビットに対応する 2 ビットの値と、に応じて、前記現在の画像に対するウェーブレット変換係数を修正する平滑化処理であって、当該平滑化処理では、前記先行する画像における前記下位 2 ビットのうちの低位 1 ビットの値と、前記現在の画像における前記 2 ビットのうちの低位 1 ビットの値と、が異なる場合に、前記下位 2 ビットの上位ビットと下位ビット及び前記 2 ビットの上位ビットと下位ビットのそれぞれが取りうるビット値の組み合わせ毎に設定されている量に基づいて、前記現在の画像に対するウェーブレット変換係数の切捨てビットプレーンを修正し、

前記現在の画像の圧縮について記述する情報を前記記憶装置に格納するプロセッサとを備えることを特徴とする画像圧縮システム。

【請求項 8】

コンピュータに請求項 1 乃至 5 の何れか 1 項に記載の画像圧縮方法を実行させるためのプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明はデジタルビデオ圧縮の分野に関する。特に、本発明は、ビデオフレームのシーケンスを符号化する方法及びビデオ符号化アプリケーションにおける速度制御に関する。本発明は、更に、ビデオフレームのシーケンスを符号化するための装置及びコンピュータプログラムに関する。

【背景技術】

【0002】

デジタル画像及びデジタルビデオは、汎用コンピュータ、デジタルスチルカメラ及びデ

10

20

30

40

50

デジタルビデオカメラを含めた数多くの装置に格納され、使用される。格納、メモリ及び帯域幅のコストを低減するために、デジタル画像及びデジタルビデオは、通常、圧縮形態で装置に格納される。広く使用されている画像圧縮の規格は「Joint Photographic Experts Group」、すなわち、JPEG規格である。画像圧縮に関する発展段階の新たな規格はJPEG 2000規格である。JPEG及びJPEG 2000は共に名目上は静止画像圧縮規格である。JPEGは主に静止画像を圧縮するために使用されるが、大まかに「モーションJPEG」として知られる様々なビデオ符号化フォーマットがある。モーションJPEGはJPEGを使用してビデオの各フレームを静止画像として符号化し、ビデオの全ての符号化フレームをモーションJPEG符号化ストリームにラッピングするための圧縮ビデオストリームフォーマットを提供する。しかし、モーションJPEGは正式に規格化されたわけではない。現在、JPEG 2000規格は、JPEG 2000 Final Committee Draft Version 1.0、2000年3月16日、ISO/IEC FCD 15444 1 (第I部)及びJPEG 2000 Final Committee Draft、2000年12月7日、ISO/IEC FCD 15444 2 (第II部)の2つの刊行物に記載されている静止画像圧縮モードを規定している。また、JPEG 2000規格は、Motion JPEG 2000 Final Committee Draft 1.0、2001年3月14日、ISO/IEC FCD 15444 3 (第III部)に記載された、モーションJPEG 2000として知られるビデオ符号化モードを規定している。モーションJPEG 2000は、ビデオの個々のフレームがJPEG 2000によって(静止画像モードで)符号化され、その結果得られる圧縮フレームが圧縮ビデオストリーム(例えば、ファイルフォーマット)にラッピングされるという点で、モーションJPEGに類似している。潜在的にはJPEG モーショングループ(モーションJPEG又はモーションJPEG 2000)より良好な圧縮を提供しうる、MPEGグループなどの他のビデオ符号化フォーマットも存在しているが、JPEG モーショングループはいくつかの異なる利点を提供する。加えて、モーションJPEG 2000は静止画像モードにおけるJPEG 2000に由来する一連の特徴を提供する。

【0003】

本明細書の説明の便宜上、「JPEG 2000」という用語は、静止画像モードではJPEG 2000規格、すなわち、静止画像(ビデオの1フレーム)を圧縮/伸張するための規格を表し、ビデオ圧縮/伸張モードのJPEG 2000規格に言及する場合には「モーションJPEG 2000」(MJ2K)を表すために使用される。

【0004】

画像又はビデオを圧縮する場合に実行されるべき重要な決定は、圧縮画像及び圧縮ビデオに対してどれほどの数のバイトを利用可能であるかということである。圧縮速度が速くなるほど、格納及び送信に要求されるバイトの数は少なくなる。その妥協点は、圧縮速度が増すにつれて、得られる伸張後の画像又はビデオの画質は一般に低下するという点である。1つの目標は、許容しうる画質レベルを維持しつつ圧縮を最大にするか、あるいは一定のバイト数の中で画像又はビデオの画質を最大にすることである。所定の速度で画質を最大にするためにJPEG 2000により推奨されている技法は、符号化後速度ひずみ最適化(PVRD-opt)と呼ばれる速度制御戦略を使用する方法である。最適化は、実際の符号化自体は完了しているが、利用可能なバイトバジェットを超過したときに採用される。そこで、例えば、実現可能な最高の画質で所望の目標ビットレートに適合させるように圧縮データをトリムバックする(すなわち、圧縮データの一部を放棄する)ためにPCRD optが使用される。JPEG 2000により推奨されるPCRD optを利用する圧縮方式の1つの問題点は、コードブロックが過剰符号化される、すなわち、必要な数より多くのコーディングパスが生成される場合が多いということである。余りに大きな過剰符号化は圧縮方式を遅延させる。更に、過剰符号化は中間圧縮データをバッファするために要求されるメモリの量を増加させる。

【0005】

速度制御に加えて、画質に影響を及ぼす様々な重要な圧縮システムパラメータがある。例えば、JPEG及びJPEG 2000においては、人間の視覚系統に相応する高周波数

10

20

30

40

50

とは対照的に、低い周波数の重要性を強調するために1組のサブバンド又は係数重みが見られる。J P E G 2 0 0 0では、それらは通常は「視覚重み」と呼ばれる。

【0006】

ビデオシーケンスを圧縮すると、単一のフレームを符号化する場合に現れるのとは異なる一連のひずみが導入される。例えば、M J 2 Kビデオが復号され、表示される場合、フリッカと呼ばれる一種の急速に動く雑音としてひずみが現れる可能性がある。多くの場合、このフリッカは各々の伸張フレームでは容易にははっきりと見えず、復号ビデオを表示するときのようにフレームが急速に連続して表示されて初めて明白に現れる。更に、M J 2 Kは各ビデオフレームを独立して符号化するので、P C R D o p tが使用される場合、後続するフレームにおける対応するコードブロックはそれぞれ異なるビットプレーンに切捨てられることが多く、その結果、「コードブロック切捨てフリッカ」と呼ばれる別の形態のフリッカが現れる。フリッカを改善する手段として、人間の様々な視覚重み集合が提案されている。それらの視覚重みは、一般に、人間のコントラスト感度機能(C S F)に従って取り出される。視覚重み付けは、通常、C S Fに従って輝度成分及び色信号成分に対して固定重み付けすることから成る。しかし、フリッカを減少させるためのそのような特定の重みによる改善は十分なものではなく、更なる改善が依然として望ましい。

【発明の開示】

【発明が解決しようとする課題】

【0007】

本発明の目的は、既存の構成の1つ以上の欠点をほぼ克服すること又は少なくとも改善することである。

【課題を解決するための手段】

【0008】

本発明の第1の面によれば、画像のシーケンスのうちの現在の画像を圧縮する装置が行う画像圧縮方法であって、

(a) 前記装置が有する処理手段が、現在の画像をウェーブレット変換によって変換して、変換係数の集合を提供する工程と、

(b) 前記装置が有する処理手段が、変換係数の集合を複数のブロックに分割する工程と、

(c) 前記装置が有する処理手段が、少なくとも1つのブロックに関して符号化終了条件を判定する工程と、

(d) 前記装置が有する処理手段が、前記少なくとも1つのブロックを一連のコーディングパスで符号化する工程と、

(e) 前記装置が有する処理手段が、現在のコーディングパスにおける圧縮処理の速度値と、現在のコーディングパスを最終コーディングパスとした場合に生じるブロックの劣化を示すひずみ値と、を計算し、当該計算の結果に基づいて後続するコーディングパスに係る速度とひずみに関する性能尺度を予測する工程と、

(f) 前記装置が有する処理手段が、前記予測された性能尺度が前記判定する工程で符号化終了と判定されるための前記符号化終了条件を満たす場合、前記後続するコーディングパスで符号化する前に前記少なくとも1つのブロックの符号化を終了する工程と、

(g) 前記ブロックの劣化が画像間で連続することによって発生するフリッカを減少させるために、前記シーケンスにおいて先行する画像に対するウェーブレット変換係数において係数の切捨てビットプレーンまでの下位2ビットの値と、前記現在の画像に対するウェーブレット変換係数において前記下位2ビットに対応する2ビットの値と、に応じて、前記現在の画像に対するウェーブレット変換係数を修正する平滑化処理を行う工程とを備え、

前記平滑化処理を行う工程では、前記先行する画像における前記下位2ビットのうちの下位1ビットの値と、前記現在の画像における前記2ビットのうちの下位1ビットの値と、が異なる場合に、前記下位2ビットの上位ビットと下位ビット及び前記2ビットの上位ビットと下位ビットのそれぞれが取りうるビット値の組み合わせ毎に設定されている量に

10

20

30

40

50

基づいて、前記現在の画像に対するウェーブレット変換係数の切捨てビットプレーンを修正する

ことを特徴とする画像圧縮方法が提供される。

【0012】

本発明の別の面によれば、画像のシーケンスのうちの現在の画像を圧縮する画像圧縮装置であって、

(a) 現在の画像をウェーブレット変換によって変換して、変換係数の集合を提供する手段と、

(b) 変換係数の集合を複数のブロックに分割する手段と、

(c) 少なくとも1つのブロックに関して符号化終了条件を判定する手段と、

(d) 前記少なくとも1つのブロックを一連のコーディングパスで符号化する手段と

、
(e) 現在のコーディングパスにおける圧縮処理の速度値と、現在のコーディングパスを最終コーディングパスとした場合に生じるブロックの劣化を示すひずみ値と、を計算し、当該計算の結果に基づいて後続するコーディングパスに係る速度とひずみに関する性能尺度を予測する手段と、

(f) 前記予測された性能尺度が前記判定する手段で符号化終了と判定されるための前記符号化終了条件を満たす場合、前記後続するコーディングパスで符号化する前に前記少なくとも1つのブロックの符号化を終了する手段と、

(g) 前記ブロックの劣化が画像間で連続することによって発生するフリッカを減少させるために、前記シーケンスにおいて先行する画像に対するウェーブレット変換係数において係数の切捨てビットプレーンまでの下位2ビットの値と、前記現在の画像に対するウェーブレット変換係数において前記下位2ビットに対応する2ビットの値と、に応じて、前記現在の画像に対するウェーブレット変換係数を修正する平滑化処理を行う手段と
を備え、

前記平滑化処理を行う手段は、前記先行する画像における前記下位2ビットのうちの下位1ビットの値と、前記現在の画像における前記2ビットのうちの下位1ビットの値と、が異なる場合に、前記下位2ビットの上位ビットと下位ビット及び前記2ビットの上位ビットと下位ビットのそれぞれが取りうるビット値の組み合わせ毎に設定されている量に基づいて、前記現在の画像に対するウェーブレット変換係数の切捨てビットプレーンを修正する

ことを特徴とする画像圧縮装置が提供される。

【0016】

本発明の別の面によれば、画像のシーケンスのうちの現在の画像を圧縮する画像圧縮システムであって、

シーケンス中の先行する画像の圧縮について記述する情報を格納する記憶装置と、

前記記憶装置に接続され、

現在の画像をウェーブレット変換によって変換して、変換係数の集合を提供し、

前記変換係数の集合を複数のブロックに分割し、

少なくとも1つのブロックに関して、シーケンス中の1つ以上の先行する画像に関する格納情報に従属する符号化終了条件を判定し、

前記少なくとも1つのブロックを一連のコーディングパスで符号化し、

現在のコーディングパスにおける圧縮処理の速度値と、現在のコーディングパスを最終コーディングパスとした場合に生じるブロックの劣化を示すひずみ値と、を計算し、当該計算の結果に基づいて後続するコーディングパスに係る速度とひずみに関する性能尺度を予測し、

前記予測された性能尺度が前記判定で符号化終了と判定されるための前記符号化終了条件を満たす場合、前記後続するコーディングパスで符号化する前に前記少なくとも1つのブロックの符号化を終了し、

前記ブロックの劣化が画像間で連続することによって発生するフリッカを減少させるた

10

20

30

40

50

めに、前記シーケンスにおいて先行する画像に対するウェーブレット変換係数において係数の切捨てビットプレーンまでの下位2ビットの値と、前記現在の画像に対するウェーブレット変換係数において前記下位2ビットに対応する2ビットの値と、に応じて、前記現在の画像に対するウェーブレット変換係数を修正する平滑化処理であって、当該平滑化処理では、前記先行する画像における前記下位2ビットのうちの下位1ビットの値と、前記現在の画像における前記2ビットのうちの下位1ビットの値と、が異なる場合に、前記下位2ビットの上位ビットと下位ビット及び前記2ビットの上位ビットと下位ビットのそれぞれが取りうるビット値の組み合わせ毎に設定されている量に基づいて、前記現在の画像に対するウェーブレット変換係数の切捨てビットプレーンを修正し、

前記現在の画像の圧縮について記述する情報を前記記憶装置に格納するプロセッサとを備えることを特徴とする画像圧縮システムが提供される。

10

【0024】

次に、図面を参照して、従来の技術のいくつかの面及び本発明のいくつかの実施例を説明する。

【発明を実施するための最良の形態】

【0025】

添付の図面のうちの1つ以上の図面において、同じ図中符号を有するステップ及び/又は特徴を参照する場合、それらのステップ及び/又は特徴は、この明細書の便宜上、特に指定のない限り、同じ機能又は動作を有する。

【0026】

20

ここで説明する構成の原理はビデオフレームのシーケンスの符号化に一般に適用されることが可能である。しかし、説明を容易にするため、MJ2Kに関連して構成を説明する。しかし、本発明が説明される構成に限定されることは意図されていない。例えば、本発明はウェーブレット符号化に基づくいくつかのフレーム内ビデオ圧縮方法に適用できるであろう。

【0027】

1.0 JPEG2000の概要

本発明の様々な実現形態を説明する前に、本発明の理解を助けるためにJPEG2000の概要を簡単に説明する。以下の説明は本発明に最も大きく関連するJPEG2000の部分に集中している。更に、本発明の構成を説明する便宜上、ここで使用する用語は、特に指定のない限り、前述のJPEG2000刊行物(第I部、第II部及び第III部)で説明されている用語と同じである。

30

【0028】

JPEG2000画像の符号化の概要を図5に示す。ステップ501で、入力画像500は任意に前処理される。この前処理はレベルシフティング及び成分変換を含む。例えば、入力RGB色空間画像をYCbCr色空間画像に変換することができる。ステップ502では、離散的ウェーブレット変換とは無関係に、画像の各色空間成分が変換される。ステップ503では、ウェーブレット変換係数が整数値に量子化され、複数のコードブロックにタイリングされる。次に、ステップ504では、算術エンコーダによって、各コードブロックがビットプレーン又はその一部分で何らかの最小部分ビットプレーンまでエントロピー符号化される。この符号化が完了した後、速度制御のために、JPEG2000規格により推奨されている符号化後速度-ひずみ最適化アルゴリズム(PCRD-opt)が使用される(ステップ505)。ステップ506では、JPEG2000コードストリーム507を提供するために、符号化コードブロックがヘッダ情報と共にグルーピングされる。

40

【0029】

JPEG2000画像は、これらのステップの各々を逆の順序で実行することにより復号される。

【0030】

次に、図5の符号化プロセスを更に詳細に説明する。

50

【 0 0 3 1 】

1.1 離散的ウェーブレット変換及び多重分解能復号

画像成分の単一レベル離散的ウェーブレット変換 (DWT) が図 1 に示されている。画像成分 1 1 0 は、4 つ合わせて 1 つの単一レベル DWT 画像 1 2 0 を形成する 4 つのサブバンド LL 1 1 3 0、HL 1 1 4 0、LH 1 1 5 0 及び HH 1 1 6 0 に分解される。DWT は画像の多重分解能記述を与える。LL 1 サブバンド 1 3 0 は低い分解能の画像成分を表す。特に、低分解能画像は行ごとに元の画像の公称で $1/2$ の数の画素を含み、列ごとに元の画像の公称で $1/2$ の数の画素を含む大きさで表現されている。

【 0 0 3 2 】

DWT 画像 1 2 0 (LL 1、HL 1、LH 1 及び HH 1 サブバンド) に対して単一レベル逆 DWT を実行して、元の画像 1 1 0 を得ることができる。従って、DWT 画像 1 2 0 は画像を暗黙に 2 つの分解能、すなわち、分解能 1 と呼ばれる元の画像の分解能と、分解能 0 と呼ばれる LL 1 サブバンドの分解能で表現する。

10

【 0 0 3 3 】

LL 1 サブバンド 1 3 0 に対して別の単一レベル DWT を実行することができる。LL 1 サブバンドを 4 つのサブバンド LL 2 1 7 0、HL 2 1 7 2、LH 2 1 7 4 及び HH 2 1 7 6 に解析すると、その結果、サブバンド LL 2、HL 2、LH 2、HH 2、HL 1、LH 1 及び HH 1 を含む 2 レベル DWT 画像 1 8 0 が得られる。LL 2 サブバンドは元の画像を更に低い分解能で表現 (また、LL 1 サブバンドを同様により低い分解能で表現) する。特に、画像は行ごとに元の画像の公称で $1/4$ の数の画素を含み、列ごとに元の画像の公称で $1/4$ の数の画素を含む大きさで表現されている。サブバンド HL 1、LH 1 及び HH 1 はレベル 1 サブバンドと呼ばれる。サブバンド LL 2、HL 2、LH 2 及び HH 2 はレベル 2 サブバンドと呼ばれる。便宜上、元の画像は LL 0 サブバンドと呼ばれる。

20

【 0 0 3 4 】

更に後続する LL サブバンドに対して単一レベル DWT を更に適用することにより、より高いレベルの DWT 画像、及び画像成分の更に低い分解能の表現が得られる。1 つの S レベル DWT は LL S サブバンドと、サブバンド H L S、L H S、H H S、H L (S - 1)、L H (S - 1)、H H (S - 1)、...、H L 1、L H 1 及び H H 1 とを含む。2 文字から成るサブバンドの名称の後の数は DWT レベルを表す。従って、H L (S - 1) はレベル S - 1 における H L サブバンドを表す。S レベル DWT は暗黙に (S + 1) の分解能で画像を表現する。それらの分解能は分解能 0、分解能 1、...、分解能 S と呼ばれる。分解能 0 は LL S サブバンドを表し、分解能 1 は L L (S - 1) サブバンドを表し、それ以降も同様である。従って、分解能 S は元の画像を現す。

30

【 0 0 3 5 】

J P E G 2 0 0 0 は各画像成分に対して DWT を使用する。各サブバンドは独立して符号化される。各サブバンドは圧縮画像コードストリームから独立して復号されることが可能である。例えば、単一レベル DWT 画像の場合、LL 1 サブバンド (すなわち、画像の分解能 0) はその他のレベル 1 サブバンド HL 1、LH 1 及び HH 1 を復号せずに符号化され、表示されることが可能である。より高いレベル DWT 画像の場合には、DWT 解析の各段階で様々な LL サブバンドを再構成することにより、画像の他の分解能バージョンを同様に復号し、表示することができる。例えば、3 レベル DWT 画像では、LL 3、HL 3、LH 3 及び HH 3 の各サブバンドから LL 2 サブバンドを再構成することにより、分解能 1 で画像を表示することができる。

40

【 0 0 3 6 】

1.2 コードブロック、コーディングパス及びレイヤ

J P E G 2 0 0 0 における変換画像の各サブバンドはコードブロックと呼ばれる複数のブロックにタイリングされる。図 2 は、HL 1 サブバンド 1 4 0 の 16 のブロックへのタイリングを示し、そのうちの 1 つのブロック 2 1 0 が陰影により指示されている。各コードブロックはコードブロック係数の整数表現を与えるために量子化され、次に、ほぼ独立

50

して(エントロピー)符号化される。各コードブロックは最上位ビットプレーンから最下位ビットプレーンの順に符号化される。公称で、ビットプレーンごとに3つのコーディングパスが存在する。

【0037】

図3は、そのようなコードブロックの符号化を示す。コードブロック210は、場合によっては単純にコードブロック320と呼ばれる圧縮コードブロックへと符号化される。コードブロック320はコーディングパスコードセグメント330のシーケンスから構成されている。従って、コードブロックはコーディングパスコードセグメント(又は単にコーディングパス)の順序付きシーケンスである。

【0038】

1つのコードブロックは複数のブロックレイヤに分割されることが可能である。説明を容易にするため、ここで使用される用語「ブロックレイヤ」は1つのコードブロックの、対応する画質レイヤに寄与するコードブロックを表す。各画質レイヤは画質を連続的に改善し、従って、デコーダは各レイヤに含まれるコードブロック寄与を順次復号することが可能であろう。従って、画質レイヤはあらゆる分解能における対応するブロックレイヤの集合体であるといえることができる。図3は、3つのレイヤ、すなわち、ブロックレイヤ0、340、ブロックレイヤ1、350及びブロックレイヤ2、360に分割されたコードブロックの一例を示す。各ブロックレイヤは負ではない数のコーディングパスを含む。第1のレイヤは第1の数のコーディングパスを含み、第2のレイヤは次の数のコーディングパスを含み、それ以降も同様である。コードブロックのブロックレイヤが空であっても良いことに注意することは重要である。図3の例はブロックレイヤごとに3つのコーディングパスがある場合を示す。

【0039】

デコーダは、エントロピー符号化を元に戻すことにより、ブロックコードからコードブロックを再構成する。第1の数のコーディングパスのみを復号することにより、コードブロックの低画質表現が得られる。例えば、図3において、レイヤ0のみからコードブロックを再構成すること、又はレイヤ0及び1からコードブロックを再構成することが可能であろう。従って、いずれかのコーディングパス境界でブロックコードを切捨てると、その結果、コードブロックの低画質表現が得られる。このようなブロックの切捨ては量子化の1つの形態である。ビットプレーンを符号化することにより、コードブロックの埋め込み形態の量子化が実行される。コードブロックに含まれる余分のビットプレーンごとに、有効量子化ステップサイズは1/2に減少する。

【0040】

画像の1つのレイヤは、対応するブロックレイヤの集合体から構成される。すなわち、レイヤ0はDWT画像におけるコードブロックごとのブロックレイヤ0から構成され、レイヤ1はコードブロックごとのブロックレイヤ1から構成され、それ以降も同様である。ブロックレイヤ内にコーディングパスが含まれないという意味で、ブロックレイヤは空であっても良い。レイヤ0のみを復号(また、適切な逆DWT、成分変換などを実行)することにより、画像を復号できる。レイヤ0を復号することは、コードブロックごとにブロックレイヤ0を復号することを意味している。この場合、各コードブロック、従って、画像は圧縮画像コードストリームで表現されるより低い画質で復号される。

【0041】

1.3 速度 ひずみ最適化

先に説明したように、各コードブロックはいくつかのコーディングパスで符号化される。言うまでもなく、各コーディングパスの符号化は与えられるバイトバジェット全体から取り出されるビットを消費することができる。所定のコーディングパスの速度は、そのコーディングパス及びそれ以前の全てのコーディングパスにより消費されたバイトの数である。

【0042】

「ひずみ」は、元の画像(又はブロック)と比較した場合の圧縮画像(又はブロック)

10

20

30

40

50

と関連する画質劣化である。コードブロックのひずみは、現在のコーディングパスが最終コーディングパスである場合にどれほど多くのビデオ劣化が導入されるかを反映している。一般に、コーディングパスが増すにつれて、ひずみは減少する（すなわち、それと同等に画質は向上する）。多くの情報が符号化されるほど、デコーダから見てコードブロックの精度は向上する。

【 0 0 4 3 】

従って、各コーディングパスはそれと関連する速度尺度及びひずみ尺度を有し、それらの尺度は一体となって速度 ひずみポイントを定義する。各コードブロックは、コーディングパスごとに対（速度、ひずみ）を構成する 1 組の速度 ひずみポイントを有する。1 組の速度 - ひずみポイントは圧縮アルゴリズムの効率を特徴付けている。

10

【 0 0 4 4 】

コードブロック n が i 番目のコーディングパスで終了した場合、各コーディングパスと関連する速度を $r^0, r^1, r^2, \dots, r^i$ とし、ひずみを $d^0, d^1, d^2, \dots, d^i$ とする。合計速度は次の式により表され、

$$R_{total} = \sum_{j=0}^i r^j$$

(1)

全ひずみの尺度は次の式により表される。

$$D_{total} = \sum_{j=0}^i d^j$$

20

(2)

尚、速度 ひずみポイント (r^0, d^0) は、コードストリームにコーディングパスが全く含まれていないポイントを表すことに注意する。「切捨てポイント」は最終コードストリームに含まれる最終コーディングパスである。切捨てポイントについては図 8 に関して更に説明する。

【 0 0 4 5 】

先に述べたように、1 つの目標は所定の速度におけるひずみを最小にすることである。すなわち、

30

$$R_{total} \leq R_{desired}$$

となるような

$$\min_{i_n} D_{total}$$

(3)

を求めることである。式中、 R_{total} は実際に実現された合計速度であり、コードブロックの異なる切捨てポイントに対して最小値を求める。この最適化問題はラグランジュの乗算器の方法を使用して解決されることが可能である。すなわち、

40

$$\{D^{i_n} + \lambda^{i_n}\}$$

(4)

が制約 $R_{total} \leq R_{desired}$ の下で最小になるような i_n (可能であれば) 及びコードブロック切捨てポイントの集合 i_n を求めるのである。また、このコードブロック切捨てポイントの集合 i_n は制約を受ける問題 (3) を解く。実際には、どのような i_n によっても厳密な制約には適合しないと考えられるため、 $R_{total} \sim R_{desired}$ が使用される (通常は $R_{total} < R_{desired}$)。

【 0 0 4 6 】

50

以下、図6を参照して、項(4)によって問題(3)を解く手続きを説明する。この手続きは図5のステップ505に相当する。手続きはステップ605で始まり、ブロックごとに各速度-ひずみポイントに対応する傾きを計算するステップ610へ進む。 i_n コーディングパスを有するブロックnの場合、速度-ひずみ傾きの集合 $\lambda^0, \lambda^1, \lambda^2, \dots, \lambda^{i_n}$ は次の式により表される。

$$\lambda^j = \begin{cases} d^{j-1} - d^j & j = 0 \\ \frac{d^{j-1} - d^j}{r^j - r^{j-1}} & j = 1, \dots, i_n \\ r^j - r^{j-1} & j = i_n + 1 \end{cases}$$

(5)

これらの傾きは減少してゆくと想定される。すなわち、コードブロックnごとに $\lambda^0, \lambda^1, \lambda^2, \dots, \lambda^{i_n}$ となる。 $j < j+1$ であれば、コードブロックnに関して可能な速度-ひずみポイントの集合から速度-ひずみポイント (r^j, d^j) が除去される。残る速度-ひずみポイントはその後再度ラベル付けされ、傾きが再度計算される。このプロセスは、傾きがすべて減少するまで継続される。このプロセスの終了時に、 $k = i_n$ とした場合にk個の速度-ひずみポイントが存在すると仮定すると、 $\lambda^0, \lambda^1, \lambda^2, \dots, \lambda^k$ となる。この傾きの集合をブロックnの速度-ひずみ傾きの集合という。あるいは、この傾きの集合はブロックnの凸閉包に依存しているという。

【0047】

次のステップ620では、初期傾き λ^0 が選択され、 λ_{low} 及び λ_{high} はそれぞれ0とに設定される。 $\lambda^0 = 10$ の傾きが初期傾きとして選択されるのが好ましい。ステップ630では、ブロックnごとの最適終了ポイント i_n が判定され、最適関連合計速度 $R(\lambda)$ が計算される。これらの終了ポイントは(4)におけるラグランジュの最小化問題の解である。ステップ630については以下に更に詳細に説明する。

【0048】

ステップ630が完了した後、 $R(\lambda) < R_{desired}$ であるか否かを判定するために、決定ブロック640で検査が実行される。決定ブロック640が「ノー」を戻した場合、処理はステップ650へ続き、 λ_{low} が λ^0 に設定される。その後、処理はステップ670で再開する。決定ブロック640が「イエス」を戻した場合には、処理はステップ660へ続き、 λ_{high} が λ^0 に設定される。その後、処理はステップ670で再開する。

【0049】

決定ブロック670では、 $R(\lambda) < R_{desired}$ であるか否か及び $R(\lambda) > R_{desired}$ であるか否かを判定するための検査が実行される。尚、 ϵ は1未満の何らかの速度許容差である。 $\epsilon = 0.99$ が使用されるのが好ましい。図6には示されていないが、繰り返しカウンタを保持すべきであり、このカウンタを超えると、決定ブロック670は「イエス」を戻す。この繰り返しの目的は、手続きが無限ループに入るのを停止することである。決定ブロック670が「イエス」を戻した場合、処理はステップ685へ続く。ステップ685では、ブロックごとの最適切捨てポイント i_n が出力される。その後、処理はステップ690で終了する。決定ブロック670が「ノー」を戻した場合には、処理はステップ680へ続く。ステップ680では、現在の傾き λ^0 が更新され、処理はステップ630に戻る。

【0050】

ステップ630において最適合計速度及び関連する切捨てポイントは次のようにして計算される。ブロックnに関して、 λ の動作傾きに対する最適切捨てポイントは i_n であり

$$\lambda^{i_n+1} \leq \lambda \leq \lambda^{i_n}$$

(6)

となっている。尚、最適切捨てポイント i_n は λ の関数である。従って、最適切捨てポイ

10

20

30

40

50

ントと関連する速度 r^{i_n} は の関数であり、合計最適速度は次の式により表される。

$$R(\lambda) = \sum_n r^{i_n}(\lambda)$$

(7)

(6) を満足させる 2 つ以上の i_n が存在する場合、可能な各々の i_n のレコードが保持される。それに対応して、異なる可能な合計最適速度の集合 $\{R(\)\}$ が存在し、この有限集合の中に最小値 $R_{\min}(\)$ 及び最大値 $R_{\max}(\)$ がある。決定ブロック 670 において $R_{\min}(\) < R_{\text{desired}}$ であり且つ $R_{\max}(\) > R_{\text{desired}}$ であれば、決定ブロック 670 は「イエス」を戻し、ステップ 685 で、 R_{desired} 以下である最大の $R(\)$ に対応する切捨てポイント i_n の集合が出力される。式(4)を介して式(3)を解くためのこの手続きの数多くの変形が存在し、それらを上記の手続きの代わりに使用することは可能である。

10

【0051】

切捨てポイント i_n と関連して、速度 - ひずみ傾き i_n (又は i_{n+1}) があり、切捨てポイント i_n におけるコードブロック n の速度 - ひずみ傾きは傾き i_n である。

【0052】

この最適化を使用して、所定の速度に対して画像を「最適に」符号化するために、コードブロックごとに保持すべきコーディングパスの数を判定することができる。それらのコーディングパスは、画像を符号化するための単一のレイヤを形成するためにコードブロックごとに符号化されることが可能である。また、最適化技法は 1 つの画像を符号化するためのいくつかのレイヤを構成するためにも使用できる。例えば、一方が画素ごとに 1 ビットの総合速度に対して最適なコーディングパス切捨てポイントの集合であり、他方は画素ごとに 2 ビットの総合速度に対して最適のコーディングパス切捨てポイントの集合であるような 2 つの切捨てポイントの集合を判定することができる。所定のコードブロックに対して、第 2 の集合に対応するコーディングパスは第 1 の集合に対応するコーディングパスと同じであるか、それより後のコーディングパスである。そこで、第 1 のレイヤは、第 1 の集合に従って判定された全てのコーディングパスを含めることにより構成される。第 2 のレイヤは、第 1 のレイヤには含まれていなかった、第 2 の集合に従った全てのコーディングパスを含めることにより形成される。

20

30

【0053】

一般に、この最適化技法はここでは PCRD - opt ルーチンと呼ばれる。PCRD - opt は圧縮後速度 - ひずみ最適化の 1 つの形態である。この技法は、一般に、データが符号化された後に切り捨てポイントの最適集合を判定することを含む。この技法を使用することの欠点は、コードブロックの過剰符号化にある。例えば、いくつかのビットプレーンがその速度 - ひずみ傾きを判定するために符号化されるが、それらは PCRD - opt が提供された後に初めて放棄される。

【0054】

PCRD - opt ルーチンを適用した後のコーディングパス終了ポイントにおけるコードブロック切捨ては量子化の 1 つの形態であることに注意すべきである。終了ポイントの後に符号化されるブロックのビットプレーンは最終ビットストリームには含まれない。一般に、PCRD - opt 量子化は固定精度 2 進フォーマットで DWT 係数を表現するとき起こるいかなる量子化より重要であり、(切捨てプロセスにおいてビットプレーンが放棄されない場合を除いて) 通常は JPEG 2000 で使用される明示量子化より重要である。

40

【0055】

速度 - ひずみ最適化の実際の有効度の重要な一面は、適切なひずみ関数の選択である。各コードブロックが属するサブバンドに従って、コードブロックごとに平均二乗誤差ひずみに重み付けするために、視覚重みを使用することができる。このように、より大きな重

50

みを有するサブバンドからのコードブロックは、一般に、小さな重みを有するサブバンドからのコードブロックより高い画質に（すなわち、より多くの部分ビットプレーンによって）符号化される。

【 0 0 5 6 】

1 . 4 コーディングパス

各ビットプレーンの中で、コーディングは図 4 に示すようにストライプ指向走査に沿っている。各ストライプ 4 3 0 は 4 つのビット記号行を含み、各々のストライプの中で、ビット記号は経路 4 2 0 に従って列ごとに、左から右へ走査される。ビットプレーン符号化は、（一般に）0 でない要素を含む最上位ビットプレーンから始まり、最下位ビットプレーンに向かって進む。各ビットプレーンは、通常、以下の 3 つのコーディングパスで符号化される。

10

【 0 0 5 7 】

1 有意性伝播パス - 有意性状態はまだ無意に設定されているが、有意状態を有する 8 つの最近隣要素のうち少なくとも 1 つを有する係数を符号化する。これは、有意性伝播パスが有意になりそうである係数を含むことを意味している。

【 0 0 5 8 】

2 マグニチュード精密化パス - 以前のビットプレーンで有意になった係数を符号化する。これにより、ウェーブレット係数のマグニチュードは更に精密化される。

【 0 0 5 9 】

3 クリーンアップパス - 所定のビットプレーンにおける最後のパスであり、先行するコーディングパスで符号化されなかった係数を含む。いくつかの係数が有意ではなく、有意の近隣要素を含まない場合、それらはランレングス符号化され、そうでない場合には、以前のコーディングパスの場合と同様にして符号化される。

20

通常、ビットプレーンは上記のコーディングパスの順序で符号化される。

【 0 0 6 0 】

1 . 5 コードストリーム構成

各分解能レベルにおけるいくつかのコードブロックはプレシントの中で回収され、1 つのプレシントの中のブロックレイヤの集合は 1 つのパケットへと配列される。「プレシント」は、特定の DWT レベルの各サブバンドにおける対応する領域を記述するために使用される用語である。単純なケースでは、プレシントは 1 つの分解能レベルのサブバンド全体（例えば、HL 1 サブバンド、LH 1 サブバンド及び HH 1 サブバンド）である。パケットは J P E G 2 0 0 0 圧縮コードストリームに対する基本ビルディングブロックであると考えことができ、1 つのパケットは特定のプレシント、ブロックレイヤ及び分解能レベルからの全ての圧縮画像データを含む。

30

【 0 0 6 1 】

J P E G 2 0 0 0 ビットストリームはパケットのシーケンスを含む。各パケットは画像タイルの単一の分解能、（画質）レイヤ、成分及び空間領域に関する符号化画像情報を含む。J P E G 2 0 0 0 コードストリームはパケット及びヘッダ情報ブロックのシーケンスにより構成されている（詳細については、J 2 K 規格を参照）。

【 0 0 6 2 】

2 . 0 モーション J P E G 2 0 0 0

モーション J P E G 2 0 0 0 は、他のビデオフレームから独立した個々のビデオフレームの J P E G 2 0 0 0 圧縮を含むビデオ圧縮システムである。ウェーブレット利用エンコーダとして、ビデオフレームの非無損失 J P E G 2 0 0 0 圧縮は伸張出力にウェーブレットアーティファクトを導入する。所定の 1 つのビデオフレームに対して、視覚的に無損失であると考えられる非無損失圧縮のレベルが存在する。そのような視覚的に無損失の圧縮速度では、符号化プロセスにより導入されるウェーブレットアーティファクトのマグニチュードは小さく、1 つの画像を見ている観測者がアーティファクトに気づくことはない。

40

【 0 0 6 3 】

しかし、各フレームが J P E G 2 0 0 0 エンコーダを使用して独立して符号化されたビ

50

デオストリームにおいては、フレーム間のわずかな変化であっても、エンコーダが非線形の性質を有するために連続する復号フレームにきわめて大きく異なる複数組のウェーブレットアーティファクトを生じる可能性がある。1つの復号ビデオストリームの中で、それらのウェーブレットアーティファクトは、その基礎にある映像コンテンツの変化に単純に関連付けられないように急速に変化する。多くの場合、それらの変化は目に見えており、ビデオの個々のフレームを視覚的には許容できると考えられる場合であっても、変化を視覚的に許容できないこともある。それらの急速に変化するアーティファクト、すなわち、フリッカアーティファクトの問題は、アーティファクトがビットレートの低下に伴って大きくなり、より明白になるために、圧縮レベルが高くなるほどはるかに大きくなる。

【 0 0 6 4 】

いくつかの用途では、ビデオを固定ビットレートで、例えば、毎秒 2 4 M b i t で符号化することが望ましい。ビデオにおける固定ビットレートを実現する自明の方法の1つは、各フレームを同じ量だけ圧縮する方法である。例えば、毎秒 3 0 フレームの場合、毎秒 2 4 M b i t の制約を満たすためには、各フレームは約 8 2 0 K b i t の目標ビットレートに圧縮されなければならない。J P E G 2 0 0 0 により推奨されている速度制御アルゴリズムは P C R D - o p t であり、これは、各ビデオフレームの圧縮データが所望の目標速度に適合することを保証するために圧縮後に適用される。この方法を使用することの利点は、どのような速度が与えられても、アルゴリズムは所望の速度に到達するために各フレームにおける各コードブロックの実際の速度値及びひずみ値を使用するため、その速度に対して最適のビデオ画質を実現できることである。しかし、P C R D - o p t を使用する

10

20

【 0 0 6 5 】

2 . 1 フリッカアーティファクト

2 . 1 . 1 コードブロック切捨てポイントフリッカ

復号された M J 2 K ビデオにおけるフリッカアーティファクトは、M J 2 K 符号化ビデオを視覚的に無損失に符号化できる最小ビットレートを制限し、それより低いどのビットレートでも符号化ビデオの画質を低下させる。M J 2 K では、速度 ひずみ最適化アルゴリズムは、通常、各フレームに独立して適用される。これにより、コードブロックごとに圧縮ビットストリームに含まれるコーディングパスの数がフレーム間で、その基礎にあるビデオフト数の総変化に関連付けられないように変化するため、フリッカが発生するであろう。すなわち、所定のコードブロックの切捨てポイントは異なるビットプレーン間で頻繁に変化し、その結果、後続するフレームの間でいくつかのウェーブレットアーティファクトが出現したり、消滅したりすることになる。例えば、いくつかの実験は、ビデオに多くのモーションが存在しない場合であっても、コードブロックのうちの 2 5 % の切捨てポイントがフレームごとに変化することを示唆している。

30

【 0 0 6 6 】

各フレームに独立して速度 ひずみ最適化を適用する通常の方法は、コードブロックが符号化されるビット深さの変化のために、得られる復号ビデオストリームが連続するフレームの中で出現し、消滅するウェーブレットアーティファクトを含むという点で不都合である。表 1 から表 4 はこの点を示している。

40

表 1 : フレーム 1 の 1 つのブロックの一部に対する D W T 係数

-2. 4 2 6 2 2 0	-1 1. 1 6 2 4 3 8	-0. 5 3 9 8 8 7
3. 6 9 7 4 8 4	1 6. 7 9 6 4 0 6	5. 7 9 8 2 5 1
-3. 6 3 9 8 4 1	-1 5. 3 6 4 2 4 9	-7. 3 0 1 1 4 3

表 2 : 1 6 の量子化ステップサイズを有する表 1 の係数に対する量子化 D W T 係数

50

0	0	0
0	1	0
0	0	0

表3：フレーム2の1つのブロックの一部に対するDWT係数

-2. 184122	-11. 267410	0. 155337
3. 005134	16. 275234	4. 651533
-2. 982268	-5. 198070	-7. 582343

10

表4：8の量子化ステップサイズを有する表3の係数に対する量子化DWT係数

0	-1	0
0	2	0
0	0	0

ビデオシーケンスの一例の第1の符号化フレームからのDWT係数の可能な3×3サブセットを表1に示す。16の量子化ステップサイズを有する対応する量子化値を表2に示す。次のフレームでは、視覚的コンテンツは非常に類似しており、DWT係数の同じサブセットを表3に示す。それらは8のステップサイズで量子化され、その結果を表4に示す。それらの(有効)量子化ステップサイズは、JPEG2000における明示量子化又は(PCRD-optプロセスの)ビットプレーン切捨てを介する暗黙量子化の結果であると考えられるであろう。しかし、この例を説明する便宜上、それらの量子化ステップサイズはビットプレーン切捨ての結果であるものとする。類似するフレームにおける同じ対応するコードブロックが2倍異なる量子化閾値を有することは稀ではない。従って、表2と表4の値の相違からわかるように、典型的なデコーダはフレームごとの復号係数を著しく異なる値に逆量子化するであろう。これは、コードブロック間のDWT係数の差が極めて小さい場合であっても、(0と0でない値との間の遷移により発生する)ウェーブレットアーティファクトの出現と消滅をもたらし、また、(係数の大きさの変化により発生する)ウェーブレットアーティファクトの見かけの大きさと強さの変化をもたらす。

20

30

【0067】

2.1.2 ウェーブレット係数フリッカ

ウェーブレット係数フリッカは、J2K符号化を使用するビデオストリームの小さな変化の非線形増幅が原因となって起こる。モーション及びシーンの変化による視覚的に重要な変化を保持しつつ、雑音に起因するウェーブレット係数のわずかな変化を除去することが望ましい。

【0068】

2.2 実現形態

上述のビデオフレームのシーケンスを符号化する方法、並びに以下に説明する第1及び第2の構成は、方法の機能又は部分機能を実行する1つ以上の集積回路などの専用ハードウェアで実現されるのが好ましい。そのような専用ハードウェアはグラフィックプロセッサ、デジタルシグナルプロセッサ、又は1つ以上のマイクロプロセッサ及び関連メモリを含むであろう。

40

【0069】

ビデオフレームのシーケンスを符号化する方法は、図27に示すような、汎用コンピュータシステム2700を使用して実施されても良く、このシステムにおいては、図5～図7、図10～図21、図25及び図26のプロセスがコンピュータシステム2700内部で実行されるアプリケーションプログラムのようなソフトウェアとして実現される。特に、図5～図7、図10～図21、図25又は図26の方法のステップは、コンピュータに

50

より実行されるソフトウェアの命令により実現される。命令は、各々が1つ以上の特定のタスクを実行する1つ以上のコードモジュールとして形成される。ソフトウェアは2つの別個の部分に分割されていても良く、その第1の部分は符号化方法を実行し、第2の部分は第1の部分とユーザとの間のユーザインタフェースを管理する。ソフトウェアは、例えば、以下に説明する記憶装置を含めたコンピュータ読み取り可能な媒体に格納されていても良い。ソフトウェアはコンピュータ読み取り可能な媒体からコンピュータにロードされ、その後、コンピュータにより実行される。そのようなソフトウェア又はコンピュータプログラムが記録されているコンピュータ読み取り可能な媒体は、コンピュータプログラム製品である。コンピュータにおけるコンピュータプログラム製品の使用は、ビデオフレームのシーケンスを符号化するための有利な装置を実現するのが好ましい。

10

【0070】

コンピュータシステム2700はコンピュータモジュール2701と、キーボード2702及びマウス2703などの入力装置と、プリンタ2715、表示装置2714及びスピーカ(図示せず)を含む出力装置とにより形成されている。変調器-復調器(モデム)トランシーバ装置2716はコンピュータモジュール2701により、例えば、電話回線2721又は他の機能媒体を介して接続可能である通信ネットワーク2720との間で通信を実行するために使用される。モデム2716はインターネット、及びローカルエリアネットワーク(LAN)又はワイドエリアネットワーク(WAN)などの他のネットワークシステムに対するアクセスを獲得するために使用されることが可能であり、実現形態によってはコンピュータモジュール2701に組み込まれていることもある。

20

【0071】

コンピュータモジュール2701は、通常、少なくとも1つのプロセッサユニット2705と、例えば、半導体ランダムアクセスメモリ(RAM)及び読み取り専用メモリ(ROM)から形成されるメモリユニット2706とを含む。モジュール2701は、ビデオ表示装置2714及びスピーカに結合するオーディオビデオインタフェース2707と、キーボード2702及びマウス2703、更にオプションであるジョイスティック(図示せず)に対するI/Oインタフェース2713と、モデム2716及びプリンタ2715に対するインタフェース2708とを有するいくつかの入出力(I/O)インタフェースを更に含む。実現形態によっては、モデム2716はコンピュータモジュール2701の内部、例えば、インタフェース2708の内部に組み込まれていても良い。記憶装置2709が設けられており、通常、これはハードディスクドライブ2710及びフロッピー(登録商標)ディスクドライブ2711を含む。磁気テープドライブ(図示せず)が更に設けられていても良い。CD-ROMドライブ2712は通常は不揮発性データ源として設けられる。コンピュータモジュール2701の構成要素2705~2713は、通常、相互接続バス2704を介して、当業者に知られている従来通りのコンピュータシステム2700の動作モードをもたらしように通信する。個々で説明する構成を実施できるコンピュータの例はIBMTM-PC及びその互換機、SUNTM Sparcstations又はそこから派生した類似のコンピュータシステムを含む。

30

【0072】

通常、アプリケーションプログラムはハードディスクドライブ2710に常駐しており、その実行時にプロセッサ2705により読み取られ、制御される。プログラム及びネットワーク2720から取り出されるデータの間格納は、おそらくはハードディスクドライブ2710と協調して、半導体メモリ2706を使用して実現される。場合によっては、アプリケーションプログラムはCD-ROM又はフロッピー(登録商標)ディスクに符号化された状態でユーザに供給され、対応するドライブ2712又は2711を介して読み取られるか、あるいはユーザによりネットワーク2720からモデム装置2716を介して読み取られても良い。更に、ソフトウェアは他のコンピュータ読み取り可能な媒体からコンピュータシステム2700にロードされることも可能である。ここで使用される用語「コンピュータ読み取り可能な媒体」は、実行及び/又は処理のためにコンピュータシステム2700に命令及び/又はデータを提供することに関与するあらゆる記憶媒体又は

40

50

伝送媒体を表す。記憶媒体の例は、コンピュータモジュール 2701 の中にある装置であるか又は外にある装置であるかに関わらず、フロッピー（登録商標）ディスク、磁気テープ、CD-ROM、ハードディスクドライブ、ROM又は集積回路、光磁気ディスク、又はPCMCIAカードなどのコンピュータ読み取り可能なカードを含む。伝送媒体の例は無線送信チャンネル又は赤外線送信チャンネル、並びに別のコンピュータ又はネットワーク接続された装置へのネットワーク接続、Eメール送信及びウェブサイトなどに記録された情報を含めたインターネット又はイントラネットを含む。

【0073】

システム 2701 で実行されるソフトウェアにより生成される圧縮ビデオデータはコンピュータネットワーク 2720 を介して送信されても良い。あるいは、圧縮データは、後の使用に備えて、例えば、ハードディスクドライブ 2710 又は CD-ROM 2712 に格納されても良い。

10

【0074】

3.0 第1の構成

図7は、ビデオシーケンスを処理するための第1の構成の概要を示す。図7の方法のいくつかのステップは、図5の方法を使用して画像を処理するときを使用されるステップと同じである。しかし、図7では、それらのステップはループ701の中に挿入されており、ループ701を繰り返すたびに、ビデオシーケンスの1つのフレームが符号化される。

【0075】

方法はステップ700で始まり、ビデオシーケンスの各フレームを順次処理するためにループ701に入る。ビデオシーケンスの全てのフレームが処理されると、ステップ709でループ701から出て、方法はステップ710で終了する。

20

【0076】

ループ701の中では、ステップ501で、プロセッサ2705において実行されるプログラムが現在の入力フレームに何らかの必要な前処理を適用する。先に1つの画像の符号化に関連して説明したように、そのような前処理はレベルシフティング又は成分変換を含むであろう。前処理が完了した後、プロセッサ2705はステップ502で現在の画像にウェーブレット変換を適用する。次に、ステップ503で、プロセッサ2705はウェーブレット変換係数を整数値に量子化する。量子化されたウェーブレット係数にフリッカ減少ステップ711が適用されるのが好ましい。しかし、フリッカ減少711が使用されない場合には、プロセス流れは直接にエントロピー符号化ステップ705へ続く。フリッカ減少ステップ711については、以下に図24～図26を参照して章5.0で説明する。

30

【0077】

エントロピー符号化ステップ705では、プロセッサ2705で実行されるプログラムはプロセスウェーブレット係数を複数のコードブロックとして配列し、各コードブロックは算術エンコーダによってビットプレーンで符号化される。各コードブロックの符号化は、過剰符号化の量を減少させるために、符号化方式714により管理される。符号化方式714については、図15及び図16を参照して章3.6で更に詳細に説明する。

【0078】

ステップ705で実行される符号化は、コードブロックが属する成分を考慮に入れる成分重み付け方式713によっても影響を受ける。1つの実現形態では、成分はYCbCr色空間成分のうちの1つである。Y成分は輝度成分であり、Cb成分及びCr成分は色信号成分である。成分重み付け方式713については、図13及び図14を参照して章3.5で更に詳細に説明する。

40

【0079】

更に、ステップ705でコードブロックの符号化を終了させるための決定は、静的ブロック切捨て安定化方式712により影響を受ける。静的ブロックのそのような切捨てについては、図10～図12を参照して章3.4で更に詳細に説明する。

【0080】

50

現在のビデオフレームのコードブロックについてエントロピー符号化705が完了すると、プロセッサ2705はステップ706で圧縮データを目標ビット数にトリミングする。トリミングステップ706はコードブロックの、カットオフ閾値以下である速度ひずみ傾きを有するコーディングパスを放棄する。カットオフ閾値を選択する方法については、以下の章3.3で説明する。

【0081】

符号化後トリミングステップ706に変わる方法として、JPEG2000規格により推奨されているPCRD-optルーチンがエントロピー符号化データに適用されても良い。PCRD-optアルゴリズムは図7にステップ505として示されており、それが代替ステップであることを示すために破線の輪郭により示されている。

10

【0082】

次に、ステップ506で、プロセッサ2705は符号化データをビットストリームに編成し、圧縮フレームを出力する。ビットストリームはネットワーク2720を介して送信されても良いし、あるいは後の使用に備えて、例えば、CD-ROM2712に格納されても良い。

【0083】

次に、ステップ709で、方法は現在のフレームがビデオシーケンスの最終フレームであるか否かを検査する。シーケンス中にまだフレームがあれば(ステップ709のNoオプション)、プロセス流れはステップ501に戻り、シーケンス中の次のフレームを処理する。現在のフレームが最終フレームである場合には(709のYesオプション)、ループ701は終了し、処理はステップ710で終了する。その後、圧縮ビデオシーケンスはコンピュータネットワーク2720を介して送信されても良いし、あるいは後の使用に備えて格納されても良い。圧縮ビデオデータはシステム2700のようなシステムを使用して復号され、たとえば、ビデオ表示装置2714に表示されても良い。

20

【0084】

先に説明した通り、第1の構成は符号化方法714、成分重み付け方法713及び静的ブロック切捨て安定化方法712を使用する。しかし、ビデオ圧縮のための別の構成においては、これら3つの方法の異なる組み合わせを使用することができる。例えば、3つの方法の各々を他の2つを使用せずにビデオ圧縮のために使用することが可能である。

【0085】

3.1 用語

ビデオシーケンスに関して、ビデオがフレーム1, . . . , Fから構成されていると仮定する。実現されるフレーム $f \in \{1, \dots, F\}$ のビットレートは $R_{achieved}$ であり、目標ビットレートは R_{target} である。フレーム f の速度-ひずみ傾き閾値を τ_f とする。速度-ひずみ傾きが閾値 τ_f より小さければ、コードブロックのビットプレーン符号化は終了される。単一のビデオフレームの場合、N個のコードブロックがあり、コードブロック $n \in \{1, \dots, N\}$ に対して、コーディングパスは i_n である。そこで、コードブロック n に対して、そのコーディングパス i_n の速度-ひずみ傾きを τ_n とする。

30

【0086】

以下に示す表記法は、ウェーブレット係数の量子化を記述するために導入される。ウェーブレット係数は $(-1, 1)$ 内に入るように正規化され、各々の係数はP桁の精度をもって固定小数点2進表現を使用して記述される。すなわち、係数Cは次の式により表される。

40

$$C = s \sum_{p=1}^P a_p 2^{-p}$$

(8)

式中、 $s = \pm 1$ は係数の符号であり、 $a_p \in \{0, 1\}$ はCの小数部のp番目の2進数字を表す。尚、この表現は、実数値ウェーブレット係数の有限精度による2進小数への明示量子化を示唆していることに注意する。従って、量子化閾値を論じる場合、それはこの

50

明示量子化を参照するのではなく、速度 - ひずみ最適化ルーチンの後に、ある1つのビットプレーンにおける各コードブロックの切捨てにより起こる暗黙量子化を参照している。

【0087】

ビットプレーンLにおけるビットプレーン切捨てにより係数Cを量子化する演算は、次の式により与えられる演算子Q [C , L]として記述されても良い。

$$Q[C,L] = s \sum_{p=1}^L a_p 2^{-p}$$

(9)

すなわち、Q [C , L]は、CのうちのはじめのL個の2進数字のみが保持されているCの量子化値である。| C |は、sが1に設定されている場合のQ [C , L]と等価であるウェーブレット係数のマグニチュードを表す。コードブロックの切捨てビットプレーンL及び切捨てポイントTは次の式により関連付けられる。

$$L = 2 + \text{floor}[(T + 2)/3]$$

(10)

式中、floorはどの実数もそれより小さく、最も近い整数により近似されるような演算子である。以下の説明中、fは現在のフレームを表し、(f - 1)は先行するフレームを表す。

【0088】

コーディングパス i_n から構成されるコードブロックn 8 0 1のビットプレーン符号化を示す図8を参照して示すように、「切捨てポイント」と「終了ポイント」とは明確に異なっている。尚、コードブロックnの各ビットプレーンは3つの公称コーディングパス(有意性パス、マグニチュード精密化パス及びクリーンアップパス)のうちの1つで符号化される。

【0089】

コードブロックnの符号化はコーディングパス $i_n = 1 \quad 8 \quad 1 \quad 1$ で始まり、コーディングパス $i_n = 2 \quad 8 \quad 1 \quad 2$ へ進み、それ以降、最下位ビットプレーンを符号化する最終コーディングパス8 1 3に至るまで同様に進む。各コーディングパスの終了時に、符号化プロセスを終了させるべきか否かを判定するために決定が実行される。この決定は、以下に更に詳細に説明するいくつかの要因によって影響を受ける。図8の例では、コードブロックnの符号化はコーディングパス $i_n = 9$ の終了時に終了する。これをコードブロックnの終了ポイント8 0 3という。最終圧縮コードストリームはコードブロックnの切捨てポイント8 0 2までに生成された圧縮データを含む。第1の構成の符号化方式7 1 4においては、コードブロックnの切捨てポイントは、通常、コードブロックnの終了ポイントよりコーディングパス1つ分少ない。目標は、過剰符号化の量を最小にすることである。符号化プロセスを終了させるための決定は、実質的に、先行するフレームにおける対応するブロック(すなわち、ウェーブレットドメインの同じ場所)の切捨てポイントに基づいてい

【0090】

3.2 フレーム間の情報の格納

第1のフレームを処理する場合を除いて、図7の方法は現在のフレームを符号化するとき先行するフレームからの情報を利用する。フレーム間の情報を格納するのに適するメモリ構成が図9に示されている。ビデオシーケンスはフレーム1 9 0 1、フレーム2 9 0 2、フレーム3 9 0 3からフレームF 9 0 5までから構成されている。

【0091】

フレーム1 9 0 1の符号化が完了すると、関連情報は、図27に示されるメモリ2 7 0 6の1つの領域であるのが好ましいメモリ9 0 6に格納される。格納された情報は、フ

10

20

30

40

50

フレーム 2 9 0 2 を符号化するときにはプロセッサ 2 7 0 5 により使用される。そこで、フレーム 2 の関連情報はメモリ 9 0 6 の格納情報と置き換えられ、フレーム 3 を処理するときには使用される。このようなメモリ 9 0 6 の順次更新は、最終フレーム 9 0 5 が処理されるまで続く。

【 0 0 9 2 】

ビデオフレーム f が処理された後、そのフレームのウェーブレット係数値はシーケンスの次のフレームを処理するためにメモリの記憶場所 9 1 5 に格納される。1 つの実現形態では、ウェーブレット係数値の全てが格納される。しかし、ビデオフレームのウェーブレット係数値のサブセットのみが格納されるのが好ましい。例えば、フリッカ減少方式 7 1 1 は 2 ビットしか必要としないため、章 5 . 0 で更に説明するように、切り捨てられた係数のうちの 2 つの最下位ビットのみが格納されれば良い。

10

【 0 0 9 3 】

ビデオフレーム f を処理した後、フレーム f の各コードブロックの切捨てポイントはメモリ 9 0 6 の記憶場所 9 1 1 に格納される。切捨てポイント情報は、次のフレームを順次処理する際に符号化を終了させるべきか否かを考慮するときには使用される。1 つの実現形態では、記憶場所 9 1 1 は各コードブロックの切捨てビットプレーン L_n を格納する。この実現形態は、例えば、ビデオフレームの符号化中に P C R D - o p t ルーチンが使用される場合に使用できるであろう。

【 0 0 9 4 】

記憶場所 9 1 1 に各コードブロック n の切捨てビットプレーン L_n を格納するもう 1 つの方法は、コードブロックごとに最終ビットストリームに含まれるコーディングパスの数を格納する。コーディングパスは 0 でないビットプレーンのみを符号化するために採用される。コーディングパスを要求する 0 でないビットプレーンに到達する前に、1 つ以上の 0 であるビットプレーンに遭遇する可能性はある。コードブロック n の 0 であるビットプレーンの数を A_n とする。

20

【 0 0 9 5 】

第 1 の構成において、記憶場所 9 1 1 に格納される値は次の式により表される。

$$\text{value stored} = \left\lfloor \frac{I_n}{3} \right\rfloor + A_n$$

30

(1 1)

式中、 I_n はコードブロック n のビットストリームに含まれるコーディングパスの数である。尚、各ビットプレーンと関連するコーディングパスは通常は 3 つ存在していることに注意する。コードブロックの 0 であるビットプレーンの数はフレームごとに異なるため、各コードブロックにおける 0 であるビットプレーンを補償することにより、次のフレームにおける切捨てポイントの等価の位置を厳密に推定することができる。

【 0 0 9 6 】

1 つのビデオフレームにおけるコードブロックごとの絶対切捨てビットプレーンは、切捨てポイントが判定された後に格納される。切捨てポイントの記憶場所 9 1 1 への格納は、図 7 のステップ 5 0 6 の間に起こるのが好ましい。

40

【 0 0 9 7 】

記憶場所 9 1 3 は、フレームの各成分について実現された速度を格納する。図 9 は、1 つの実現形態においては、画像の Y 成分、Cb 成分及び Cr 成分である 3 つの成分を示す。尚、ビデオフレームについて実現される総合速度は成分ごとの実現速度の和であることに注意する。すなわち、

$$R_{\text{achieved}}^f = \sum_{m=1}^M R_{\text{achieved}}^{f_m}$$

50

(1 2)

式中、Mは成分の数である。

【 0 0 9 8 】

フレームfのカットオフ閾値 λ_f は記憶場所 9 1 2 に格納される。目標速度 R_{target} は記憶場所 9 1 4 に格納される。

【 0 0 9 9 】

記憶場所 9 1 6 はコードブロックnごとの速度 - ひずみ傾きを格納するために使用される。

【 0 1 0 0 】

3 . 3 カットオフ閾値の推定

10

画像を符号化するとき適切な速度制御を実行するためには、適切なカットオフ傾き閾値を選択することが重要である。ビデオシーケンスで現在の画像に関して λ_f を推定することは、先行するビデオフレームのカットオフ速度 λ_{f-1} 及び先行するフレームの実現速度 $R_{achieved}^{f-1}$ を使用することを含む。言い換えれば、フレームfの λ_f は、 λ_{f-1} 、 $R_{achieved}^{f-1}$ 及び目標ビットレート R_{target} の関数である。

【 0 1 0 1 】

第1の構成においては、 λ_1 、すなわち、ビデオシーケンスにおける第1のフレームの傾き閾値はPCRD - o p tアルゴリズムにより選択されるのが好ましいが、他の選択方法が使用されても良い。

【 0 1 0 2 】

20

後続するフレームに関しては、f = 1 の場合の λ_f は次のように推定される。

$$\text{error} = \frac{R_{achieved}^{f-1} - R_{target}}{R_{target}} \times 100$$

(1 3)

及び

$$\lambda_f = \lambda_{f-1} + 10 \times \text{error}$$

(1 4)

30

従って、現在のフレームに関して、プロセッサ 2 7 0 5 は、メモリ 9 0 6 に格納されている先行するフレームからの値を使用して傾きカットオフ閾値を計算する。

【 0 1 0 3 】

第1の構成においては、 λ_f は符号化ステップに先立って現在のフレームに関して計算される。

【 0 1 0 4 】

3 . 4 静的コードブロック切捨て安定化

第1の構成は静的ブロック切捨て安定化方式 7 1 2 を含む。図 1 0 は、ビデオシーケンスの各フレームをMJ2Kフォーマットに圧縮する方法を示す。図 1 0 は図 7 のステップの多くを含むが、静的ブロック切捨て安定化 7 1 2 を伴う符号化ステップ 7 0 5 に関して更に詳細に示す。特に、ステップ 1 0 0 3 から 1 0 0 6 は図 7 のステップ 7 0 5 及び 7 1 2 に対応している。

40

【 0 1 0 5 】

方法はステップ 7 0 0 で始まり、次に、ループ 7 0 1 に入る、ループが繰り返されるたびに、ビデオシーケンスの現在のフレームが圧縮される。全てのフレームが圧縮されると、ループ 7 0 1 は終了し、方法はステップ 7 1 0 で終了する。

【 0 1 0 6 】

ループ 7 0 1 の中では、プロセッサ 2 7 0 5 はステップ 1 0 0 2 で現在のビデオフレームfに対してウェーブレット変換ステップを実行するが、それらのステップは前処理ステップ 5 0 1、ウェーブレット変換ステップ 5 0 2、量子化ステップ 5 0 3 及び、好ましく

50

はフリッカ減少ステップ 7 1 1 を含む。

【 0 1 0 7 】

次に、プロセッサ 2 7 0 5 で実行されるプログラムは、 f の各コードブロック n_f を符号化するためのループ 1 0 0 3 に入る。コードブロックごとに、ステップ 1 0 0 4 でモーション検出パラメータが計算される。このステップについては、図 1 1 を参照して以下に更に詳細に説明する。ステップ 1 0 0 4 で、プロセッサ 2 7 0 5 は現在のコードブロック n_f を先行するフレーム ($f - 1$) の対応するコードブロック n_{f-1} と比較する。コードブロック n_f とコードブロック n_{f-1} との間でモーションが検出されなければ、コードブロック n_f は「静的」であるとフラグ付けされる。そうでない場合には、コードブロック n_f は「非静的」コードブロックであるとフラグ付けされる。

10

【 0 1 0 8 】

モーション検出パラメータが計算されると、ステップ 1 0 0 5 でフレーム f が符号化される。この符号化は符号化手続きを終了させる方法を使用する。ステップ 1 0 0 5 では、現在のコードブロックが静的であるか又は非静的であるかに応じて、様々に異なる終了戦略が適用される。ステップ 1 0 0 5 については、以下に図 1 2 を参照して更に詳細に説明する。

【 0 1 0 9 】

次に、ステップ 1 0 0 6 で、プロセッサは、フレーム f に関して処理すべきコードブロックがまだ存在しているか否かを判定するために試験を実行する。試験 1 0 0 6 が肯定の応答を戻した場合、方法はステップ 1 0 0 4 に戻って、フレーム f の次のコードブロックを処理する。しかし、試験 1 0 0 6 が否定の応答を戻した場合（すなわち、現在のフレームにそれ以上のコードブロックが存在しない場合）には、プロセス流れはステップ 1 0 0 7 へ進む。

20

【 0 1 1 0 】

ステップ 1 0 0 7 で、プロセッサ 2 7 0 5 は各コードブロック n_f [$n = 1, \dots, N$] の切捨てポイントを判定し、格納する。尚、 N は f におけるコードブロックの総数である。各コードブロック n_f の切捨てポイントは、 i_n に対応する最大のポイント i_n を選択することにより判定される（すなわち、図 7 のステップ 7 0 6）。別の実現形態は、J P E G 2 0 0 0 規格により推奨されている P C R D - o p t ルーチンを使用することを含む（すなわち、図 7 のステップ 5 0 5）。他の速度制御戦略が使用されても良い。

30

【 0 1 1 1 】

次に、符号化値はビットストリームとして編成される（ステップ 5 0 6）。

【 0 1 1 2 】

ステップ 7 0 9 で、現在のフレーム f がビデオシーケンスの最終フレームであるか否かを判定するための試験が実行される。試験 7 0 9 が否定の応答を戻した場合、処理はステップ 1 0 0 2 に戻り、シーケンス中の次のフレームを処理する。試験 7 0 9 が肯定の応答を戻した場合（すなわち、それ以上のフレームが存在しない場合）には、ループ 7 0 1 から出て、方法はステップ 7 1 0 で終了する。

40

【 0 1 1 3 】

3 . 4 . 1 コードブロック間のモーションの計算

モーション検出パラメータを計算するステップ 1 0 0 4 について、図 1 1 を参照して更に詳細に説明する。パラメータ計算はステップ 1 1 0 0 で始まり、ステップ 1 1 0 1 へ進み、モーション検出を実行するために使用される全ての必要なパラメータが 0 に初期設定される。

【 0 1 1 4 】

初期設定されるパラメータは、コードブロックが静的ブロックであるか又は非静的ブロックであることを示すために使用される StaticBlock フラグと、Md（モーション検出パラメータ）である。Md の計算で使用される 3 つのパラメータ、すなわち、CorrelCurrPrev、Co

50

rrelCurr及びCorrelPrevも0に初期設定される。

【0115】

第1の構成では、Mdは次のように計算されるのが好ましい相関係数である。

$$Md = \frac{\sum_{j=1}^L (x_j \times y_j)}{\sqrt{(\sum_{j=1}^L x_j^2)(\sum_{j=1}^L y_j^2)}}$$

(15)

式中、 x_1, x_2, \dots, x_L 及び y_1, y_2, \dots, y_L は、共にエントリ数Lを有する2つのベクトルX及びYに対するエントリである。Xはコードブロック n_f のベクトル化表現であり、Yはコードブロック n_{f-1} に対応するベクトル化表現である。Lはコードブロック n_f のサイズである。例えば、コードブロック n_f が高さ×幅の大きさである場合、 $L = \text{高さ} \times \text{幅}$ である。

【0116】

変数の初期設定の後、ループ1102に入り、コードブロック n_f の各ウェーブレット係数を順次処理する。現在考慮されている係数は C^n である。係数ごとに、ステップ1103で、現在のフレーム $f = 1$ であるか否かを判定するために試験が実行される。 $f = 1$ であれば(ステップ1103のYesオプション)、先行するフレームとの比較を実行することができず、ウェーブレット係数に対する修正なしにコードブロック n_f は符号化される。ステップ1102のYesオプションの場合、プロセス流れは直接にステップ1106へ進む。ステップ1103が否定の応答を戻した場合(すなわち、現在のフレームが第1のフレームではない場合)には、コードブロック n_f のあらゆるウェーブレット係数に対して、ステップ1104でメモリからコードブロック n_{f-1} の対応するウェーブレット係数が検索される。先行するフレームの係数は記憶場所915に格納される。

【0117】

ステップ1105で、変数CorrelCurrPrev、CorrelCurr及びCorrelPrevは次のように更新される。

$$\text{CorrelCurr} += C^n * C^n$$

$$\text{CorrelPrev} += C^n_{f-1} * C^n_{f-1}$$

$$\text{CorrelCurrPrev} += C^n * C^n_{f-1}$$

演算子 $+=$ は変数の以前の値に対する累積加算を表す。

現在のコードブロックの全ての係数 C^n が処理され終わると、以下の合計が得られることが理解されるであろう。

【0118】

1. Xがコードブロック n_f である場合、

$$\text{CorrelCurr} = \sum_{j=1}^L x_j^2$$

2. Yがコードブロック n_{f-1} である場合、

$$\text{CorrelPrev} = \sum_{j=1}^L y_j^2$$

3. Xがコードブロック n_f であり且つYがコードブロック n_{f-1} である場合、

10

20

30

40

$$\text{CorrelCurr Prev} = \sum_{j=1}^L (x_j \times y_j)$$

ステップ 1 1 0 6 で、シーケンス中の次のフレームを処理するときを使用するために、現在のコードブロック n_f のウェーブレット係数がメモリ 9 1 5 に格納される。ステップ 1 1 0 7 で、コードブロック n_f の全てのウェーブレット係数が処理され終わったか否かを判定するために試験が実行される。試験 9 0 7 が肯定の応答を戻した場合（すなわち、ブロック中にそれ以上の係数が存在しない場合）、ループ 1 1 0 2 を出て、処理流れはステップ 1 1 0 8 へ進む。現在のコードブロックにまだ係数がある場合には（ステップ 1 1 0 7 の No オプション）、プロセス流れはステップ 1 1 0 3 に戻る。

10

【 0 1 1 9 】

ステップ 1 1 0 8 では、式 (1 5) を使用して Md が計算される。ステップ 1 1 0 9 で、 Md が閾値 T_s より大きいかが否かを判定するために試験が実行される。 Md が閾値より大きい場合（ステップ 1 1 0 9 の Yes オプション）、ステップ 1 1 1 0 で、StaticBlock フラグは、現在のコードブロックが先行するコードブロックと極めて大きく関連していることを示す 1 に設定される。 Md が閾値より小さい場合には（ステップ 1 1 0 9 の No オプション）、StaticBlock は 0 で変わらないままである。プロセス流れはステップ 1 0 0 5 で戻る。

【 0 1 2 0 】

第 1 の構成では、閾値 T_s は 0 . 9 に設定される。他の値も可能である。1 つの実現形態においては、異なる DWT レベルに対して異なる T_s の値が使用される。別の実現形態では、画像の異なる成分に対して異なる T_s の値が使用される。 T_s の値が低くなるほど、安定化プロセスの効率は高くなる。しかし、画質との間に妥協点がある。

20

【 0 1 2 1 】

フレーム間で係数ごとに 2 ビットしか格納されない場合、相関係数 Md はそれら 2 つの利用可能なビットのみを使用して計算される。このようにして計算される相関成分でも、フレーム間のモーションの変化を示す。

【 0 1 2 2 】

3 . 4 . 2 ブロックを符号化する際のモーション検出パラメータの使用

図 1 2 は、ステップ 1 0 0 4 で計算されたモーション検出パラメータが現在のコードブロック n_f の符号化を終了させるべきか否かを判定するときどのように使用されるかを示す。図 1 2 のステップはステップ 1 0 0 5 に対応している。

30

【 0 1 2 3 】

ステップ 1 2 0 1 で、現在の符号化パス T_f は 0 に設定される。次に、ステップ 1 2 0 3 で、先行するビデオフレームの対応するコードブロック n_{f-1} の切捨てポイント $T(f-1)$ がメモリ 9 1 1 から検索される。

【 0 1 2 4 】

次に、ステップ 1 2 0 5 で、通常の JPEG 2 0 0 0 符号化方法を使用してコードブロック n_f が符号化される。次に、ステップ 1 2 0 7 で、現在のブロックの StaticBlock フラグが 1 に設定されているか否かを判定するために試験が実行される。現在のブロックが静的ブロックであれば（ステップ 1 2 0 7 の Yes オプション）、プロセス流れはステップ 1 2 0 9 へ進む。しかし、現在のブロックが静的ブロックではない場合には（ステップ 1 2 0 7 の No オプション）、プロセス流れはステップ 1 2 1 1 へ進む。

40

【 0 1 2 5 】

ステップ 1 2 0 9 で、プロセッサ 2 7 0 5 は現在のコーディングパス $T(f)$ の値を $T(f-1)$ の値と比較する。 $T_f = T(f-1)$ であり且つ現在のコードブロックが静的コードブロックである場合（ステップ 1 2 0 9 の Yes オプション）、現在のコードブロックの符号化はステップ 1 2 1 2 で終了される。

【 0 1 2 6 】

しかし、 $T(f)$ がまだ $T(f-1)$ と等しくない場合には（ステップ 1 2 0 9 の No

50

オプション)、プロセス流れはステップ1211へ進む。

【0127】

現在のブロックが静的ブロックではない場合又は T_f がまだ $T(f-1)$ と等しくない場合には、ステップ1211が実行される。ステップ1211では、現在のブロックについてまだコーディングパスが存在するか否かを判定するために試験が実行される。まだコーディングパスがあれば(ステップ1211のYesオプション)、プロセス流れはステップ1205に戻る。しかし、現在のブロックについてそれ以上コーディングパスが存在しない場合には(ステップ1211のNoオプション)、現在のブロックの符号化はステップ1212で終了する。

【0128】

ステップ1212で現在のブロックの符号化が完了すると、処理はステップ1006で再開する。

【0129】

3.5 フレーム従属成分重み付け

第1の構成は、各ビデオフレームの画像成分に適応重み付けする方法を提供する。このフレーム従属成分重み付け方法713は、YCbCr色空間において輝度成分と色信号成分に対して異なるカットオフ速度-ひずみ閾値 τ_f を使用する。以下の説明中、成分0は輝度成分Yを表し、成分1及び2は、それぞれ、色信号成分Cb及びCrを表す。輝度成分は色信号成分より大きい影響を認知画質に対して及ぼし、従って、重み付け方法713は色信号成分を犠牲にしても輝度成分の符号化を優先する。

【0130】

異なる成分に対する重み付け τ_f は、視覚重み付けを適用する方法に対する改善である。図13及び図14をそれぞれ参照して、2つの二者択一的な重み付け方法を説明する。これら2つの方法のいずれにおいても、現在のフレームの圧縮度は先行するフレームに関して判定される。第1のバージョンでは、圧縮度は色信号成分に基づいて推定される。フレームがより多く圧縮可能である(すなわち、色信号成分を符号化するために要求されるビットの数がより少ない)と判定された場合、輝度成分に割り当てられるビットバジェットは増加する。

【0131】

重み付け方法713の第2のバージョンでは、輝度成分の符号化はある1つのひずみレベルに維持され、色信号成分の符号化は、輝度成分の過剰符号化がある場合にそれを補償するために重み付けされる。

【0132】

更に、方法713の第1のバージョンは、ビデオフレームの圧縮データを色信号成分に限定されるようにトリミングさせることができる。従って、フレームの輝度成分の圧縮が余りにも寛容であった場合には、そのフレームの色信号成分からより多くのデータをトリミングして、ビットを節約することになる。

【0133】

3.5.1 輝度カットオフ閾値の重み付け

図13は、輝度カットオフ閾値が潜在的に調整される成分重み付け方式713を示す。図7を参照すると、図13は、成分重み付け方式713を利用する符号化ステップ705に関して更に詳細に示している。明瞭にするため、図13は図7のステップの一部を再現している。例えば、図13のステップ1300、1301、1312及び1313は図7のステップ700、701、709及び710に対応している。成分重み付け方式がビデオシーケンスの各フレームに順次適用されることを強調するために、それらのステップは図13でも保たれている。

【0134】

フレーム従属成分重み付けの方法713はステップ1300で始まり、次に、プロセッサ2705がビデオシーケンスの各フレームを順次処理するループ1301に入る。ループ1301では、プロセッサ2705は第1のパスでビデオシーケンスの第1のフレーム

10

20

30

40

50

を処理し、後続するパスの間にビデオの後続するフレームを処理する。ビデオの全てのフレームが処理され終わった後、ステップ1312でループ1301を出て、方法はステップ1313で終了する。

【0135】

ループ1301の中のステップ1302で、プロセッサ2705は、まず、章3.3で先に説明したように現在のフレームfのカットオフ閾値 λ_f を判定する。また、現在のフレームは、現在のフレームを符号化するためのビットバジェットを判定する目標ビットレートを有する。

【0136】

次に、(図7のステップ501~503で示されるように)前処理、成分変換及び量子化などの通常のJ2K符号化ステップがプロセッサ2705により実行される。次に、ループ1303に入り、ビデオフレームの各成分を順次処理する。ループ1303では、プロセッサ2705は、まず、色信号成分を処理し、その後の最終パスで輝度成分を処理する。すなわち、ループ1303は成分を1、2、続いて0の順に処理する。

【0137】

ステップ1304では、現在の成分が2つの色信号成分のうち的一方であるか否かを判定するために、プロセッサ2705により試験が実行される。試験1304が肯定の応答を戻した場合、ステップ1302で判定された λ_f を使用して通常のJ2K符号化プロセスに従って現在の成分を符号化するために、ステップ1305が実行される。現在の成分が符号化された後、ステップ1306で、プロセッサ2705は現在の成分が成分2であるか否かを判定するために試験を実行する。現在の成分が1であれば(ステップ1306のNoオプション)、処理はステップ1308へ続き、現在の成分と関連する値がメモリ906にセーブされる。特に、現在の成分の実現速度が記憶場所913にセーブされる。

【0138】

次のステップ1309で、プロセッサ2705は現在の成分が最後の成分であるか否かを判定するために試験を実行する。試験1309が否定の応答を戻した場合、プロセス流れはステップ1304に戻り、次の成分を処理する。

【0139】

現在の成分が2である場合には(ステップ1306のYesオプション)、メモリ906から先行するフレームの統計値を検索するためにステップ1307が実行される。検索される統計値は先行するフレームf-1の実現成分速度 $R_{achieved}$ を含む。ステップ1307では、プロセッサは輝度成分を符号化するとき使用されるべき λ_f の重み付けを更に計算する。重み付けは、現在のビデオフレームの総合圧縮度を推定するために、現在のフレームfと先行するビデオフレームf-1の双方の色信号成分の実現速度を使用する。例えば、現在の成分の実現速度が先行するビデオフレームの実現速度より速い場合、現在のビデオフレームは先行するビデオフレームと比較して圧縮度が低いと思われる。第1の構成はこの情報を使用して、輝度成分をより有効に符号化する。

【0140】

ステップ1307で現在のフレームがより圧縮度が高いと判定された場合、輝度成分の実現速度を増加させるように λ_f が重み付けされるのが好ましい。現在のフレームがより圧縮度が高いとき、それは色信号成分を符号化するのに使用されるビットの数がより少なく、従って、フレームの目標ビットレート内にとどまりつつ、輝度成分を符号化するためにより多くのビットを利用できることを意味している。

【0141】

ステップ1307で、プロセッサ2705は次の式に従って λ_f を調整する。

$$\lambda_f = \frac{\sum_{m=1,2} R_{achieved}^{f_m}}{\sum_{m=1,2} R_{achieved}^{f-1_m}} \lambda_f$$

10

20

30

40

50

(1 6)

式中、 $R_{achieved}^{f_m}$ は現在のフレーム f における成分 m の実現速度であり、 $R_{achieved}^{(f-1)_m}$ は先行するフレーム $f - 1$ における成分 m の実現速度である。

【 0 1 4 2 】

現在のフレームが先行するフレームより低い圧縮度である場合、式 (1 6) の重み付け係数は 1 に限定される。重み付け係数を 1 に限定する理由は、輝度が色信号成分より大きな影響を認知画質に対して及ぼすからである。従って、圧縮度の低いフレームの場合、輝度成分の符号化は妥協されない。そこで、色信号成分をトリミングすることが必要になるであろう (以下のステップ 1 3 1 0 を参照) 。

【 0 1 4 3 】

ステップ 1 3 0 7 が完了した後、プロセス流れはステップ 1 3 0 8 へ進み、現在の係数の実現速度がメモリ 9 0 6 に格納される。

【 0 1 4 4 】

現在の成分が輝度成分であれば (ステップ 1 3 0 4 の No オプション)、ステップ 1 3 1 4 で、ステップ 1 3 0 7 で判定された調整済み λ_f を使用して成分の符号化が実行される。尚、ステップ 1 3 0 7 で λ_f が減少された場合、一般に、そうでない場合よりも多くのコーディングパスが輝度成分の符号化に使用される。ステップ 1 3 1 4 の後、プロセス流れはステップ 1 3 0 8 へ進み、実現速度をメモリ 9 0 6 にセーブする。

【 0 1 4 5 】

現在のビデオフレーム f の全ての成分が処理された後 (ステップ 1 3 0 9 の Yes オプション)、ステップ 1 3 1 0 に入り、データがトリミングを要求するか否か、すなわち、目標ビットレートを超過したか否かを判定する。トリミングが要求されていれば (ステップ 1 3 1 0 の Yes オプション)、成分 1 及び 2 の圧縮データをトリミングするためにステップ 1 3 1 1 が実行される。圧縮データのトリミングは、 $i_n \geq Z$ (所定の閾値) である成分 1 及び 2 のコードブロック n_f のコーディングパス i_n の圧縮データを放棄することから成る。閾値 Z は次のように判定されるのが好ましい。

$$Z = \frac{\sum_{m=1,2} R_{achieved}^{f_m}}{\sum_{m=1,2} R_{achieved}^{f-1_m}} \lambda_f$$

(1 7)

トリミングステップ 1 3 1 0 及び 1 3 1 1 は図 7 の符号化後トリミングステップ 7 0 6 に対応している。

【 0 1 4 6 】

ステップ 1 3 1 1 が完了した後、ステップ 1 3 1 2 は、現在のフレームが最終フレームであるか否かを検査する。フレームが残っていれば (ステップ 1 3 1 2 の No オプション)、プロセス流れはステップ 1 3 0 2 に戻り、次のフレームを処理する。それ以上フレームが存在しない場合には、プロセスはステップ 1 3 1 3 で終了する。

【 0 1 4 7 】

トリミングプロセスが要求されない場合 (ステップ 1 3 1 0 の No オプション)、ステップ 1 3 1 1 を飛び越し、処理はステップ 1 3 1 2 へ続く。

【 0 1 4 8 】

3 . 5 . 2 色信号カットオフ閾値の重み付け

図 1 4 は、フレーム従属成分重み付け方式 7 1 3 の別の実現形態を示す。この場合、輝度成分のカットオフは調整されず、推定された圧縮度に従って色信号成分の重み付けが重み付けされる。

【 0 1 4 9 】

方法は 1 4 0 0 で始まり、ビデオシーケンスの各フレームを順次処理するループ 1 4 0 1 に入る。全てのビデオフレームが処理された後、ステップ 1 4 1 0 でループ 1 4 0 1 が

10

20

30

40

50

ら出て、方法は1411で終了する。

【0150】

ループ1401の中で、ステップ1402において、プロセッサ2705は現在のビデオフレームfのカットオフ閾値 λ_f を判定する。ステップ1402はステップ1302と同じであり、章3.3で説明した手続きを使用する。次に、現在のフレームに対して前処理及び成分変換などの通常のJ2K符号化処理ステップ(図7のステップ501~503)が実行される。

【0151】

次に、ステップ1403で、現在のビデオフレームの各成分を順次処理するためのループに入る。ループ1403において、プロセッサ2705は、まず、輝度成分を処理し、次に後続するパスで色信号成分の各々を順次処理する。従って、ループ1403は成分を0、1、2の順に処理することになる。

【0152】

ステップ1404では、現在の成分が輝度成分であるか否かを判定するために試験が実行される。試験1404が肯定の応答を戻した場合、プロセス流れはステップ1406へ進み、ステップ1402で判定された λ_f を使用して、通常のJ2K符号化プロセスに従って成分を符号化する。成分が符号化された後、ステップ1407はメモリ906から先行するビデオフレームf-1の関連統計値を検索する。特に、先行するフレームで実現された速度がメモリ913から検索される。

【0153】

ステップ1407は、色信号成分を符号化するときに使用されるべき λ_f の重み付けを更に判定する。現在のフレームf(ステップ1406で判定される)と先行するフレームf-1の双方の輝度成分の実現速度に基づいて、現在のビデオフレームの総合圧縮度を推定することができる。従って、fの輝度成分の実現速度がf-1の輝度成分より高ければ、ビデオfの圧縮度はビデオフレームf-1より低いと思われる。これに従って、fの色信号成分の実現速度を低減するために、カットオフ閾値 λ_f が重み付けされる。

【0154】

閾値 λ_f は次の式に従って重み付けされるのが好ましい。

$$\lambda_f = \frac{R_{achieved}^{f_m}}{R_{achieved}^{f-1_m}} \lambda_f$$

(18)

式中、m = 0、すなわち、輝度成分である。

【0155】

現在の成分に対して符号化が完了した後、次のフレームを処理するとき使用するのに備えて成分の実現速度をメモリ916にセーブするためにステップ1408が実行される。

【0156】

現在の成分が色信号成分の一方である場合には(ステップ1404のNoオプション)、ステップ1405で、先にステップ1407で判定された重み付き λ_f を使用して現在の色信号成分が符号化される。プロセッサ2705で実行されるプログラムは次にステップ1408へ進み、現在の成分の実現速度をメモリ913にセーブする。

【0157】

次に、ステップ1409は現在の成分が最後の成分であるか否かを検査する。フレームの他の成分が存在する場合、プロセス流れはステップ1404に戻り、次の成分を処理する。全ての成分が処理され終わっている場合には(ステップ1409のYesオプション)、ステップ1410は現在のフレームが最終フレームであるか否かを検査する。全てのフレームが処理され終わっている場合(ステップ1410のYesオプション)、ループ1401から出て、方法は1411で終了する。そうでない場合には(ステップ1410

10

20

30

40

50

のNoオプション)、プロセス流れはステップ1402に戻り、次のフレームを順次処理する。

【0158】

3.6 符号化方式

第1の構成は、現在のビデオフレーム f のコードブロック n_f ごとの過剰符号化の量を減少させる符号化方式714を使用する。符号化方式714は、フレーム f 中のコードブロックの符号化を終了させるべき時点を決断するために、ビデオシーケンスの先行するフレームにおけるコードブロックの切捨てポイントを使用する。

【0159】

図15は、図7の符号化プロセスで使用された符号化方式714を示す。明瞭にするため、図15は図7のステップの全てではないが、その一部を再現している。プロセスはステップ700で始まり、ビデオシーケンスの各フレーム f を順次処理するためのループ701に入る。ループ701は第1回目の繰り返しでビデオシーケンスの第1のフレームを処理し、後続する繰り返しの中で、ループ701はビデオの対応する後続フレームを処理する。ビデオの全てのフレームが処理された後、ステップ709でループ701を出て、方法はステップ710で終了する。

10

【0160】

ループ701の中で、符号化方法は図15に示されていないいくつかのステップを実行する。ビデオフレームごとに、方法はカットオフ閾値 τ を推定し(図13のステップ1302)、次に、プロセッサ2705は前処理、離散的ウェーブレット変換(DWT)及び量子化から成る標準J2K符号化ステップ(図7のステップ501~503)を実行する。次に、ループ701は、ビデオフレーム f の各コードブロック n_f を順次処理するための別のループ1502に入る。全てのコードブロック n_f が処理され終わった後、ステップ1509でループ1502を出て、処理は、コードブロック n_f の切捨てポイントを格納するステップ1510へ続く。

20

【0161】

ループ1502の中で、ステップ1503において、プロセッサ2705は先行するフレーム $f-1$ の対応するコードブロック n_{f-1} の切捨てポイントをメモリ906から検索する。尚、第1のフレームを符号化するときを使用すべき先行するフレームからの情報は存在しないので、第1のフレームは特殊ケースとして扱われることに注意する。従って、第1のフレームの場合、ステップ1503及び1504は省略され、第1のフレームの符号化はステップ1505で始まる。

30

【0162】

ステップ1504では、現在のコードブロック n_f の符号化を進めるべきか否かを判定するために試験が実行される。ステップ1504は、現在のコードブロックが静的ブロックであるか否か及びコードブロック n_{f-1} が符号化されたか否か(すなわち、ビットストリームの中で有意ビットプレーンが符号化されたか否か)を検査する。コードブロック n_f が静的ブロックであるか否かを判定するための手続きについては、先に図11を参照して説明した。コードブロック n_f が静的コードブロックであると判定され、コードブロック n_{f-1} は符号化されていなかった場合(ステップ1504のYesオプション)、処理はステップ1509へ進み、コードブロック n_f の符号化は実行されない。しかし、試験1504が否定の応答を戻した場合には(ステップ1504のNoオプション)、プロセス流れはステップ1505へ進む。

40

【0163】

ステップ1505では、コードブロック n_f の各コーディングパスを符号化するためのループに入る。特定の1つのコードブロックについて符号化すべきコーディングパスの最大数は、符号化に先立って判定される(図示せず)。パスの最大数は標準JPEG2000方法を使用して判定される。尚、J2Kでは、ビットプレーンごとに名目上は3つのコーディングパスが存在することに注意する。

【0164】

50

ループ1505の中で、ステップ1506においては、プロセッサ2705はコードブロック n_f の量子化ウェーブレット係数を符号化する。最初にループ1505に入るとき、ステップ1506はクリーンアップパスを実行する。それに続く繰り返しでは、ステップ1506は、有意性伝播パスから再開する前に、有意性伝播パス、マグニチュード精密化パス又はクリーンアップパスをこの順序で実行する。1つのコーディングパスを符号化した後、符号化ステップ1506は現在のコーディングパスと関連するひずみ値及び速度を判定する。ひずみ及び速度は、現在のコーディングパスの後に符号化を終了させるべきか否かを判定するために使用される。終了試験はコーディングパス1506が完了するたびにステップ1507で実行されるが、この試験については以下に図16を参照して更に詳細に説明する。

10

【0165】

終了試験1507が否定であり且つ処理すべきそれ以上のコーディングパスが存在しない場合、処理はステップ1506に戻り、次のコーディングパスを処理する。しかし、終了試験1507が肯定であれば、現在のコーディングパスは最終コーディングパスと判定され、プロセス流れはステップ1508へ進み、コードブロック n_f の符号化を終了する。

【0166】

ステップ1507は、コードブロック n_f の終了ポイントを確認する。次に、ステップ1508で、コードブロック n_f に対してトリミングが適用される。トリミングは、コードブロック n_f の切捨てポイントを判定することと、速度-ひずみ傾き i_n を有するコーディングパス i_n の圧縮データを放棄することから成る。ステップ1508で実行されるトリミングは、ステップ706の符号化後トリミングに対応している。

20

【0167】

トリミングが完了した後、ステップ1509で、プロセッサ2705は現在のコードブロックが最後のコードブロックであるか否かを試験する。処理すべきコードブロックがそれ以上存在しない場合(ステップ1509のNoオプション)、処理はステップ1503に戻る。しかし、全てのコードブロックが処理され終わっている場合には(ステップ1509のYesオプション)、ループ1502から出る。

【0168】

全てのコードブロックが処理された後、図15には示されていないが、いくつかの標準J2K符号化ステップが実行される。符号化ステップはビットストリーム編成(図7のステップ506)を含む。

30

【0169】

次に、ステップ1510で、次のビデオフレーム($f+1$)を処理するときを使用するために、全てのコードブロックの切捨てポイントがメモリ911に格納される。次に、ステップ709は現在のフレームが最終フレームであるか否かを試験する。これが当てはまらない場合(ステップ709のNoオプション)、プロセス流れはステップ1502に戻り、シーケンス中の次のフレームを符号化する。それ以上のフレームが存在しない場合には(ステップ709のYesオプション)、方法はステップ710で終了する。

【0170】

3.6.1 終了基準

次に、図16を参照して、コードブロック n_f の終了ポイントを判定する手続きを更に詳細に説明する。図16において、ステップ1603~1608は図15のステップ1507に対応している。第1の構成では、コーディングパスの後にコードブロック n_f を終了させるための決定はコードブロック n_{f-1} の切捨てポイントを利用する。

40

【0171】

図16を参照すると、図15を参照して先に説明したようにコードブロック n_f の各コーディングパスを処理するためにループ1505に入る。要求される全てのコーディングパスを符号化した後、ループ1505を出て、プロセス流れはステップ1508へ進み、符号化プロセスを終了する。尚、コードブロック n_f の符号化プロセスの終了は、通常、

50

コードブロック n_f の最終コーディングパスを符号化する前に起こることに注意する。

【0172】

ステップ1603は、現在のコードブロック n_f が静的コードブロックであるか否か及び現在の符号化プロセスが対応するコードブロック n_{f-1} と同じ切捨てポイント（ビットプレーン）に到達したか否かを試験する。これら2つの条件に適合していると（ステップ1603のYesオプション）、コードブロック n_f の符号化は終了し、プロセス流れはステップ1508へ進み、符号化を終了する。しかし、現在のコードブロックが静的ブロックではない場合又は対応する先行するコードブロックの切捨てポイントにまだ到達していない場合には（ステップ1603のNoオプション）、プロセス流れはステップ1604へ進み、更に終了試験が実行される。コードブロックが静的ブロックであるか否かを判定する手続きについては、先に図11に関して説明した。

10

【0173】

コードブロック n_f に対して、符号化ステップ1506は現在のコーディングパス i_n の速度 - ひずみ傾き i^n を計算する。ステップ1604は、速度 - ひずみ傾き i^n が現在のフレームのカットオフ閾値 θ_f より大きいと否かを試験する。カットオフ閾値の計算については先に章3.3で説明した。現在のコーディングパスの速度 - ひずみ傾きがカットオフ閾値より大きい場合（ステップ1604のNoオプション）、処理はステップ1506に戻り、次のコーディングパスを符号化する（すなわち、 $i_n = i_n + 1$ ）。しかし、現在のコーディングパスの速度 - ひずみ傾きがカットオフ閾値より小さい場合には（ステップ1604のYesオプション）、プロセス流れはステップ1605へ進み、更に試験が実行される。

20

【0174】

ステップ1605はループ1505においてコードブロック n_f に関してそれまでに処理されたすべてのコーディングパスを検査し、0でない i^n を有するクリーンアップコーディングパスが存在するか否か、すなわち、現在のコードブロックについて少なくとも1つのクリーンアップパスで有意ビットが符号化されたか否かを判定する。試験1605が肯定の応答を戻した場合（すなわち、クリーンアップパスで以前に有意ビットが符号化されていた場合）、プロセス流れはステップ1606へ進み、更に試験が実行される。試験1605が否定である場合（すなわち、0でない i^n を有するクリーンアップコーディングパスが存在する場合には、プロセス流れはステップ1506に戻り、次のコーディングパスを符号化する。

30

【0175】

ステップ1606では、プロセッサ2705は、コードブロック n_{f-1} で3つ以上の有意ビットプレーンが符号化されたか否かを試験する。試験1606が否定の応答を戻した場合、処理はステップ1608へ続く。試験1606が肯定の応答を戻した場合（すなわち、コードブロック n_{f-1} で3つ以上の有意ビットプレーンが符号化されていた場合には、ステップ1607で別の試験が実行される。ステップ1607で、コードブロック n_f についてそれまでに3つ以上の有意ビットプレーンが符号化されていた場合（ステップ1607のYesオプション）、処理はステップ1608へ続く。しかし、試験1607が否定の応答を戻した場合には、処理はステップ1506に戻り、次のコーディングパスを処理する。

40

【0176】

ステップ1608では、ステップ1506でコーディングパス i_n により生成された圧縮データがAバイトの長さを超えるか否かを判定するために、更に試験が実行される。第1の構成では、 $A = 3$ バイトであるのが好ましい。試験1608が否定の応答を戻した場合、処理はステップ1506に戻り、別のコーディングパスを処理する。しかし、試験1608が肯定の応答を戻した場合は、生成されたコード語がAバイトを超える場合には、ループ1505を出て、プロセス流れはステップ1508で再開する。ループ1505を出ることにより、現在のコーディングパスはコードブロックの終了ポイントとして確定される。

50

【 0 1 7 7 】

第1の構成の変形においては、例えば、バイトがクリーンアップパスなどの特定の種類のコーディングパスによりコード語に寄与されたか否かを検査するなど、追加の終了条件も可能である。更に、異なる値のAを使用しても良いであろう。

【 0 1 7 8 】

3.7 符号化後トリミング

先に述べた通り、第1の構成は、以下に図17を参照して説明する符号化後トリミングの方法を使用する。図17のステップは図7のステップ706に対応しており、図15のステップ1508のトリミングにおいても使用される。トリミング方法は、1つのコードブロックの符号化が終了した後に一度適用される。コードブロックを符号化することにより、第1のコーディングパスから符号化が終了した最終コーディングパスまでの一連のコーディングパスが得られる。各コーディングパスと関連して、速度 - ひずみ傾きが存在する。

10

【 0 1 7 9 】

トリミング方法はステップ1700で始まり、コードブロック n_f の第1のコーディングパスから始めて、コードブロック n_f のコーディングパスを巡って繰り返すためのループ1701に入る。

【 0 1 8 0 】

ループ1701の中で、ステップ1702では、プロセッサ2705は、現在のコーディングパス i_n の速度 - ひずみ傾き i_n が現在のフレーム f のカットオフ閾値 τ_f 以下であるか否かを判定する。試験1702が否定の応答を戻した場合（すなわち、 $i_n > \tau_f$ ）、現在のコーディングパスはコード語に含められるべく保持され、処理はステップ1703へ進み、現在のコーディングパスがコードブロック n_f の最後のコーディングパスであるか否かを判定する。現在のコーディングパスが最後のコーディングパスではない場合（ステップ1703のNoオプション）、ステップ1704はコーディングパス i_n の指標を増分し、プロセス制御はステップ1702に戻り、次のコーディングパスを考慮する。符号化は最大数のコーディングパスまでしか進まない。試験1703が肯定の応答を戻した場合（すなわち、現在のコーディングパスが最後のコーディングパスである場合）には、ループ1701を出て、トリミング方法はステップ1706で終了する。

20

【 0 1 8 1 】

$i_n < \tau_f$ であれば（すなわち、ステップ1702のYesオプション）、コーディングパス i_n と関連する圧縮データはステップ1705で放棄され、方法はステップ1706で終了するので、後続するコーディングパスは放棄される。

30

【 0 1 8 2 】

トリミングが完了した後、最後に残っているコーディングパスがコードブロック n_f の切捨てポイントである。

【 0 1 8 3 】

方法1700は、システム2701がコードブロック n_f の符号化を完了した後に実行されるのが好ましい。方法1700はコードブロック n_f の圧縮データをトリミングするために使用され、トリミングの後も残っている圧縮データはビデオフレーム f の最終ビットストリームを構成するために保持される。この方法は、コードブロックを τ_f により判定される切捨てポイントに切り捨てることとして考えられることも可能である。

40

【 0 1 8 4 】

別の実現形態においては、トリミング方法706はMJ2K符号化プロセスの間のいくつかの場所で、例えば、1つのサブバンドを符号化した後又は分解能レベルを符号化した後のいずれかで実現されることが可能である。

【 0 1 8 5 】

符号化後トリミング方法706は符号化後ひずみ最適化の一形態として考えられることが可能であり、ひずみの最適化が符号化に先立って判定される τ_f の値を使用して実現されるという点で、JPEG2000により推奨されているPCRD-optアルゴリズム

50

とは異なる。この方法とは対照的に、PCRD-optは符号化の後に f を選択する。

【0186】

4.0 第2の構成

ビデオシーケンスを圧縮するための第2の構成が図18に示されている。第2の構成は図7に関して説明した第1の構成と共通する要素をいくつか有するが、異なる符号化方式が使用されるという点で異なっている。現在のコードブロックについてコーディングパスを符号化するとき、第2の構成は次のコーディングパスの速度-ひずみ傾きを予測し、符号化を終了させるべきか否かを判定する際に、予測された傾きを使用する。

【0187】

そこで、図18を参照して、ビデオシーケンスを処理するための第2の構成の概要を説明する。方法は700で始まり、次に、ビデオシーケンスの第1のフレームから始めて、シーケンスの各フレーム f を順次処理するためのループ701に入る。全てのフレームが処理された後、ステップ709でループを出て、方法1800はステップ710で終了する。

10

【0188】

ループ701の中で、ステップ501において、プロセッサ2705は、必要に応じたレベルシフティング及び成分変換を含めて、現在のフレームの必要な前処理を実行する。前処理後、ステップ502で、プロセッサ2705により画像に離散的ウェーブレット変換が適用される。次に、ステップ503で、ウェーブレット変換された係数が整数値に量子化される。ウェーブレット係数を量子化した後、プロセッサ2705はフリッカ減少方式711を適用するのが好ましい。ここまでの方法の流れは第1の構成の方法と同じである。

20

【0189】

次に、ステップ1805で、現在の画像の処理済み量子化ウェーブレット係数が複数のコードブロックとして配列され、各コードブロックは算術エンコーダによってビットプレーンで符号化される。コードブロックの符号化1805は、起こりうる場合に過剰符号化を回避するための符号化方式1814により管理される。

【0190】

加えて、コードブロック n_f の符号化1805は先に説明した成分重み付け方式713を利用するのが好ましい。更に、コードブロック n_f を符号化している間、先に説明した静的ブロック切捨て安定化方式712により、ステップ1805の符号化プロセスを終了させるための決定が実行されるのが好ましい。

30

【0191】

現在のフレーム f のコードブロックについて符号化が完了した後、更に速度制御を実行するために、符号化後トリミング方式706が任意に適用される。

【0192】

次に、ステップ506で、圧縮ビデオフレームを出力するために、フレーム f の符号化データがビットストリームとして編成される。圧縮フレームはコンピュータネットワーク2720を介して送信されても良いし、あるいは後の使用に備えて、例えば、CDROM2712に格納されても良い。

40

【0193】

ステップ709では、 f がビデオシーケンス中の最後のビデオフレームであるか否かを判定するための試験に入る。試験709が否定の応答を戻した場合、処理はステップ501に戻り、次のビデオフレームを処理する。試験709が肯定の応答を戻した場合には、ループ701を出て、方法はステップ710で終了する。

【0194】

4.1 速度-ひずみ傾きの予測を伴う符号化方式

第2の構成は、各コードブロックの過剰符号化を回避するために符号化方法1814を使用する。次に、図19を参照して、符号化方式1814を使用するエントローピー符号化1805を説明する。明瞭にするため、図18のステップの全てではないが、その一部が

50

図 19 でも再現されている。

【 0 1 9 5 】

方法はステップ 700 で始まり、第 1 のフレームから始めて、ビデオの各フレーム f を順次処理するループ 701 に入る。ビデオの全てのフレームを処理し終わった後、ステップ 709 でループ 701 を出て、方法はステップ 710 で終了する。

【 0 1 9 6 】

ループ 701 の中で、プロセッサ 2705 は、ビデオフレームごとに、前処理、DWT 及び量子化などの通常の J2K 符号化ステップ (図 19 には示されていないステップ 501 ~ 503) を実行する。次に、方法は、フレーム f 中の各コードブロック n_f を処理するための別のループ 1902 に入る。1つのビデオフレームの全てのコードブロックを処理した後、1910 でループ 1902 を出る。続くステップ 1911 は各コードブロック n_f の切捨てポイントを格納する。

10

【 0 1 9 7 】

ループ 1902 の中で、ステップ 1903 において、プロセッサ 2705 は、まず、コードブロック n_{f-1} の切捨てポイントを検索する。尚、 $f = 1$ である場合、それは特殊ケースとして扱われるため、このステップを飛び越すことに注意する。第 1 のフレームの場合には、ステップ 1903 に入らずに、処理はステップ 1904 に入る。

【 0 1 9 8 】

ステップ 1904 では、コードブロック n_f の符号化を進めるべきか否かを判定するために試験が実行される。コードブロック n_f が静的コードブロックであると判定され且つコードブロック n_{f-1} が符号化されていなかった場合、試験 1904 は肯定であり、その場合、処理はステップ 1910 へ進み、コードブロック n_f に対して符号化は実行されない。コードブロックが静的ブロックであるか否かを判定する手続きについては、先に図 11 を参照して説明した。

20

【 0 1 9 9 】

しかし、試験 1904 が否定の応答を戻した場合には、現在のコードブロックを符号化するためにプロセス流れはステップ 1905 へ進む。ステップ 1905 で、コードブロック n_f の各コーディングパス i_n を符号化するためのループに入る。特定の 1つのコードブロックに対して符号化すべきコーディングパスの最大数は、符号化に先立って判定される。尚、J2K の場合、ビットプレーンごとに名目上は 3つのコーディングパスが存在することに注意する。

30

【 0 2 0 0 】

ステップ 1906 は、通常の JPEG 2000 符号化方法を使用して現在のコードブロックの現在のコーディングパスを符号化する。初めにループ 1905 に入った後、ステップ 1906 はクリーンアップパスを実行する。後続する繰り返しにおいては、ステップ 1906 は、有意性伝播パスによってサイクルを再開する前に、有意性伝播パス、マグニチュード精密化パス及びクリーンアップパスをこの順序で実行する。ステップ 1906 は、現在のコーディングパス i_n と関連するひずみ値及び速度を判定する。

【 0 2 0 1 】

次に、ステップ 1907 で、次のコーディングパス i_{n+1} の速度 - ひずみ傾き i_{n+1} を推定するために、プロセッサ 2705 により速度値及びひずみ値が使用される。推定は、現在のコードブロックについてそれまで処理されたコーディングパスの以前の速度 - ひずみ傾きに基づいて実行される。推定方法については、以下に図 21 を参照して更に詳細に説明する。

40

【 0 2 0 2 】

次に、ステップ 1908 は、現在のコーディングパスの後で符号化プロセスを終了すべきか否かを試験する。終了試験 1908 が否定であり且つ処理すべきコーディングパスがまだ存在している場合、プロセス流れはステップ 1906 に戻り、次のコーディングパスを符号化する。しかし、終了試験 1908 が肯定である場合には、それは現在のコーディングパスが最終コーディングパスであると判定されたことを意味しているため、プロセス

50

流れはステップ1909へ進み、コードブロック n_f の符号化を終了する。符号化プロセスを終了させることは、算術符号化エンジンを終了させること、資源クリーンアップなどを含む。

【0203】

符号化が完了すると、ステップ1910は処理すべきコードブロックがまだ存在しているか否かを試験する。存在していれば（ステップ1910のYesオプション）、処理はステップ1903に戻り、次のコードブロックを処理する。試験1910が否定の応答を戻した場合には、ステップ1910でループ1902を出る。全てのコードブロックが処理された後、ビットストリーム編成などのJ2K符号化ステップが実行される（図示せず）。

10

【0204】

ステップ1911では、次のビデオフレームを処理するときを使用するために、全てのコードブロックの切捨てポイントがメモリ906に格納される。次に、ステップ709は現在のフレームが最終フレームであるか否かを試験する。試験709が肯定の応答を戻した場合、プロセス流れはステップ1902に戻り、次のフレームを処理する。試験709が否定の応答を戻した場合には、ループ701を出て、方法はステップ710で終了する。

【0205】

4.1.1 終了基準

図20(a)及び図20(b)は、ステップ1908の終了試験を説明するフローチャートである。明瞭にするため、図20(a)及び図20(b)は図19のステップの全てではないが、その一部を再現している。現在のコーディングパスの後にコードブロック n_f の符号化を終了すべきか否かの決定は、次のコーディングパスの推定速度 - ひずみ傾き i_{n+1} 及びコードブロック n_{f-1} の切捨てポイントを使用する。

20

【0206】

図20(a)及び図20(b)を参照すると、ステップ1905で、ビデオフレーム f の現在のコードブロックの各コーディングパスの符号化を処理するためのループに入る。要求される全てのコーディングパスについて符号化を実行した後、ステップ1909でループ1905を出て、現在のコードブロックの符号化プロセスを終了する。

【0207】

ループ1905は、現在のコードブロックの第1のコーディングパスから最後のコーディングパスまでコーディングパス(i_n)を繰り返す。まず、ステップ1906は、先に説明したように、3つのパスのうちの1つでコードブロック n_f の量子化ウェーブレット係数を符号化する。次に、ステップ2003では、コードブロック n_f が静的コードブロックとして分類されるか否か及びコードブロック n_{f-1} の最後のB個のビットプレーンの中に符号化が到達しているか否か（すなわち、現在のコードブロックの符号化ビットプレーンがコードブロック n_{f-1} の切捨てビットプレーンからB個のビットプレーンを減算したビットプレーンを超えているか否か）を判定するために試験が実行される。B=1であるのが好ましいが、他の値のBを使用することも可能である。試験2003が否定の応答を戻した場合、処理はステップ2005へ続き、 i_{n+1} は推定されない。しかし、試験2003が肯定の応答を戻した場合には、ステップ1907で i_{n+1} が推定される。

30

40

【0208】

処理はステップ2005へ続き、コードブロック n_f が静的コードブロックであるか否か及びコーディングパス i_n がコードブロック n_{f-1} の切捨てポイントと等しいか否かを判定するために試験が実行される。試験2005が肯定の応答を戻した場合、コードブロック n_f の符号化は終了し、プロセス流れはステップ1909へ進み、符号化を終了する。試験2005が否定の応答を戻した場合には、ステップ2006で更に試験が実行される。

【0209】

ステップ2006は推定された i_{n+1} （ステップ1907で判定される）が f より

50

大きいか否かを試験する。これが当てはまる場合、処理はステップ1906に戻り、コードブロック n_f の次のコーディングパス(すなわち、 $i_n = i_n + 1$)を符号化する。試験2006が肯定である場合(すなわち、推定された速度 - ひずみ傾きがカットオフ閾値より小さい場合)には、ステップ2007で更に試験が実行される。

【0210】

ステップ2007は、既に符号化されているコーディングパスの中で0ではないクリーンアップパス速度 - ひずみ傾きを求めて試験する。それが存在しない場合、処理はステップ1906に戻り、次のコーディングパスを処理する。言い換えれば、ステップ2007はコードブロック n_f についてそれまでに処理された全てのコーディングパス i_n を検査し、0でない i_n を有するクリーンアップコーディングパスが存在するか否かを判定するのである。試験2007が肯定である場合、ステップ2008で更に試験が実行される。

10

【0211】

ステップ2008は、コードブロック n_{f-1} において3つ以上の有意ビットプレーンが符号化されたか否かを判定する。試験2008が否定の応答を戻した場合、処理は2010へ続く。試験2008が肯定の応答を戻した場合には、ステップ2009はコードブロック n_f においてこれまでに3つ以上の有意ビットプレーンが符号化されているか否かを試験する。これが当てはまる場合、処理はステップ2010へ続く。しかし、試験2009が否定の応答を戻した場合(すなわち、符号化されているビットプレーンが2つに満たない場合)には、処理はステップ1906に戻り、次のコーディングパスを符号化する。

【0212】

20

ステップ2010では、ステップ1906においてコーディングパス i_n により生成された圧縮データの長さがEバイトを超えるか否かを判定するために更に試験が実行される。第2の構成においては、 $E = 3$ バイトであるのが好ましい。試験2010が否定の応答を戻した場合、処理はステップ1906に戻り、更にコーディングパスを処理する。試験2010が肯定の応答を戻した場合には、ループ1905を出て、コードブロック n_f の符号化プロセスは1909で終了する。

【0213】

尚、ステップ1907で $i_n + 1$ の推定を実行するために使用されるコードブロック n_f の i_n は、コードブロック n_f の速度 - ひずみ凸閉包上に位置する傾きであることに注意する。言い換えれば、コードブロック n_f の凸閉包はコーディングパスが終了するごとに計算されるということになる。

30

【0214】

第2の構成の更に別の変形においては、図20(a)及び図20(b)のステップ2003 ~ 2010で説明された符号化終了条件とは別の追加終了条件を含めることができる。

【0215】

4.1.2 次の速度 - ひずみ傾きの推定

$i_n + 1$ を推定する方法が図21に示されている。図21のステップはステップ1907に対応している。

【0216】

40

ステップ2101で、コードブロック n_f についてそれまでに生成された速度 - ひずみ傾きポイントの凸閉包が形成される。凸閉包は次のようにして形成される。

【0217】

1. コードブロック n_f について各コーディングパスの速度 - ひずみ傾きをグラフに表す。図22は、速度 - ひずみ傾きとコーディングパスとの関係を表すグラフの一例を示す。y軸は速度 - ひずみ傾きであり、x軸は対応するコーディングパスである。図22の例は8つのコーディングパスに関わる傾きを示す。傾き2208は第1のコーディングパスに対応し、傾き2207は第2のコーディングパスに対応し、傾き2206は第3のコーディングパスに対応し、それ以降も同様であり、第8のコーディングパスに対応する傾き2201に至る。一般に、連続するコーディングパスごとに傾きは小さくなるが、その

50

減少は単調ではない。

【0218】

2. コードブロック n_i の最後のコーディングパスから、第1のコーディングパスに向かって逆戻りし、速度 - ひずみ傾きの値が増加しない場合、速度 - ひずみ傾きを0に設定する。図22の例では、傾き2201から始めて、傾き2202の値は傾き2201の値より小さいため、速度 - ひずみ傾き2202は0に設定される。傾き2203の値は速度 - ひずみ傾き2201より大きいいため、速度 - ひずみ傾き2203は凸閉包に含まれる。同様に、傾き2204は凸閉包に含まれるが、傾き2205は0に設定される。このプロセスが傾き2208まで繰り返される。

【0219】

コーディングパス i_n の速度 - ひずみ傾き i^n が凸閉包上にない場合、その速度 - ひずみ傾きは0に設定される。凸閉包が形成されると、ステップ2102は凸閉包上の速度ひずみ傾きを通る直線を当てはめる。これは図23に示されており、直線2301は図22の凸閉包の傾きに当てはめられている。

【0220】

ステップ2103では、速度 - ひずみ傾き i^{n+1} を推定するために、当てはめられた直線が線形補外される。例えば、図23を参照すると、コーディングパス9(2303)の速度 - ひずみ傾き i^{n+1} 2303は、それが直線2301上に位置すると仮定することにより推定される。

【0221】

他の変形においては、 i^{n+1} の推定はこれまで説明した線形推定ではなく、他の形態の推定によって(例えば、スプライン推定を使用して)獲得される。更に、別の推定プロセスの変形例は、ステップ2101で凸閉包を形成せずに、クリーンアップパスのみの i^n 値を使用する。

【0222】

5.0 フリッカ減少

図7及び図18に示すように、第1及び第2の構成はフリッカ減少方法711を利用するのが好ましい。

【0223】

モーションJPEG2000の非線形量子化プロセスは、元のビデオシーケンスに存在するセンサ雑音を増幅する。この雑音増幅を安定化させることが望ましい。モーションJPEG2000においては、現在のフレームのウェーブレット係数を平滑化するために先行するフレームのウェーブレット係数を使用するような平滑化技法が使用されてきた。係数の格納と関連するメモリのコストは著しく高いものになる可能性がある。

【0224】

しかし、フリッカアーティファクトを回避するために、先行するフレームのウェーブレット係数の全てを格納する必要はない。全ての係数ではなく、1つの係数のうちの2ビットのみを使用して平滑化を実現できる。従って、先行するフレームの係数を格納するときには32ビットが通常に使用された場合、以下に説明する2ビット平滑化技法は先行するフレーム全体を格納するために要求されるメモリを1/16に減少させる。

【0225】

雑音の非線形増幅をウェーブレット係数のビットパターンの中における雑音の伝播としてみなすことができる。以下に図24～図26を参照して説明する技法は、現在のフレームのウェーブレット係数のビットを介して伝播する増幅を平滑化する。要求されるのは、先行するフレームの対応する係数のうちの2ビットのみである。

【0226】

いくつかの制約の下で、現在のウェーブレット係数に下記の公式を適用することにより、フリッカは平滑化される。

10

20

30

40

$$|C^f| = |C^f| + (a_{L(f)}^f - a_{L(f)}^{f-1}) * (2 * |a_{L(f)-1}^f - a_{L(f)-1}^{f-1}| - 1) * 2^{-L(f)}$$

(1 9)

式中、 $|C^f|$ は係数のマグニチュードを表し、 $a_p \in \{0, 1\}$ である。

【 0 2 2 7 】

先に定義した通り、値 a_p は係数の小数部の p 番目の 2 進数字を表す。指標 f は現在のフレームを表し、 $(f - 1)$ は先行するフレームを表す。

【 0 2 2 8 】

切捨てビットプレーン $L(f)$ は係数を平滑化するための基準として使用されるため、式 (19) の重要なパラメータである。 $L(f)$ は現在のフレームにおける切捨てビットプレーンである。しかし、切捨てビットプレーンはコードブロックの符号化が完了するまで利用できないであろう。従って、その代わりに、構成は対応する先行するコードブロックの切捨てビットプレーン $L(f - 1)$ を利用する。

【 0 2 2 9 】

式 19 の方法バージョンは次のようになる。

$$|C^f| = |C^f| + (a_{L(f-1)}^f - a_{L(f-1)}^{f-1}) * (2 * |a_{L(f-1)-1}^f - a_{L(f-1)-1}^{f-1}| - 1) * 2^{-L(f-1)}$$

(2 0)

図 24 は、フリッカ平滑化技法 711 で使用される 2 進数字の位置を示す概略図である。図 24 は、現在のフレーム f における係数 2404 と、先行するフレーム $(f - 1)$ における対応する係数 2402 とを示す。図示の便宜上、それら 2 つの係数は 8 ビットを有するものとして示されている。係数 2402 の最上位ビットはビット 2406 であり、係数 2402 の最下位ビットはビット 2409 である。ビットの指標付けは最上位ビットで始まり、最下位ビットに到達するまで増分される。

【 0 2 3 0 】

線 2410 はフレーム $(f - 1)$ における係数の切捨てビットプレーン $L(f - 1)$ を表す。この例では、係数 2402 の 3 つの最下位ビットが放棄される。

【 0 2 3 1 】

ビット 2407 は $a_{L(f-1)-1}^{f-1}$ である。ビット 2408 は $a_{L(f-1)}^{f-1}$ である。ビット 2407 及び 2408 は係数 2402 の切捨て形態の 2 つの最下位ビットである。

【 0 2 3 2 】

現在のフレーム f の係数 2404 において、ビット 2413 は $a_{L(f-1)-1}^f$ である。ビット 2414 は $a_{L(f-1)}^f$ である。ビット 2415 は $a_{L(f-1)+1}^f$ である。従って、係数 2404 がビットプレーン $L(f - 1)$ で切り捨てられた場合、ビット 2415 が放棄される最上位のビットであり、ビット 2413 及び 2414 は係数 2404 の切捨て形態に残る 2 つの最下位ビットである。

【 0 2 3 3 】

フリッカ減少技法は信号と雑音を区別しない。その結果、現在のフレーム f のウェーブレット係数が先行するフレーム $f - 1$ の対応するウェーブレット係数と異なる時、その差が真に信号に存在しており、雑音の結果ではないとしても、現在の係数は平滑化される。しかし、ウェーブレット係数のマグニチュードの変化はごくわずかであるため、それは顕著ではない。

【 0 2 3 4 】

図 25 は、係数ごとに先行するフレームからの 2 ビットを使用してモーション JPEG 2000 ビデオシーケンスのフリッカを減少させる技法のフローチャートを示す。フリッカ平滑化はプロセスのステップ 2504 及び 2508 で実行される。図 25 のその他のステップは標準モーション JPEG 2000 符号化技法を使用して実現されることが可能で

10

20

30

40

50

ある。

【 0 2 3 5 】

方法はステップ 2 5 0 0 で始まり、必要な全ての変数が初期設定される。次に、ビデオシーケンスにおいて一度に 1 フレームを圧縮するためのループ 2 5 0 1 が開始される。ループ 2 5 0 1 が全てのフレームを処理し終わると、ステップ 2 5 0 9 でループから出て、処理はステップ 2 5 1 0 で終了する。

【 0 2 3 6 】

ループ 2 5 0 1 の中で、ステップ 2 5 0 2 において現在のフレームに離散的ウェーブレット変換 (DWT) が適用される。次に、各々の繰り返しが現在のフレームの 1 つのコードブロックを符号化するループ 2 5 0 3 に入る。

10

【 0 2 3 7 】

ステップ 2 5 0 4 では、現在のコードブロックのウェーブレット係数に 2 ビットフリッカ平滑化技法が適用される。ステップ 2 5 0 4 については、図 2 6 を参照して更に詳細に説明する。

【 0 2 3 8 】

次に、ステップ 2 5 0 5 で、現在のコードブロックの算術符号化が処理される。次に、ステップ 2 5 0 6 で、プロセッサ 2 7 0 5 は現在のコードブロックがフレーム中の最後のコードブロックであるか否かを検査する。フレーム中にコードブロックがまだ存在している場合 (ステップ 2 5 0 6 の No オプション)、プロセス制御はステップ 2 5 0 4 に戻り、次のコードブロックが処理される。現在のブロックが現在のフレームにおける最後のブロックである場合には (ステップ 2 5 0 6 の Yes オプション)、プロセス流れはステップ 2 5 0 7 へ進む。

20

【 0 2 3 9 】

ステップ 2 5 0 7 では、目標ビットレートに適合するように、全てのコードブロックの切捨て深さが判定される。ステップ 2 5 0 7 に適する技法は、先に図 6 を参照して説明した、J P E G 2 0 0 0 規格により推奨されている P C R D - o p t アルゴリズムである。切捨てポイント情報を提供するために、他のビットレート制御技法が使用されても良い。この切捨てポイント情報はモーション J P E G 2 0 0 0 規格に従ったデコーダの必要条件である。現在のフレームのコードブロックの切捨てポイントは、次のフレームを符号化するときを使用するためにメモリに格納される。

30

【 0 2 4 0 】

ステップ 2 5 0 8 では、現在のフレームの各ウェーブレット係数の 2 つのビット $a_{L(f)}$ 及び $a_{L(f)-1}$ が取り出される。係数ごとに取り出される 2 つのビットは、ビデオシーケンスの次のフレームを処理するために使用するためにメモリに格納される。

【 0 2 4 1 】

次に、ステップ 2 5 0 9 で、プロセッサ 2 7 0 5 は現在のフレームがビデオシーケンス中の最終フレームであるか否かを検査する。残っているフレームがある場合 (ステップ 2 5 0 9 の No オプション)、プロセス流れはステップ 2 5 0 2 に戻り、次のフレームに DWT が適用される。しかし、現在のフレームが最終フレームである場合には (ステップ 2 5 0 9 の Yes オプション)、処理はステップ 2 5 1 0 で終了する。

40

【 0 2 4 2 】

図 2 6 は、ステップ 2 5 0 4 の 2 ビットウェーブレット係数平滑化を更に詳細に示すフローチャートである。2 ビットフリッカ平滑化技法はステップ 2 6 0 0 で始まる。次に、ステップ 2 6 0 1 は現在のフレームがシーケンス中の第 1 のフレームであるか否かを試験する。第 1 のフレームより以前のフレームに関連する情報を利用することは不可能であるので、第 1 のフレームを 2 ビット平滑化技法により平滑化することはできない。現在のフレームが第 1 のフレームである場合 (ステップ 2 6 0 1 の Yes オプション)、制御流れはステップ 2 6 1 2 で終了する。これにより、全アルゴリズムは平滑化アルゴリズムのステップを迂回し、ステップ 2 5 0 5 へ続く。

【 0 2 4 3 】

50

現在のフレームがシーケンス中の第1のフレームではない場合には(ステップ2601のNoオプション)、制御流れはステップ2602へ進み、現在のコードブロック中の各係数が考慮されるループに入る。

【0244】

ループ2602の中の第1のステップであるステップ2603で、メモリからフレーム(f-1)の対応する係数の2つの格納ビット($a_{L(f-1)}$ 及び $a_{L(f-1)-1}$)が検索される。図24に示す例では、ステップ2603で検索される2つのビットはビット2407及び2408、すなわち、先行するフレームの対応する係数の2つの最下位ビットである。

【0245】

ステップ2603はフレーム(f-1)の対応するコードブロックの切捨てポイントTを更に検索する。この先行する切捨てポイントT(f-1)は、ステップ2508で説明したようにメモリに格納されている。切捨てビットプレーンL(f-1)は次の式を適用することにより得られるであろう。

$$L(f-1) = 2 + \text{floor}[(T(f-1) + 2)/3]$$

(21)

式中、floorはどの実数もそれより小さく、最も近い整数により近似されるような演算子である。

【0246】

次に、ステップ2604で、現在の係数 C^f の2進数字が検索される。次に、ステップ2605は現在の係数が平滑化を要求するか否かを判定する。

【0247】

ステップ2605で使用するのに適する試験は次の通りである。

$$\begin{aligned} & (Q[C^f, L(f-1)] > 0) \parallel (Q[C^f, L(f-1)] == 0 \\ & \& Q[C^f, L(f-1)+1] != 0 \\ & \& a_{L(f-1)}^{f-1} == 1) \end{aligned}$$

(22)

式中、演算子 \parallel は論理ORであり、 $==$ は双方のオペランドが同じ値を有する場合にTRUEを戻す演算子であり、 $\&\&$ は論理ANDであり、 $!=$ (すなわち、等しくない)はオペランドが互いに異なる場合にTRUEを戻す演算子である。

【0248】

論理ORの左側のオペランドは、ビットプレーンL(f-1)で切り捨てられた現在の係数のマグニチュードを検査する。これは次の加算に相当する。

$$Q[C^f, L(f-1)] = \sum_{p=1}^{L(f-1)} a_p 2^{-p}$$

(23)

式(23)の和が0より大きい場合、それは、ビットプレーンL(f-1)で切り捨てられた現在の係数の少なくとも1つの2進数字が1であることを示唆する。図24の例では、論理ORの左側の表現はビット2412~2414のうちの少なくとも1つが0でないかを試験する。式(22)の表現がTRUEである場合、それは現在の係数がビットストリームに符号化されようであることを示唆し、従って、現在の係数に2ビット平滑化を適用することが適切である。

【0249】

10

20

30

40

50

式(22)における論理ORの右側の表現は、切捨てビットプレーンが先行するフレームの切捨てビットプレーンと同じままであるときに現在の係数が符号化されないと思われる場合にTRUEである。しかし、論理ORの右側の表現は、現在の係数に2ビット平滑化技法を適用することを望ましくする特定のビットの集合体を識別する。

【0250】

まず、表現は式(23)の和が0に等しいか否かを検査する。これは、現在の係数がビットプレーン $L(f-1)$ で切り捨てられる場合、現在の係数の残る全ての2進数字が0であることを示唆している。図24の例において、この条件はビット2412~2414の全てが0に等しい場合に満たされるであろう。これが当てはまるとき、現在の係数は符号化されないと思われる。

【0251】

表現の第2の項は、現在の係数の切捨て形態に次に上位のビットが追加されるときに、現在の係数の有意性を見る。図24の例では、考慮されるこの追加ビットはビット2415である。第2の表現は、以下に示す和を含む。

$$Q[C^f, L(f-1)+1] = \sum_{p=1}^{L(f-1)+1} a_p 2^{-p}$$

(24)

図24を参照すると、式(22)において論理ORの右側の表現により識別される特定のビットの集合体は、ビット2408が「1」であり、ビット2415が「1」であり且つビット2412~2414は全て「0」である場合に現れる。

【0252】

現在の係数が平滑化を要求しない場合(ステップ2605のNoオプション)、制御流れは平滑化ステップ2606及び2607を迂回して、ステップ2610へ進む。しかし、平滑化が要求される場合には(ステップ2605のYesオプション)、制御流れはステップ2606へ進み、現在の係数のマグニチュードが平滑化される。

【0253】

ステップ2606の平滑化は式(20)の適用により実行される。あるいは、平滑化ステップ2606に対して次のルックアップテーブルを使用することが可能である。

表5：現在の係数を修正するためのルックアップテーブル

$a_{L(f-1)}^f$	$a_{L(f-1)-1}^f$	$a_{L(f-1)-1}^{f-1}$	L(f-1)切捨てビットプレーンで現在の係数に追加される
0	0	0	+1
0	0	1	-1
0	1	0	-1
0	1	1	+1
1	0	0	-1
1	0	1	+1
1	1	0	+1
1	1	1	-1

表5において、初めの3つの列で指定される条件が満たされるとき、第4の列に示される量が現在の係数のL(f-1)切捨てビットプレーンに追加される。表5は、 $a_{L(f-1)}^f$ と $a_{L(f-1)-1}^{f-1}$ が異なる場合に限って適用される。式(20)からわかるように、切捨てビットプレーンL(f-1)におけるビットがフレームfとフレーム(f-1)とで同じで

10

20

30

40

50

あるとき、現在の係数は不変である。

【 0 2 5 4 】

次のステップ 2 6 0 7 では、ウェーブレット係数の符号が復元される。

【 0 2 5 5 】

次に、ステップ 2 6 1 0 で、プロセッサ 2 7 0 5 は現在の係数がコードブロックにおける最後の係数であるか否かを検査する。コードブロックに係数がまだ残っている場合（ステップ 2 6 1 0 の No オプション）、プロセス流れはステップ 2 6 0 3 に戻り、次の係数を処理する。しかし、現在の係数がコードブロックの最後の係数である場合には（ステップ 2 6 1 0 の Yes オプション）、ステップ 2 6 1 1 で 2 ビットウェーブレット係数平滑化は終了し、全アルゴリズムは図 2 5 のステップ 2 5 0 5 へ進む。

10

【 0 2 5 6 】

ステップ 2 6 0 5 に他の条件を追加することができる。例えば、下記の条件を試験しても良い。

$$L(f - 1) == L(f - 2)$$

この条件は、コードブロックの切捨てビットプレーンがフレーム（ $f - 2$ ）からフレーム（ $f - 1$ ）まで変化しなかったことを保証するために使用される。これにより、フレーム（ $f - 1$ ）から現在のフレーム f までに切捨てビットプレーンが変化しなかった確率がより高くなる。この条件を使用するために、フレーム（ $f - 2$ ）の各コードブロックの切り捨てポイントを格納することができるであろう。あるいは、変化が起こったか否かを示すフラグを格納することも可能であろう。

20

【 0 2 5 7 】

更に、いくつかの望ましくないアーティファクトを回避するために、ステップ 2 6 0 5 で条件

$$Q[C^f, L(f - 1)] == 0$$

及び

$$Q[C^f, L(f - 1)] \neq \sum_{j=1}^L 2^{-j}$$

30

を使用することができる。

【 0 2 5 8 】

現在の係数の 2 つのビットを格納するために必要とされるメモリを更に縮小することが可能である。ステップ 2 5 0 8 では、4 種類の 2 ビット語、すなわち、（0, 0）、（0, 1）、（1, 0）及び（1, 1）のみが格納される。関連する適用状況において、発明者は格納される 2 ビット語は 80 % を越える割合で（0, 0）であることを認めた。従って、単一のビット 0 を使用して（0, 0）語を格納し且つ他の語を 1 0 1, 1 1 0 及び 1 1 1 としてそれぞれ格納することにより、ウェーブレット係数ごとに平均でメモリを更に 1.4 ビットまで縮小することができる。他の技法を使用して、この数字を更に減少させることは可能であろう。

40

【 0 2 5 9 】

2 ビットウェーブレット係数平滑化技法は、1 つのフレームの全ての成分に適用されるのが好ましい。それでも尚、いくつかの実験によれば、Y C b C r 色空間を使用するとき、Y 成分のみの平滑化がフリッカによって起こる視覚的アーティファクトを著しく減少させるのに十分であったことがわかっている。Y 成分のみを平滑化した場合、その結果、必要とされるメモリをウェーブレット係数当たり 1 ビット未満まで減少させることが可能であろう（例えば、4 : 2 : 0 Y C b C r 色空間を使用してウェーブレット係数当たり 0.95 ビット）。

50

【0260】

2ビットウェーブレット係数平滑化技法は1つのフレームの1つの成分の全てのコードブロックに適用されるのが好ましい。しかし、DWT分解能レベルのレベル1、2及び3にのみ平滑化技法を適用することが可能である。これにより、フリッカアーティファクトを著しく減少させつつ、ビットを格納するために要求されるメモリは更に縮小されるであろう。

【0261】

方法の1つの実現形態においては、係数を切り捨てるために使用される切捨てビットプレーンLはセンサ雑音の関数として選択されても良い。

【0262】

先に概要を述べた通り、現在のフレームfにおけるウェーブレット係数の修正は先行するフレーム(f-1)からのウェーブレット係数当たり2つのビットに基づいている。尚、相関係数Mdを計算する場合、最終フレーム(f-1)のウェーブレット係数も必要とされることに注意する。相関係数は、モーションに起因する対応するコードブロックの変化を示すために使用される。ウェーブレット係数当たりわずか2ビットの精度で計算された場合、相関係数はモーションの変化を示す。従って、図26の方法により現在の係数をフィルタリングするときに使用される2つのビットで、十分に相関係数を計算できるのである。

【0263】

6.0 産業上の適用可能性

以上の説明から、説明された構成は映像及びコンピュータの産業分野に適用可能であることが明白である。

【0264】

以上、本発明のいくつかの実施例のみを説明したが、それらの実施例は単なる例であって、限定的な意味を持たず、本発明の趣旨から逸脱せずに変形及び/又は変更を実施することができる。例えば、入力画像に離散的ウェーブレット変換(DWT)を適用する場合を例として方法を説明したが、ここで説明した方法は、離散的コサイン変換(DCT)などの異なる変換が入力画像に対して適用される場合にも使用されることが可能である。

【図面の簡単な説明】

【0265】

【図1】従来のJPEG2000規格に従った元の画像成分の対応する単一レベルDWT画像及び2レベルDWT画像への変換を示す図。

【図2】従来のJPEG2000規格に従ったサブバンドの複数のコードブロックへのタイリングを示す図。

【図3】従来のJPEG2000規格に従ったコードブロックの複数のブロックレイヤへの符号化を示す図。

【図4】従来のJPEG2000規格に従ったコードブロックにおけるコーディングパスのストライプ指向走査を示す図。

【図5】入力画像をJPEG2000フォーマットに圧縮する方法のフローチャート。

【図6】図5の方法で使用するためのPCRD-optアルゴリズムのフローチャート。

【図7】本発明の開示に従ってビデオ画像を圧縮する第1の構成のフローチャート。

【図8】切捨てポイントと終了ポイントの相違を示す概略図。

【図9】図7の方法を適用するときにフレーム間情報を格納する際に使用するためのメモリ構造を示す概略ブロック線図。

【図10】図7の静的ブロック切捨て安定化の方法のフローチャート。

【図11】図10の方法においてモーション検出パラメータを計算する方法のフローチャート。

【図12】図10の方法において図11のモーション検出パラメータがどのように使用されるかを示すフローチャート。

【図13】図7の方法で使用するためのフレーム利用成分重み付けの第1の方法のフロー

10

20

30

40

50

チャート。

【図14】図7の方法で使用するためのフレーム利用成分重み付けの別の方法のフローチャート。

【図15】図7の方法で使用するための符号化方式のフローチャート。

【図16】図15の方法で使用するための終了条件のフローチャート。

【図17】図15の方法で使用するための符号化後トリミングの方法のフローチャート。

【図18】本発明の開示に従ってビデオ画像を圧縮する第2の構成のフローチャート。

【図19】図18の第2の構成と共に使用するための符号化方式のフローチャート。

【図20(a)】図19の方法で使用するための終了条件のフローチャート。

【図20(b)】図19の方法で使用するための終了条件のフローチャート。

【図21】図20(a)及び図20(b)の方法で使用するための、次のコーディングパスの速度 ひずみ傾きを予測する方法のフローチャート。

【図22】コーディングパスの数と、速度 - ひずみ傾きとの関係の一例を示すグラフ。

【図23】図22のグラフの凸閉包を示す図。

【図24】連続するフレームにおける2つの対応する係数の一例を示す図。

【図25】フリッカ減少のために2ビット平滑化を使用してビデオのフレームのシーケンスをモーションJPEG2000フォーマットに圧縮する方法のフローチャート。

【図26】図25の方法で使用するための係数平滑化ステップ2504のフローチャート

。

【図27】ここで説明する構成を実施できる汎用コンピュータの概略ブロック線図。

【符号の説明】

【0266】

2700... 汎用コンピュータシステム、2701... コンピュータモジュール、2702... キーボード、2703... マウス、2704... 相互接続バス、2705... プロセッサユニット、2706... メモリユニット、2707... オーディオビデオインタフェース、2708... インタフェース、2709... 記憶装置、2710... ハードディスクドライブ、2711... フロッピー(登録商標)ディスクドライブ、2712... CD-ROMドライブ、2713... I/Oインタフェース、2714... 表示装置、2715... プリンタ、2716... モデムトランシーバ装置、2720... 通信ネットワーク、2721... 電話回線

10

20

【 図 1 】

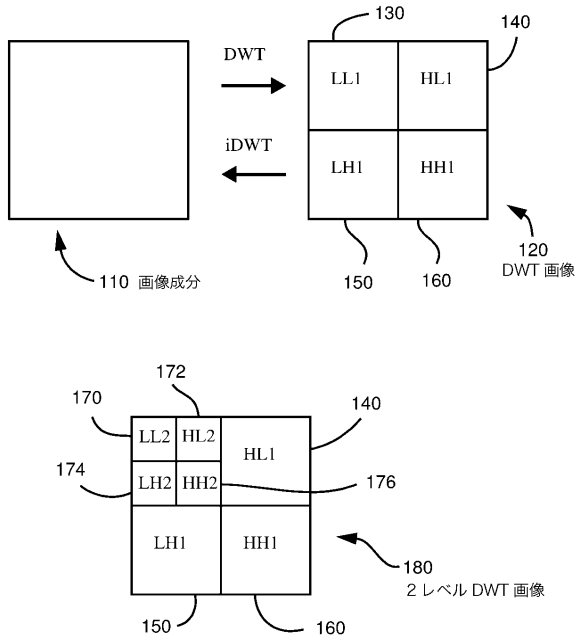


Fig. 1

【 図 2 】

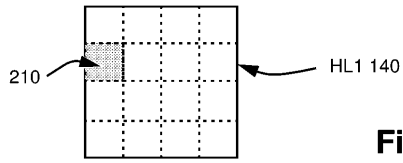


Fig. 2

【 図 3 】

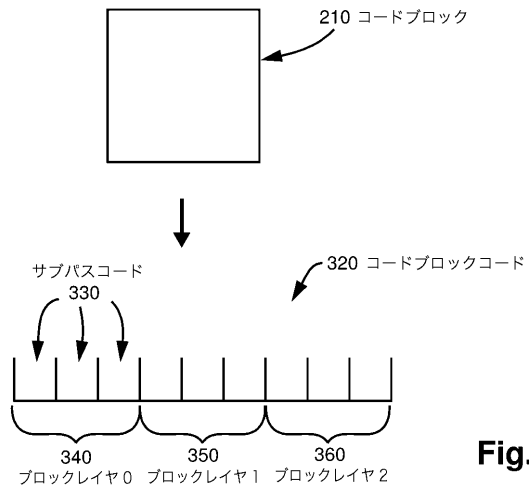


Fig. 3

【 図 4 】

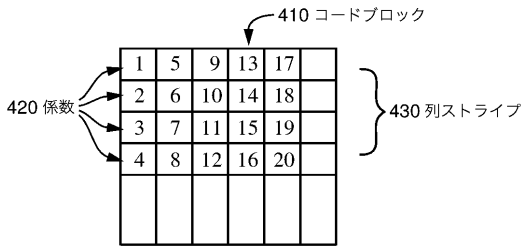


Fig 4

【 図 5 】

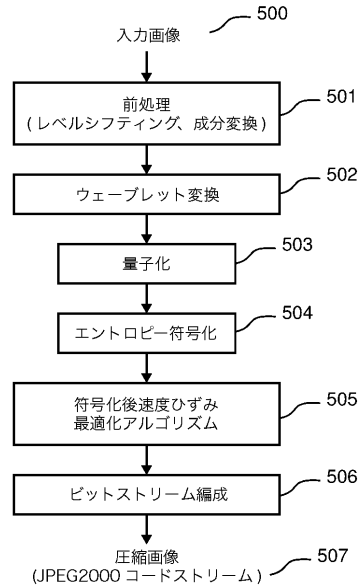


Fig. 5

【図6】

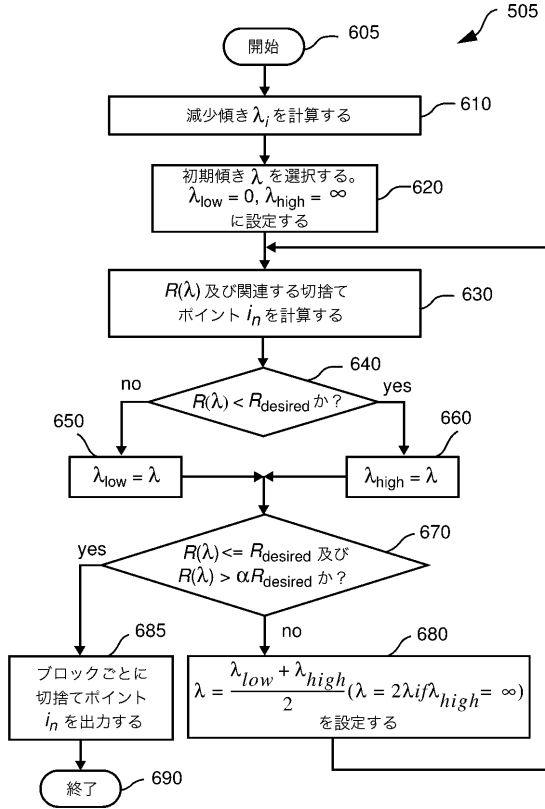


Fig. 6

【図7】

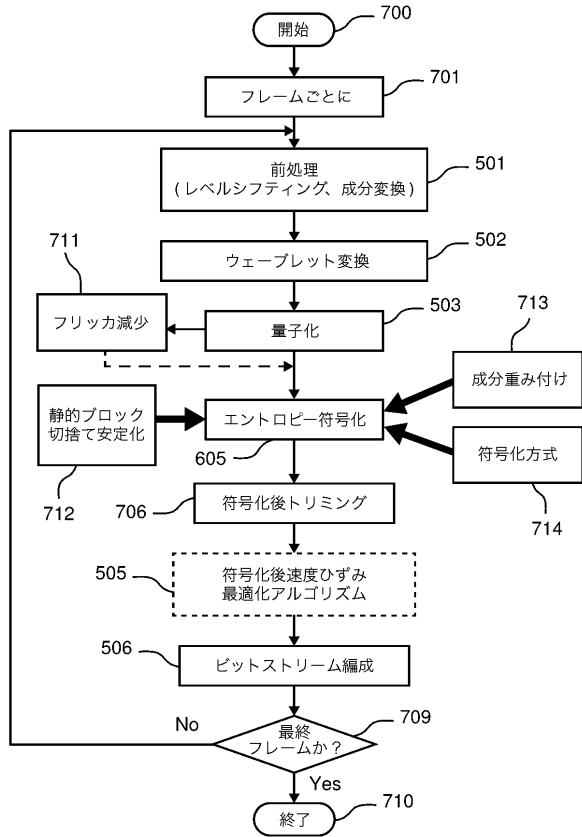


Fig. 7

【図8】

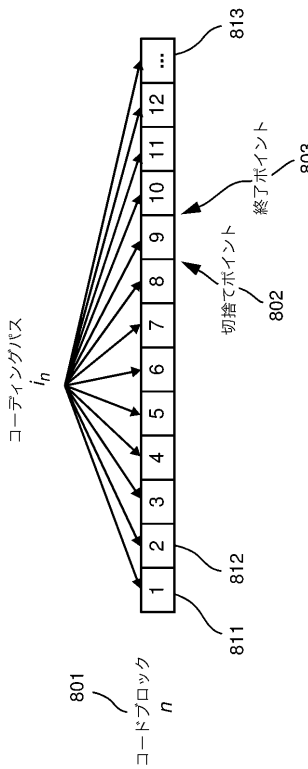


Fig. 8

【図9】

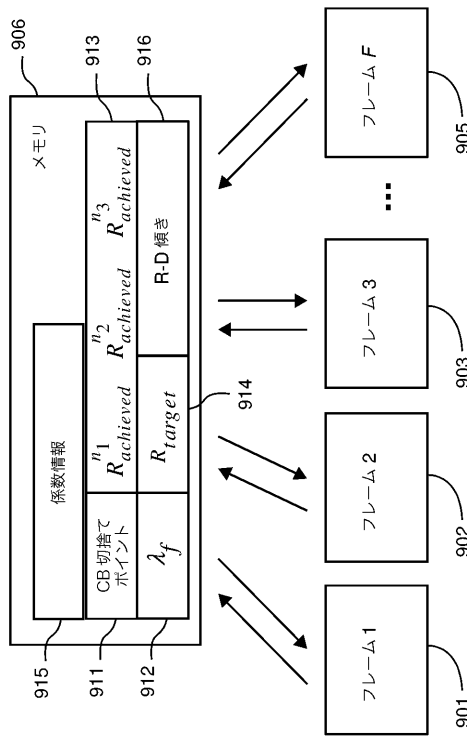


Fig. 9

【図10】

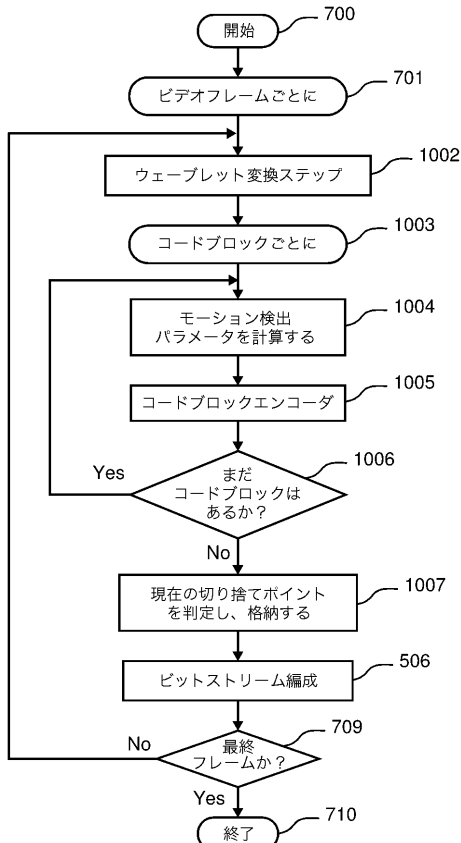


Fig. 10

【図11】

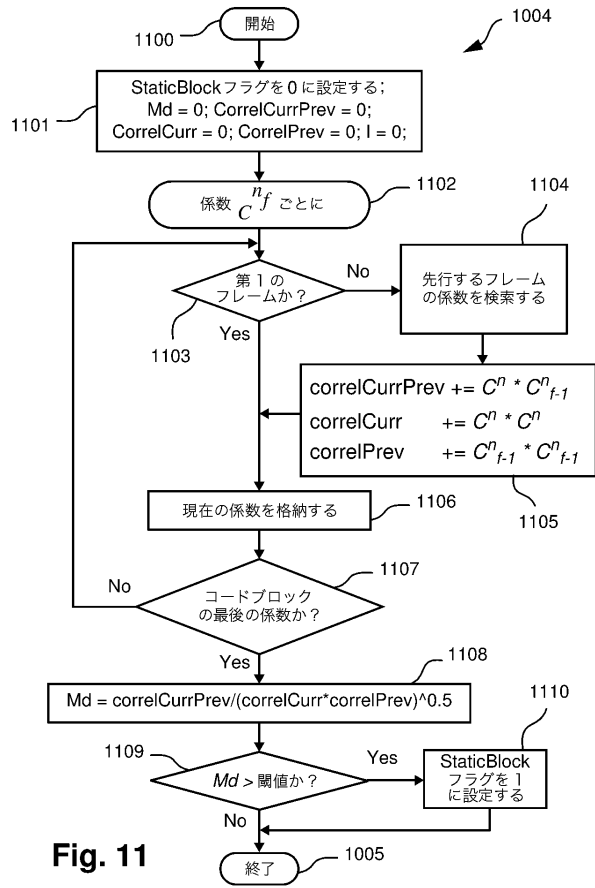


Fig. 11

【図12】

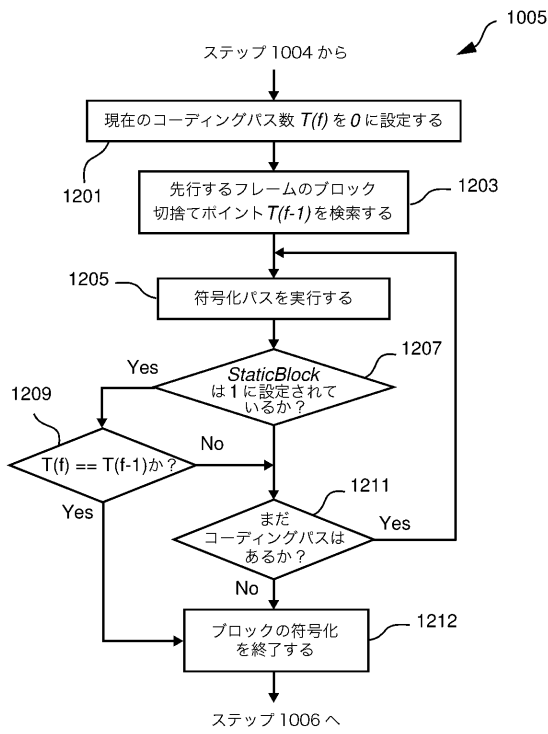


Fig. 12

【図13】

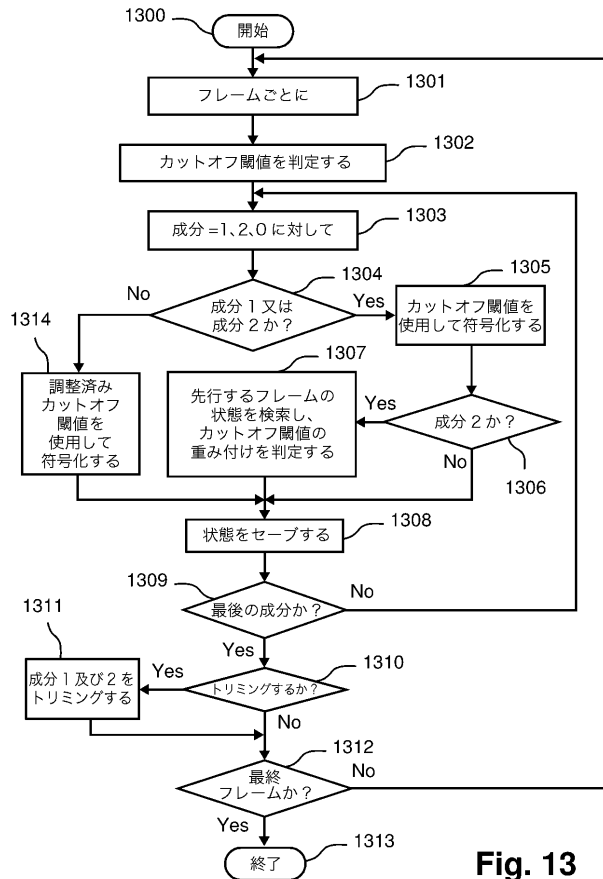


Fig. 13

【図14】

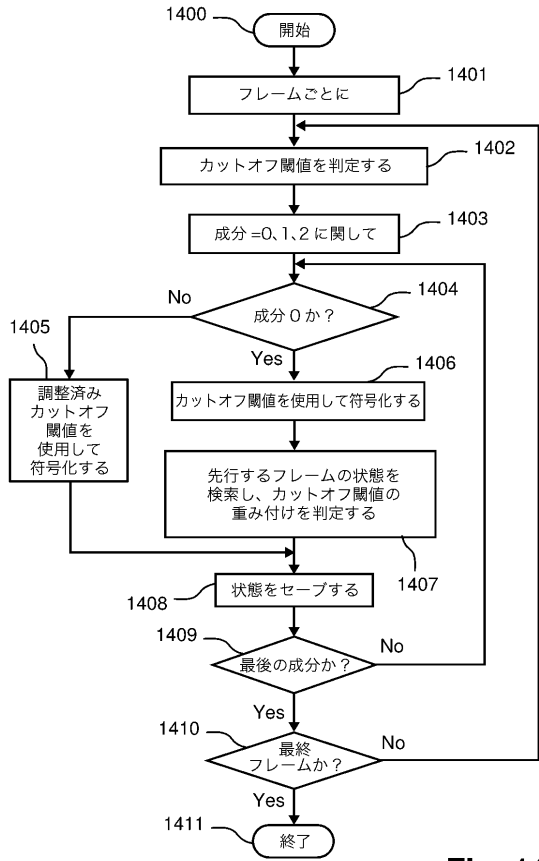


Fig. 14

【図15】

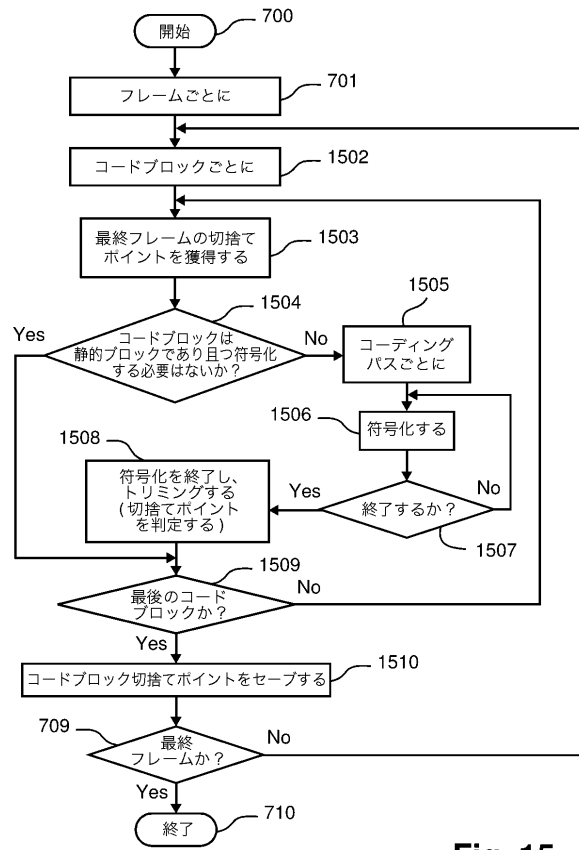


Fig. 15

【図16】

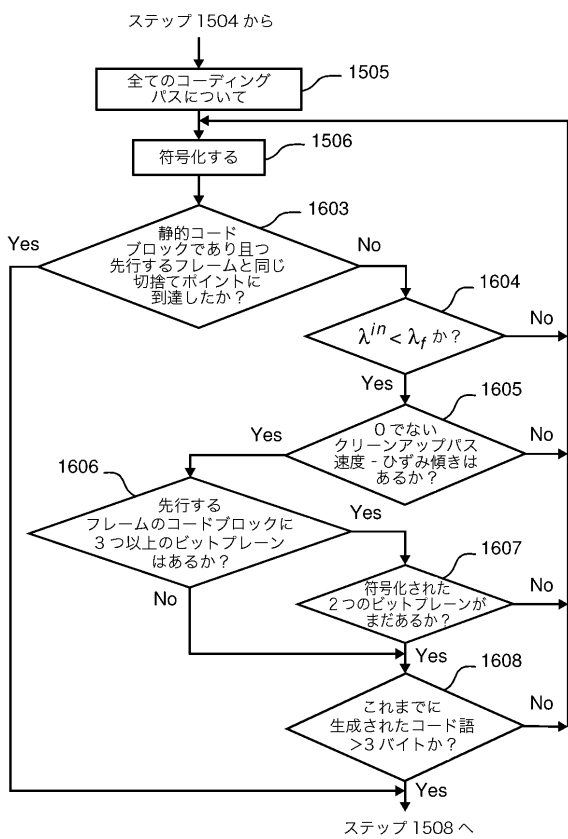


Fig. 16

【図17】

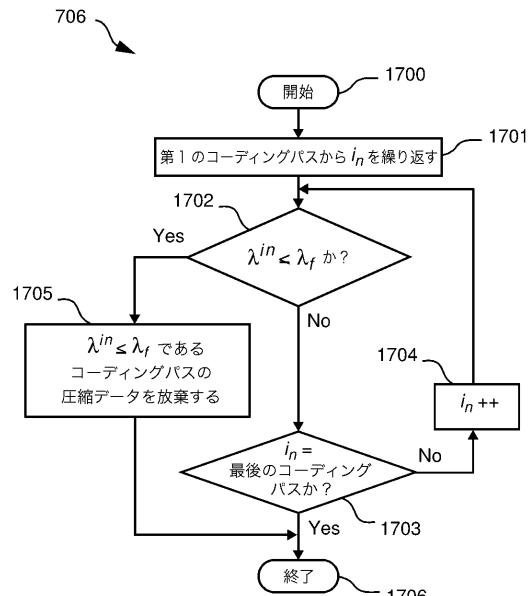


Fig. 17

【図18】

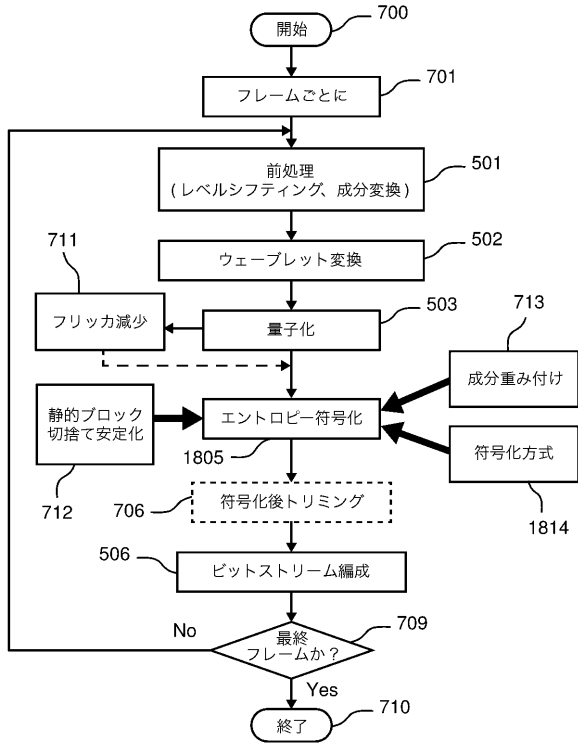


Fig. 18

【図19】

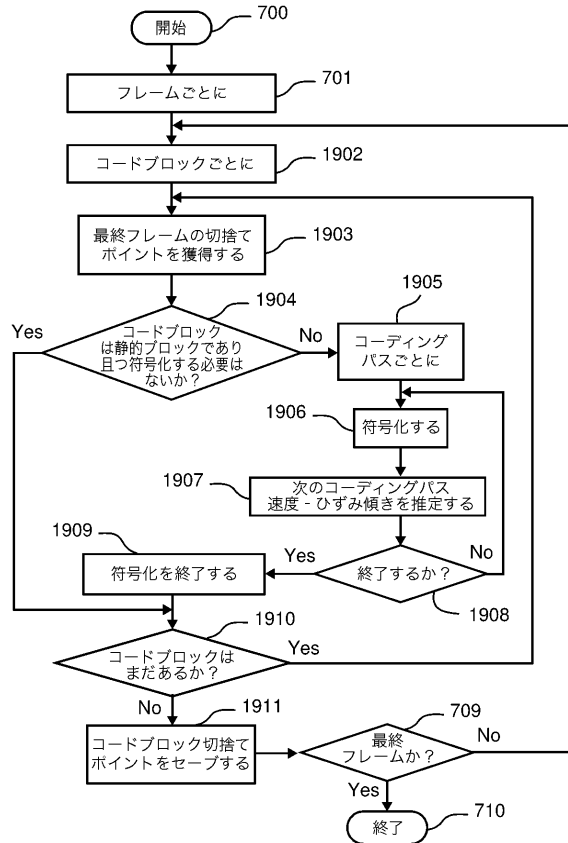


Fig. 19

【図20(a)】

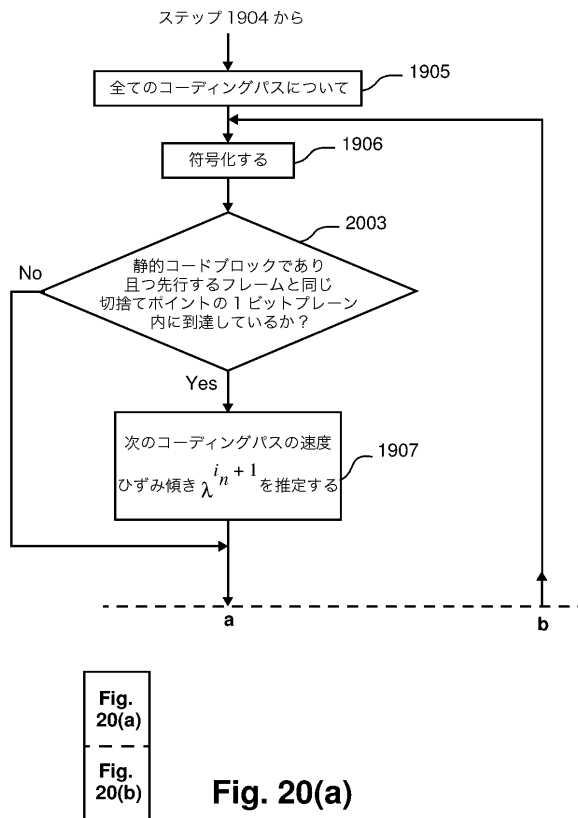


Fig. 20(a)

【図20(b)】

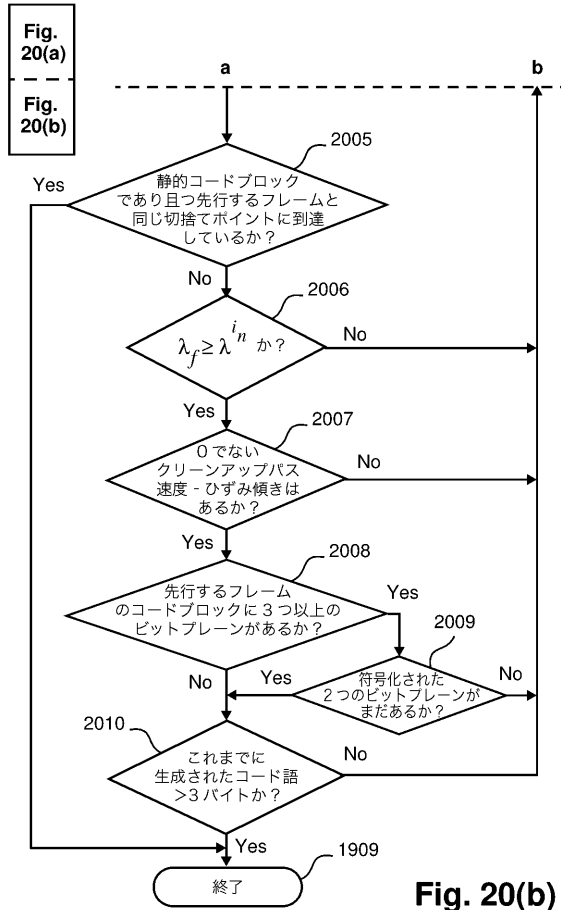


Fig. 20(b)

【図 2 1】

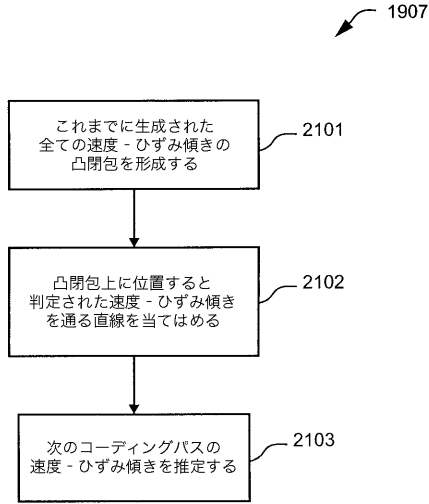


Fig. 21

【図 2 2】

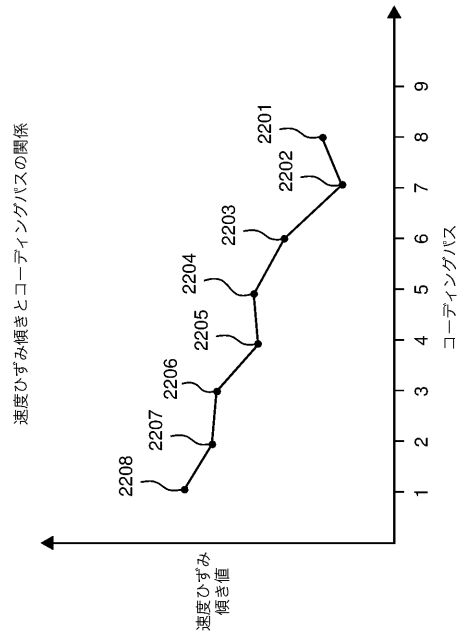


Fig. 22

【図 2 3】

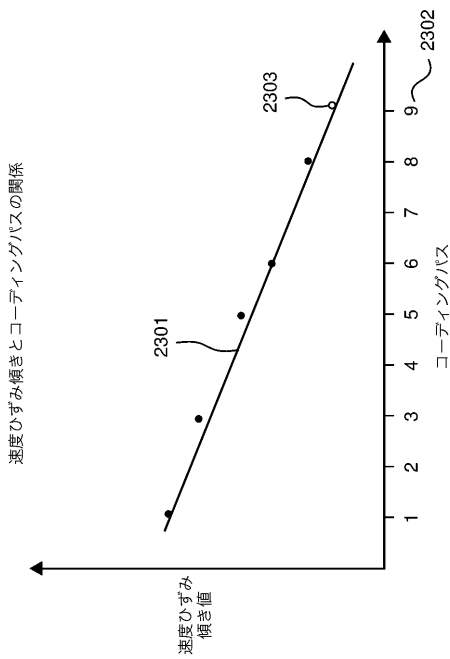


Fig. 23

【図 2 4】

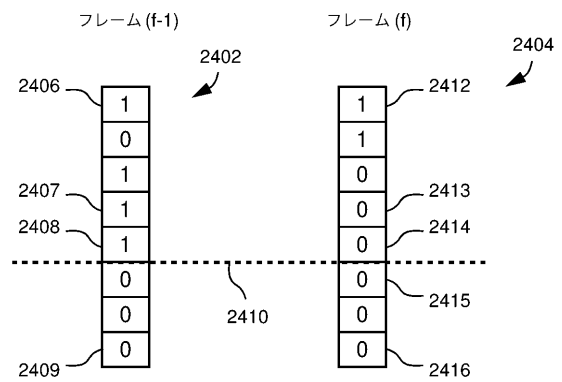


Fig. 24

【図 25】

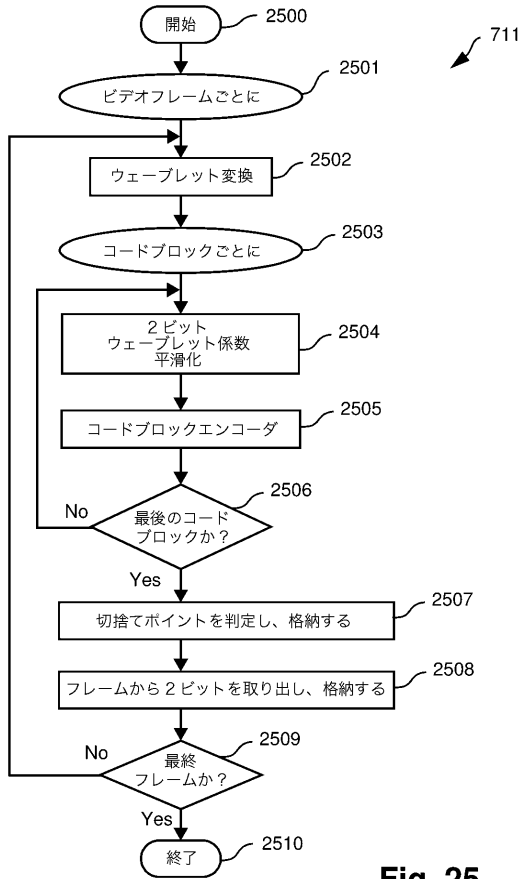


Fig. 25

【図 26】

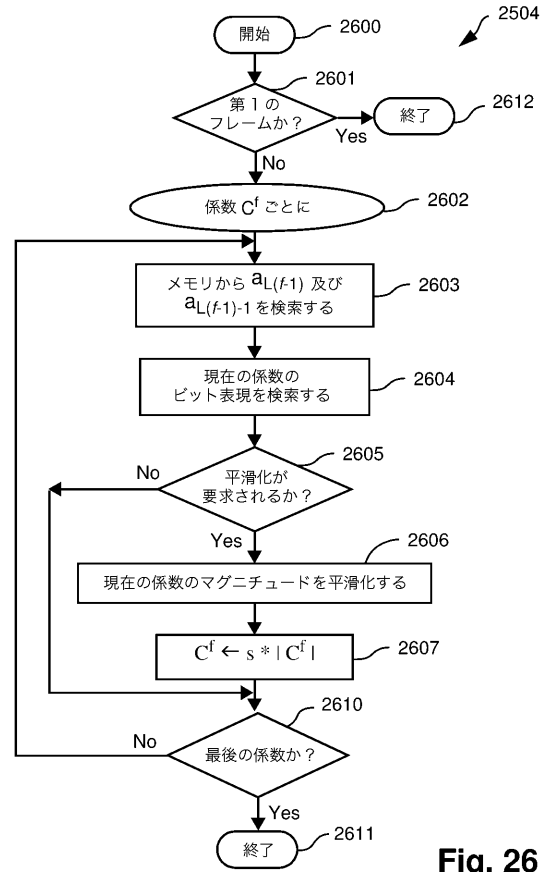


Fig. 26

【図 27】

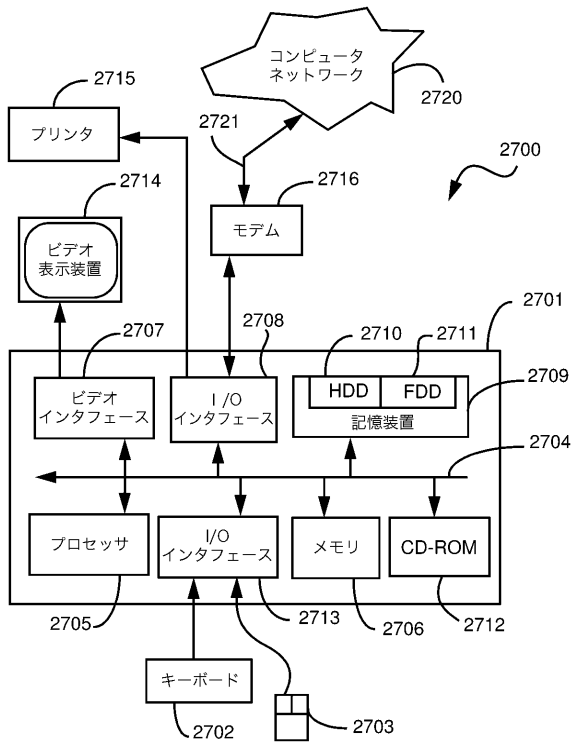


Fig. 27

フロントページの続き

(31)優先権主張番号 2003906091

(32)優先日 平成15年10月31日(2003.10.31)

(33)優先権主張国 オーストラリア(AU)

(72)発明者 アクセル ベッカー

オーストラリア国 2113 ニュー サウス ウェールズ州, ノース ライド, トーマス
ホルト ドライブ 1, キヤノン インフォメーション システムズ リサーチ オーストラ
リア プロプライエタリー リミテッド 内

(72)発明者 ウォーエイ チャン

オーストラリア国 2113 ニュー サウス ウェールズ州, ノース ライド, トーマス
ホルト ドライブ 1, キヤノン インフォメーション システムズ リサーチ オーストラ
リア プロプライエタリー リミテッド 内

(72)発明者 デイビッド ドミニク エム. プールアン

オーストラリア国 2113 ニュー サウス ウェールズ州, ノース ライド, トーマス
ホルト ドライブ 1, キヤノン インフォメーション システムズ リサーチ オーストラ
リア プロプライエタリー リミテッド 内

(72)発明者 ジェームス フィリップ アンドリュー

オーストラリア国 2113 ニュー サウス ウェールズ州, ノース ライド, トーマス
ホルト ドライブ 1, キヤノン インフォメーション システムズ リサーチ オーストラ
リア プロプライエタリー リミテッド 内

審査官 坂東 大五郎

(56)参考文献 特開2002-359850(JP,A)

国際公開第98/034398(WO,A1)

特開2000-188552(JP,A)

Y. M. Yeung, Oscar C. Au, Andy Chang, Successive bit-plane rate allocation technique for JPEG2000 image coding, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. (ICASSP '03), 2003年 4月 6日, Vol. 3, p. III-261 - III-264

Te-Hao Chang, Li-Lin Chen, Chung-Jr Lian, Hong-Hui Chen, and Liang-Gee Chen, Computation reduction technique for lossy JPEG2000 encoding through EBCOT Tier-2 feedback processing, Proceedings of International Conference on Image Processing. 2002., 2002年, Vol. 3, p. III-85 - III-88

(58)調査した分野(Int.Cl., DB名)

H04N 7/24 - 7/68

H04N 1/41 - 1/419