

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5236947号
(P5236947)

(45) 発行日 平成25年7月17日 (2013. 7. 17)

(24) 登録日 平成25年4月5日 (2013. 4. 5)

(51) Int. Cl.

H04N 7/26 (2006.01)

F I

H04N 7/13

Z

請求項の数 1 (全 30 頁)

(21) 出願番号 特願2007-545501 (P2007-545501)
 (86) (22) 出願日 平成17年11月21日 (2005. 11. 21)
 (65) 公表番号 特表2008-523720 (P2008-523720A)
 (43) 公表日 平成20年7月3日 (2008. 7. 3)
 (86) 国際出願番号 PCT/US2005/042384
 (87) 国際公開番号 W02006/065477
 (87) 国際公開日 平成18年6月22日 (2006. 6. 22)
 審査請求日 平成20年11月20日 (2008. 11. 20)
 (31) 優先権主張番号 60/635, 116
 (32) 優先日 平成16年12月10日 (2004. 12. 10)
 (33) 優先権主張国 米国 (US)
 (31) 優先権主張番号 11/269, 428
 (32) 優先日 平成17年11月7日 (2005. 11. 7)
 (33) 優先権主張国 米国 (US)

(73) 特許権者 504290273
 タット・システムズ・インコーポレイテッド
 アメリカ合衆国、オレゴン州 97035
 、レイク オスウィーゴ、エスタブリュ
 ー メドウズ ロード 6000、スイ
 ト 200
 6000 S. W. Meadows R
 oad, Suite 200 Lake
 Oswego, Oregon 970
 35, U. S. A.

(74) 代理人 110001209
 特許業務法人山口国際特許事務所

前置審査

最終頁に続く

(54) 【発明の名称】 マルチプロセッサアーキテクチャ並びにピクチャベースの先読みウィンドウを具備したデジ
 タルビデオエンコーダのためのパラレルレートコントロール

(57) 【特許請求の範囲】

【請求項 1】

複数のデュアルパスマルチプロセッサシングるチャンネルエンコーダを有するマルチ
 チャンネルビデオエンコーダであって、各シングるチャンネルエンコーダが共通のビデオク
 ロックを共有し第1のパスエンコーダと第2のパスエンコーダを有するものを操作する方
 法であって、

複数のデュアルパスマルチプロセッサシングるチャンネルエンコーダのそれぞれが、
 複数の一連のピクチャと組み込まれたビデオクロックを含む、単独の非エンコードブ
 ログラムビデオストリームを受信し、

前記単独の非エンコードプログラムビデオストリームのコピーを生成し、

前記単独の非エンコードプログラムビデオストリームを保管し、

前記コピーの組み込まれたビデオクロックを修正してそれを共通のビデオクロックに同
 期させ、

前記コピーに対して第1のパスエンコードを実施して、該コピーの複雑系統計を行う工
 程と、

複雑系統計を行うことによって、それぞれの前記単独の非エンコードプログラムビデオ
 ストリームのピクチャの好適なターゲットサイズを判定する工程と、

前記複数のデュアルパスマルチプロセッサシングるチャンネルエンコーダのそれぞれが
 、

入力として前記好適なターゲットサイズを用いて、前記単独の非エンコードプログラム

10

20

ビデオストリームに対し第2のパスエンコードを実施して、エンコードされた単独のプログラムビデオストリームを生成し、

複数のエンコードされた単独のプログラムビデオストリームを互いに積算してマルチプログラムトランスポートストリームとし、

そのマルチプログラムトランスポートストリームを送信する工程を具備する方法。

【発明の詳細な説明】

【発明の背景】

【0001】

発明の背景

規格H.264が以下に特別に参照されるけれども、記載された思想や原理は特定のビデオコーディング規格に依存するものではなく、規格MPEG-2、規格ISO MPEG-4 part 10 AVC、又は、最近の規格SMPTE VC-1にとって等しく有効である。

【0002】

映像圧縮部門においては、特にテレビジョン放送やビデオ会議のようなリアルタイム用途において、ビデオエンコードの効率を永続的に改善する絶え間ない要求が存在する。最近の規格ITU-T H.264は、この効率改善要求に合致するように設計されているが、それに応じてアルゴリズムの複雑さが増す犠牲を伴う。例えば、H.264は同じ画質を得るためにMPEG-2ビデオストリームのビットレートの約半分を必要とするが、一方、H.264エンコーダの実装の複雑さはMPEG-2エンコーダに比べて桁違いに大きくなる。

【0003】

典型的な例では、一連のピクチャからなる非圧縮ビデオストリームがビデオエンコーダによって受信され、そして、ビデオエンコーダはビデオシーケンス内の各ピクチャのエンコードされたバージョンを生成し、それによって、非圧縮ビデオストリームのエンコードされたバージョンを生成する。そして、エンコードされたビデオストリームは固定ビットレート(CBR)チャンネル上でビデオデコーダに送信され、そして、ビデオデコーダは、エンコードされたビデオストリームをデコードし、それによって、元の非圧縮ビデオストリームから理論上見た目に区別の付かない非圧縮ビデオストリームを生成する。

【0004】

エンコーダがビデオシーケンスのピクチャの圧縮されたバージョンを生成するために用いるビットが増えれば増えるほど、そのピクチャの圧縮されたバージョンをCBRチャンネル上で送信するにはより時間がかかってしまう。デコーダにおいて、エンコードされたピクチャが受信されたときには、それらのピクチャはデコーダのバッファにロードされてデコードのために待機する。エンコードされたピクチャの各ビットは、到着した順に順次デコーダのバッファにロードされるが、ピクチャのデコード時点において、全てのビットがバッファから同時に取り除かれる。

【0005】

上記の単純化されたモデルにおいて、特定の期間に亘って、デコーダのバッファは、変動する数のピクチャに対応して一定数のビットを受け取る。同じ期間に亘って、デコーダは、変動する数のビットに対応してデコーダのバッファから一定数のピクチャを取り除く。その期間内においてエンコーダが比較的に大規模なピクチャを数多く送信している場合には、その期間内において受信されるピクチャの数は比較的に小さいものとなる。デコーダのバッファが新たなピクチャを受け取るよりも早いレートでピクチャを取り除くことがあるので、このことはデコーダのバッファを空にするか、又は、アンダーフローにしてしまう。反対に、その期間内においてエンコーダのバッファが比較的に小規模のピクチャを数多く受信している場合には、その期間内において受信されるピクチャの数は比較的に大きいものとなる。デコーダのバッファがピクチャを取り除くよりも早いレートで新たなピクチャを受け取ることがあるので、このことはデコーダのバッファをオーバーフローにしてしまう。アンダーフローもオーバーフローもデコーダによって生

10

20

30

40

50

成された非圧縮ビデオストリームを崩壊させることがあり、したがって、アンダーフローもオーバーフローも生じることがないようにすべきである。よって、ビデオエンコーダがエンコードされたビデオストリームを生成している間に、ビデオエンコーダが下流のデコーダのバッファのフルネスを注視していることが重要である。しかし、デコーダはエンコーダと通信することができず、したがって、デコーダのバッファの実際のフルネスはエンコードにとって利用できない。このために、H. 264のような圧縮ビデオ規格は、仮想参照デコーダ(HRD)を規定し、ビデオエンコーダは、一般的に仮想バッファと呼ばれる、HRDの符号化されたピクチャバッファの数学的モデルを保持している。仮想バッファが決してアンダーフロー/オーバーフローしないとしたら、デコーダのバッファは反対に決してオーバーフロー/アンダーフローをするものではない。そして、エンコーダはエンコードされたビデオストリームを調整して、ビデオストリームのピクチャのエンコードされたバージョンのサイズを合わせてセーフレベルで仮想バッファのフルネスを維持することによって、下流のデコーダバッファのアンダーフロー又はオーバーフローを回避することができる。

【0006】

より新しく、計算上より複雑な、映像符号化規格の計算要件に合致するために、並行処理を用いるマルチプロセッサ設計が実装されている。例えば、ある一つのプロセッサがIピクチャとPピクチャをエンコードするために指定され、そして、残りのプロセッサがBピクチャをエンコードするために指定されている。I又はPピクチャをエンコードするためよりもBピクチャをエンコードするにはより多くの計算サイクルが必要であり、一般的に所定のビデオストリームにはIピクチャとPピクチャに比べて相当多くのBピクチャが存在するので、より多くのプロセッサがBピクチャのエンコードに使用される。話を単純にするために、この例において、他はH. 264規格が適用されるけれども、基準のピクチャとしてBピクチャが用いられないものと仮定する。以下のgroup of picture (GOP) 構造記述において、下付の数字はデコード(エンコード)順を表している。すなわち、

$I_0 B_2 B_3 P_1 B_5 B_6 P_4 B_8 B_9 P_7 B_{11} B_{12} P_{10} B_{14} B_{15} I_{13} \dots$

表1は、上記GOPの各ピクチャがマルチプロセッサエンコーダにおいてどのように配分されるのかを単純化された形で表示している。表1において、各タイムスロットuは、3つのピクチャのためのリアルタイムでの表示期間を表しており、入力がNTSCビデオであった場合、各タイムスロットは3003/30000秒間の期間を有する。

【0007】

【表1】

プロセッサ	U=1	u=2	u=3	u=4	u=5	u=6	u=7	u=8
1	I_0	P_1	P_4	P_7	P_{10}	I_{13}	P_{16}	P_{19}
2			B_2	B_2	B_8	B_8	B_{14}	B_{14}
3			B_3	B_3	B_9	B_9	B_{15}	B_{15}
4				B_5	B_5	B_{11}	B_{11}	B_{17}
5				B_6	B_6	B_{12}	B_{12}	B_{18}
利用可能な エンコード された ピクチャ				I_0 P_1	B_2 B_3 P_4	B_5 B_6 P_7	B_8 B_9 P_{10}	B_{11} B_{12} I_{13}

10

20

30

40

50

【 0 0 0 8 】

表 1 に示すように、プロセッサ 1 は、I ピクチャと P ピクチャを含む全ての基準ピクチャをエンコードするために用いられる。プロセッサ 2、3、4、5 は、B ピクチャをエンコードするために用いられる。このように、表示された例にとっては、処理パイプラインがはじめに充填された後で、少なくとも 2 つタイムスロットのパイプライン遅延が存在する。各 B ピクチャが 2 つのタイムスロットに割り当てられるが、I 又は P ピクチャはそれぞれ 1 つのタイムスロットに割り当てられることに特に言及する。全て必要な基準ピクチャのエンコードが完了するとすぐに、参照する B ピクチャのエンコードが並列的に始まる。例えば、ピクチャ P_4 のエンコードが終了するとすぐに、ピクチャ B_5 と B_6 のエンコードがそれぞれプロセッサ 4 と 5 において並列的に始まる。プロセッサ 1 は、その他のプロセッサに活動に関わらず、I 又は P ピクチャのエンコードを継続する。表 1 の最下行は、エンコードされた各ピクチャがエンコード順に利用可能なので、それらを示している。表 1 は簡略化された例である。実際は、非圧縮ピクチャの受信やピクチャのエンコードや再構成されたピクチャの送信や再構成されたピクチャの受信や必要な遅延などの、もっと多くの活動が考えられる。しかし、それらの詳細は本例にとって不可欠のものではないので、それらは省略される。

【 0 0 0 9 】

ビデオエンコーダの重要なタスクは、エンコードされたビデオストリームの所望の画質とビデオストリームが送信されているチャンネルのビットレート制限とデコーダのバッファのフルネスのセーフレベルを維持することとの間の適切なバランスを見いだすことである。エンコーダのレートコントロールアルゴリズムは、仮想バッファのフルネスと個々のピクチャの相対的な複雑さを用いて、ビデオストリームの各ピクチャに対する適切なビット割り当てを算出する。シーケンシャルエンコーディングにおいては、レートコントロールアルゴリズムが、後続のピクチャのエンコードが始まる前において各ピクチャのエンコードの完了時点で仮想バッファのフルネスをチェックする。パイプラインパラレルエンコーダの場合、仮想バッファのフルネスは、複数ピクチャの同時及び非シーケンシャルエンコーディングによって引き起こされたパイプライン遅延のためにレートコントロールアルゴリズムにとって必ずしも直ぐに利用されるものではない。パイプラインパラレルエンコーダの全てのプロセッサが互いに独立して作動するので、「後続のピクチャ」の概念は「ある一定の遅延の後に続くピクチャ」に置き換えねばならず、その遅延の間に、いくつかの別のピクチャがエンコードされて、仮想バッファのフルネスを変更する。例えば、レートコントロールアルゴリズムは、シーケンシャルなエンコード順に従って、ピクチャ B_6 のエンコードの完了後ピクチャ P_7 のエンコードの前に仮想バッファのフルネスをチェックする。ピクチャ B_6 と P_7 のエンコードが同時に始まりピクチャ P_7 のエンコードがピクチャ B_6 のエンコードの終了前に完了するので、このことは上記のパラレルエンコーダにとって不可能である。したがって、レートコントロールアルゴリズムは、現時点の値を単純にチェックするよりも、未来のある時点で仮想バッファのフルネスを予測する必要がある。レートコントロールアルゴリズムの条件を複雑にするパラレルエンコーダについての関連する問題は、充填ビットの潜在的な必要性である。エンコーダは、仮想バッファが空になるか又はそうなる危険性があるときに、エンコードされたビデオストリームに充填ビットを挿入する必要がある。シーケンシャルエンコーダのレートコントロールアルゴリズムは、ピクチャのエンコードが完了すると直ぐに、どれくらいの数のビットを充填するのかが正確に分かる。しかし、パラレルエンコーダのレートコントロールアルゴリズムは、前記のパイプライン遅延による仮想バッファの真のフルネスの正確な測定が利用不可能であるために、別の方法で必要な数の充填ビットを算出する必要がある。パラレルレートコントロールアルゴリズムの別の要件は、いくつかのピクチャに対するいくつかのビットレートターゲットを同時に決定する能力である。表 1 に示すように、例えば、タイムスロット 4 の始めにおいて、パラレルレートコントロールアルゴリズムはピクチャ P_7 と B_5 と B_6 のビットレートターゲットを決定する必要がある。

【 0 0 1 0 】

シングルパスエンコーディングに比べデュアルパスエンコーディングの方がより高度の符号化効率を提供することは画像圧縮部門において広く知られている。しかし、効率の改善は、デュアルパスアーキテクチャと共に二つのパイプラインエンコーダを用いることによる比較的に大きなコストに勝るものではないので、デュアルパスエンコーディングは必ずしも実際の解決にはつながらない。

【 0 0 1 1 】

図 1 に関連して、シーケンシャルデュアルパスエンコーダ 1 は、非エンコードビデオストリームを入力として受信し、エンコードされたシングルプログラムトランスポートストリーム (S P T S) を (用途に応じて、変動ビットレート (V B R) 又は固定ビットレート (C B R) の何れかで送信することが可能な) 出力として送信するが、第 1 パスエンコーダ 2 と第 2 パスエンコーダ 3 と保管及び遅延ユニット 4 を具備する。第 1 パスエンコーダ 2 は第 2 パスエンコーダ 3 と比較して割合単純である。例えば、第 1 パスエンコーダ 2 への入力、ダウンサンプリングされ、そして、第 2 パスエンコーダ 3 の入力に比較される。第 1 パスエンコーダ 2 と保管及び遅延ユニット 4 は、非エンコードビデオストリームを入力として受信する。保管及び遅延ユニット 4 は、ビデオストリームをバッファに保管する、一方、簡単な第 1 パスエンコーダ 2 はビデオストリームの各ピクチャの複雑さの情報を算出する。そして、ビデオストリームの各ピクチャと対応する複雑系統計は、第 2 パスエンコーダ 3 に送られる。第 2 パスエンコーダ 3 は、第 1 パスエンコーダ 2 によって生成されたその複雑さの情報をを用いて、入力ビデオストリームのエンコードされたバージョンを生成する。入力において、より複雑なエンコーダの代わりに簡単な第 1 パスエンコーダ 2 を用いることによって、実装コストが抑えられて複雑なシングルパスエンコーダのそれに近づく。しかし、第 1 と第 2 のパスエンコーダの実装の相違のために、第 1 パスエンコーダ 2 によって生成されたその複雑さの情報は、比較的に複雑な第 2 パスエンコーダ 3 にとって正に望ましい情報ではない。

【 0 0 1 2 】

この欠陥にもかかわらず、第 1 パスエンコーディングにおけるピクチャの複雑さ推定と第 2 パスエンコーディングにおけるピクチャの複雑さ推定の間には相関関係が存在する。多くの場合、第 1 パスエンコーダ 2 にとって比較的に複雑 / 単純であるピクチャ又は G O P は、また、第 2 パスエンコーダ 3 にとっても比較的に複雑 / 単純である。複雑系統計は、エラーに耐えられる状態で、一層各ピクチャとマクロブロック (M B) の間の重要な関係を示している。したがって、複雑なシングルパスエンコーディングと比較して、デュアルパスエンコーダは、若干高価な実装コストのみを有するも画像符号化効率において優れている。

【 0 0 1 3 】

発明の要約

本発明の第 1 の側面に関連して、ビデオシーケンスを構成する一連のピクチャ群のうちのピクチャのエンコードされたバージョンを生成するときを使用すべき好適な数のビットに対応するターゲットサイズを決定することによってマルチプロセッサビデオエンコーダを作動する方法であって、(イ)第 1 の時点で符号化されたピクチャのバッファの第 1 のフルネスの程度を算出する工程と、(ロ)前記第 1 のフルネスの程度に作用して、第 2 の時点で前記符号化されたピクチャのバッファの想定された第 2 のフルネスの程度に戻す工程と、(ハ)前記第 2 のフルネスの程度に作用して、前記ピクチャの当初のターゲットサイズに戻す工程とからなり、前記第 1 の時点が、前記符号化されたピクチャのバッファの正確なフルネスの程度が算出できるもっとも近接する時点に対応していると共に、前記第 2 の時点が前記第 1 の時点の後に生じる方法が提供される。

【 0 0 1 4 】

本発明の第 2 の側面に関連して、ビデオシーケンスのピクチャをエンコードするマルチプロセッサビデオエンコーダを生成する方法であって、(イ)前記ピクチャを受信する工程と、(ロ)第 1 の時点で符号化されたピクチャのバッファの第 1 のフルネスの程度を算出する工程と、(ハ)前記第 1 のフルネスの程度に作用して、第 2 の時点で前記符

号化されたピクチュアのバッファの第2のフルネスの程度に戻す工程と、(ニ)ある時間の間、その時間の前記ピクチュアを保管する工程と、(ホ)前記ピクチュアの複雑さの第1の程度を測定する工程と、(ヘ)前記ピクチュアの複雑さの第1の程度と前記第2のフルネスの程度に作用して、前記ピクチュアの好適なターゲットサイズに戻す工程と、(ト)工程(ニ)に引き続き、前記ピクチュアと前記好適なターゲットサイズを前記マルチプロセッサビデオエンコーダに提供する工程とからなり、前記第1の時点が、前記符号化されたピクチュアのバッファの正確なフルネスの程度が算出できるもっとも近接する時点に対応していると共に、前記第2の時点が前記第1の時点の後に生じる方法が提供される。

【0015】

本発明の第3の側面に関連して、一連のピクチュアのうち単独のピクチュアを構成する一連のマクロブロックのうちの所定のマクロブロックをエンコードする時に使用する量子化パラメータの値を決定する方法であって、(イ)第1の時点で符号化されたピクチュアのバッファの第1のフルネスを推定する工程と、(ロ)前記一連のマクロブロックのうち、前記所定のマクロブロックに先行するマクロブロックの複雑さの累積単位を決定する工程と、(ハ)前記第1のフルネスの程度と前記複雑さの累積単位とターゲットビット数に作用して、第2の時点で前記符号化されたピクチュアのバッファの第2のフルネスの程度に戻す工程と、(ニ)前記第2のフルネスの程度に作用して、前記量子化パラメータの値に戻す工程とからなり、前記ターゲットビット数が、前記単一のピクチュアのエンコードされたバージョンを発生するときに前記一連のマクロブロックを分割するビット数に対応していると共に、前記第2の時点が前記第1の時点の後に生じる方法が提供される。

【0016】

本発明の第4の側面に関連して、ビデオシーケンスのうち単独のピクチュアを構成する一連のマクロブロックの各マクロブロックをエンコードする時に使用する量子化パラメータの値を算出する方法であって、(イ)第1の時点で符号化されたピクチュアのバッファの第1のフルネスの程度を推定する工程と、(ロ)前記単独のピクチュアのエンコードされたバージョンのターゲットサイズを決定する工程と、(ハ)前記一連のマクロブロックの各マクロブロックに対して、(ハ1)前記一連のマクロブロックのうち、先行するマクロブロックの複雑さの累積された程度を決定することと、(ハ2)前記第1のフルネスの程度と前記複雑さの累積された程度と前記ターゲットサイズに作用して、第2の時点で前記符号化されたピクチュアのバッファの第2のフルネスの程度に戻すことと、(ハ3)前記第2のフルネスの程度に作用して、前記所定のマクロブロックをエンコードするときに使用する量子化パラメータの値に戻すことを行う工程とからなり、前記ターゲットサイズが、前記単一のピクチュアのエンコードされたバージョンを発生するときに前記一連のマクロブロックを分割すべきビット数に対応しており、所定のマクロブロックのエンコードされたバージョンのビットサイズが前記マクロブロックをエンコードするときに用いる量子化パラメータの値の関数であり、前記第2の時点が前記第1の時点の後に生じる方法が提供される。

【0017】

本発明の第5の側面に関連して、マルチプロセッサエンコーダにより、並列に、複数のピクチュアをエンコードするのに先立ち、未来のある時点において、符号化されたピクチュアのバッファのフルネスの程度を推定することによって前記マルチプロセッサエンコーダを操作する方法であって、

第1のビデオエンコード手順を用いて複数のピクチュアのエンコードされたバージョンを表わすのに必要なビット数を予測する工程と、

前記将来の時点に先行する第1の時点で符号化されたピクチュアのフルネスの程度を算出する工程と、

以前にエンコードされた複数のピクチュアのエンコードされたバージョンを表わすのに必要な以前に予測されたビット数を以前にエンコードされた複数のピクチュアを表わすのに必要な実際のビット数を比較することによって、不整合項を生成する工程と、

第1の時点で符号化されたピクチュアの算出されたフルネスの程度と前記第1のピクチュ

10

20

30

40

50

ュアのエンコードされたバージョンを表わすのに必要な予測されたビット数と前記不整合項を組み合わせて前記将来の時点で符号化されたピクチャの推定されたフルネスの程度に戻る工程を具備し、

前記将来の時点が前記第1のピクチャのエンコードされたバージョンが符号化されたピクチャにロードされる時点に対応すると共に、前記第1の時点が符号化されたピクチャのフルネスの程度が正確に算出される最も近接する時点を現している方法が提供される。

【0018】

本発明の第6の側面に関連して、符号化されたピクチャのバッファのアンダーフローを防ぐようにマルチプロセッサビデオエンコーダを操作する方法であって、第1のビット数を用いてビデオシーケンスのピクチャをエンコードする工程と、ある時間の間前記ピクチャのエンコードされたバージョンを保管する工程と、符号化されたピクチャのバッファのフルネスの程度を算出する工程と、符号化されたピクチャのバッファにアンダーフローの危険があるか否かを判定する工程と、符号化されたピクチャのバッファがアンダーフローする危険性が存在する場合には、符号化されたピクチャのバッファがもはやアンダーフローする危険性のないようなレベルに前記符号化されたピクチャのバッファのフルネスを増やすのに必要な充填ビットの数を算出する工程と、前記数の充填ビットを前記ピクチャのエンコードされたバージョンに付加して、それによって、第2のビット数まで前記ピクチャのエンコードされたバージョンのサイズを増やす工程と、前記ピクチャのエンコードされたバージョンを送信する工程を具備する方法が提供される。

【0019】

本発明の第7の側面に関連して、複数のデュアルパスマルチプロセッサシングルチャンネルエンコーダを有するマルチチャンネルビデオエンコーダであって、各シングルチャンネルエンコーダが共通のビデオクロックを共有し第1のパスエンコーダと第2のパスエンコーダを有するものを操作する方法であって、複数のデュアルパスマルチプロセッサシングルチャンネルエンコーダのそれぞれが、複数の一連のピクチャと組み込まれたビデオクロックを含む、単独の非エンコードプログラムビデオストリームを受信し、前記単独の非エンコードプログラムビデオストリームのコピーを生成し、前記単独の非エンコードプログラムビデオストリームを保管し、前記コピーの組み込まれたビデオクロックを修正してそれを共通のビデオクロックに同期させ、前記コピーに対して第1のパスエンコーダを実施して、該コピーの複雑系統計を行う工程と、複雑系統計を行うことによって、それぞれの前記単独の非エンコードプログラムビデオストリームのピクチャの好適なサイズを判定する工程と、前記複数のデュアルパスマルチプロセッサシングルチャンネルエンコーダのそれぞれが、入力として前記好適なターゲットサイズを用いて、前記単独の非エンコードプログラムビデオストリームに対し第2のパスエンコーダを実施して、エンコードされた単独のプログラムビデオストリームを生成し、複数のエンコードされた単独のプログラムビデオストリームを互いに積算してマルチプログラムトランスポートストリームとし、そのマルチプログラムトランスポートストリームを送信する工程を具備する方法が提供される。

【好適な実施の形態の説明】

【0020】

詳細な説明

以下の本発明の実施の形態のそれぞれは、いずれか一つのビデオコード規格に特定されるものではない。しかし、この明細書に関して、エンコードするために、問題のビデオコード規格が個別のピクチャ又はフレームをマクロブロックに分割し、複数の一連のピクチャを分割してエンコーディングのためにピクチャ群(GOP)にするような従来の技術を利用しているものと仮定する。

【0021】

図2に関連して、シングルチャンネルデュアルパスエンコーダ5は、入力として非エンコードビデオストリームINを受信し、そして、出力としてエンコードされたSPTS

OUT (VBR又はCBR)を送信する。デュアルパスエンコーダ5は、マルチプロセッサアーキテクチャを有し、低複雑性の第1パスエンコーダ8と高複雑性の第2パスエンコーダ12を具備する。第1パスエンコーダ8は、従来のシングルプロセッサエンコーダであり、第2パスエンコーダ12は、上記のようなパイプラインパラレルエンコーダであり、 m 個の処理ユニット 16_x ($x = 1 \cdots m$)を有する。ビデオストリームINの各ピクチャ j は、保管及び遅延ユニット20とデシメーションユニット24によって受信される。保管及び遅延ユニット20は、 d ピクチャの深さと入力ストリームのピクチャ用の先読みウィンドウを具備するFIFO (first-in, first-out)型バッファとして作動する。

【0022】

10

デシメーションユニット24は、決められた量だけビデオストリームINの各ピクチャ j の解像度を下げて、第1パスエンコーダ8の図示しない処理ユニットの計算上の複雑さを第2パス処理ユニット 16_x のそれと比較して低くなるようにするために対応する低解像度バージョン j_d を生成する。簡単な例として、デシメーションユニット24がビデオストリームINの解像度を水平方向と垂直方向の両方向において2分の一だけ下げるとすると、第1パスエンコーダ8の複雑さは第2パス処理ユニット 16_x の複雑さの四分の一しか必要でなくなるので、したがって、単独のピクチャをエンコードするのにデュアルパスエンコーダ5にとって必要なトータルな計算上の複雑さはシングルパスエンコーダよりも約25%大きい。

【0023】

20

デシメートされたビデオストリームが第1パスエンコーダ8に入力され、そこで、第1パスエンコーダ8に実装された従来のエンコードプロセスの過程において、ピクチャレベル及びマクロブロックレベルの複雑系統計が各デシメートされたピクチャ j_d に対して計算される。複雑系統計が使用されて、各デシメートされたピクチャ j_d に対して複雑度 c_j とマクロブロックからマクロブロックまでの累積された複雑度を生成する。これら複雑系統計と複雑度は、下記の先読みウィンドウに記憶される。第2パスエンコーダ12は、ビデオのピクチャ j を受信する。そして、このピクチャ j はエンコーディングのために処理ユニット 16_x に割り当てられる。第2パスエンコーダのパラレル設計のために、表1に示された例と同様に各タイムスロット u の始めにおいて、複数のピクチャが第2パスエンコーダにロードされうる。

30

【0024】

シングルチャンネルデュアルパスエンコーダ5が図2に実線で囲まれているスタンドアローンシングルチャンネルエンコーダである場合、第2パスエンコーダ12もコントロールユニットとして作動することができる。それとは別に、別のマスタコントロールユニット28は、ピクチャレベルレートコントロールアルゴリズムに対し、入力として、複雑度 c_j を用い、非デシメートピクチャ j のエンコードされたバージョンのサイズに対するターゲット値 T_j を選択する。ピクチャレベルレートコントロールアルゴリズムは、また、 T_j を用いてタイムスロット u の終了時にデュアルパスエンコーダの仮想バッファのフルネスの推定値 B_u を算出する。ピクチャレベルレートコントロールアルゴリズムは、第1パスエンコーダ8内に実装されうる。

40

【0025】

従って、第1パスエンコーダ8とピクチャレベルレートコントロールアルゴリズムは、各タイムスロット u の始めにおいて第2パスエンコーダ12に同時に複数のターゲット値を提供するほど十分に早いレートでデシメートされたピクチャを処理することができなければならない。第2パスエンコーダ12は、 T_j を用いて、ピクチャ j のエンコードされたバージョンを生成し、それがVBR又はCBRのSPTS OUTの一部としてシングルチャンネルデュアルパスエンコーダ5から出力される。

【0026】

第2パスエンコーダ12は、移動平均ウィンドウ内においてエンコードされた各ピクチャについての情報を一時的に記憶する。下記のように、レートコントロールアルゴリズム

50

ムは、ピクチャターゲットサイズを算出するときに先読みウィンドウと移動平均ウィンドウの双方に記憶されている情報を用いる。

【 0 0 2 7 】

シングルチャンネルデュアルパスエンコーダ 5 の出力が C B R ビデオストリームである場合、エンコーダの仮想バッファのフルネスがマイナスにならないようにビットをエンコーダの仮想バッファに充填することが時には必要となる。パイプライン遅延の後まで仮想バッファのフルネスの真の値が利用できないので、正しい数の充填ビットが付加できるように、充填ビットインサータ 3 0 において圧縮された出力ストリームが少なくとも p 個のタイムスロットの間バッファされる。

【 0 0 2 8 】

次のビデオシーケンスは左（過去）から右（未来）に向かって表示順に示されている。下付数字は下記の表 2 に示すエンコード順を示している。

【 0 0 2 9 】

・ ・ ・ B₋₅ B₋₄ P₋₆ B₋₂ B₋₁ P₋₃ B₁ B₂ I₀ B₄ B₅ P₃ B₇ B₈ P₆ B₁₀ B₁₁ P₉ B₁₃ B₁₄ P₁₂ B₁₆ B₁₇ I₁₅ ・ ・ ・

マイナスの下付数字は、以前にエンコードされたピクチャであることを示す。

【 0 0 3 0 】

表 2 は、u = 1 から始まる一連のタイムスロットにおいて図 2 に示すマルチプロセッサエンコーダの作業を示している。表 1 の場合と同様に、表 2 も単純化されておりエンコード作業のみを示している。非圧縮ピクチャを受信する作業、再構成されたピクチャを送信する作業、再構成されたピクチャを受信する作業、必要な遅延などの作業は、それらがレートコントロールアルゴリズムの例示にとって不可欠ではないので、割愛されている。

【 0 0 3 1 】

【表 2】

プロセッサ	u = 1	u = 2	u = 3	u = 4	u = 5	u = 6	u = 7	u = 8
1	I ₀	P ₃	P ₆	P ₉	P ₁₂	I ₁₅	P ₁₈	P ₂₁
2	B ₂	B ₂	B ₄	B ₄	B ₁₀	B ₁₀	B ₁₆	B ₁₆
3	B ₁	B ₁	B ₅	B ₅	B ₁₁	B ₁₁	B ₁₇	B ₁₇
4	B ₅	B ₁	B ₁	B ₇	B ₇	B ₁₃	B ₁₃	B ₁₉
5	B ₄	B ₂	B ₂	B ₈	B ₈	B ₁₄	B ₁₄	B ₂₀
利用可能な エンコード された ピクチャ	P ₉ B ₈ B ₇	P ₆ B ₅ B ₄	P ₃ B ₂ B ₁	I ₀ B ₁ B ₂	P ₃ B ₄ B ₅	P ₆ B ₇ B ₈	P ₉ B ₁₀ B ₁₁	P ₁₂ B ₁₃ B ₁₄

【 0 0 3 2 】

図 3 は、デュアルパスエンコーダ 5 が表 2 の一部に示すビデオシーケンスをエンコードしている過程にあるときの第 1 パスエンコーダ 8 と第 2 パスエンコーダ 1 2 の間の時間関係を示している。カレントタイムスロット（u = 1）の開始時点において、エンコーダのレートコントロールアルゴリズムは、先読みウィンドウに記憶された前もって得られた第 1 パスデータと移動平均ウィンドウに記憶された前もって得られた第 2 パスデータを利用して、ピクチャ I₀、B₋₂、B₋₁のターゲットサイズを算出する。先読みウィンドウに記憶された第 1 のパスデータは、第 2 パスエンコーダによってカレントにエンコードされているピクチャと第 1 のパスエンコーダ 8 によって以前にエンコードされているがま

10

20

30

40

50

だ第2パスエンコーダ12に達していない後続のピクチュアに関連する。移動平均ウィンドウに記憶された第2パスデータは、最近第2パスエンコーダ12によってエンコードされたピクチュアに関連する。先読みウィンドウと移動平均ウィンドウのサイズが、そこに記憶されるピクチュアの数に関して、大きければ大きいほど、レートコントロールアルゴリズムが第2パスエンコーダによってエンコードされるピクチュアのターゲットサイズをより正確に算出することができる。

【0033】

以下の記号と変数のリストは、シングルチャンネルエンコーダ用のピクチュアレベルレートコントロールアルゴリズムを記述するときの明確化のために提供される。

【0034】

i は、シングルチャンネルエンコーダユニットのためのインデックスとして、そして、マルチチャンネルエンコーダのシングルチャンネルエンコーダユニット内の対応するビデオストリーム用のインデックスとして、使用される。上記の図2に図示された実施の形態において、 i は指定されていない。

【0035】

j は、ピクチュア番号用のインデックスとして使用される。ピクチュアは1から始まって次のピクチュア毎に1ずつ増えていくように番号付けされる。

【0036】

u は、タイムスロット用のインデックスとして使用され、特に第2パスエンコーダに関連して使用される。タイムスロットは、1から始まって後続のタイムスロット毎に1ずつ増えていくように番号付けされている。

【0037】

U_u は、タイムスロット u の間に第2パスエンコーダによってエンコードされたピクチュアの集合を示すために使用される。表2に示された例においては、 $U_5 = \{P_{12}, B_{10}, B_{11}, B_7, B_8\}$ である。

【0038】

k は、集合 U_u におけるピクチュア用のインデックスとして使用される。集合 U_u の所定のピクチュア k がビデオストリームに関して対応するインデックス j を有することに特に言及する。使用されるインデックスは、コンテキストに依存する。

【0039】

$|B_i|$ は、シングルチャンネルエンコーダの仮想バッファのビットサイズである。例えば、MPEG-2エンコーダにおいては、仮想バッファはVBVバッファであり、H.264エンコーダにおいては、仮想バッファは符号化ピクチュアバッファ(CPB)である。 $|B_i|$ が特定の規格によって許容された最大のサイズではないが、希望のビットレートに基づくビデオストリーム i の実際の最大値であることに特に言及する。

【0040】

$B_{i,u}$ は、タイムスロット u における第2パスエンコーダの全てのピクチュアのエンコーディングの直前においてビデオストリーム i のシングルチャンネルエンコーダの仮想バッファの推定フルネスである。

【0041】

$b_{i,j}$ は、第1パスエンコーダによってエンコードされたときの、ビデオストリーム i から j 番目のピクチュアのエンコードバージョンのビットサイズである。

【0042】

$q_{i,j}$ は、ビデオストリーム i のピクチュア j のために第1パスエンコーダによって使用される量子化器である。

【0043】

$C_{i,j} = q_{i,j} b_{i,j}$ は、第1パスエンコーダによって測定されたときのビデオストリーム i のピクチュア j の複雑さを表す値である。 $C_{i,j}$ が第1パスエンコーダによって測定された後で、それは先読みウィンドウに記憶される。

【0044】

10

20

30

40

50

ビデオストリーム i のピクチャ j が I タイプであるときは $c_{i,j}^I = c_{i,j}$ であるが、

それ以外の場合には $c_{i,j}^I$ は未定義である。

【 0 0 4 5 】

ビデオストリーム i のピクチャ j が B タイプであるときは $c_{i,j}^B = c_{i,j}$ であるが、

それ以外の場合には $c_{i,j}^B$ は未定義である。

【 0 0 4 6 】

$\hat{B}_{i,u}$ は、タイムスロット u における第 2 パスエンコーダの全てのピクチャのエンコーディングの直前においてビデオストリーム i のシングルチャンネルエンコーダの仮想バッファの実際のフルネスである。

【 0 0 4 7 】

$\hat{b}_{i,j}$ は、第 2 パスエンコーダによってエンコードされたときの、ビデオストリーム i から j 番目のピクチャのエンコードされたバージョンのビットサイズである。

【 0 0 4 8 】

$\hat{q}_{i,j}$ は、ビデオストリーム i のピクチャ j のために第 2 パスエンコーダによって使用される平均量子化器である。この平均量子化器は個々のマクロブロック量子化器の合計をマクロブロック数によって割ったものである。

【 0 0 4 9 】

$\hat{c}_{i,j} = \hat{q}_{i,j} \hat{b}_{i,j}$ が、第 2 パスエンコーダによって測定されたときのビデオストリーム i のピクチャ j の複雑さを表す値である。

【 0 0 5 0 】

ビデオストリーム i のピクチャ j が I タイプであるときは $\hat{c}_{i,j}^I = \hat{c}_{i,j}$ であるが、

それ以外の場合には $\hat{c}_{i,j}^I$ は未定義である。同様の定義が $\hat{c}_{i,j}^P$ と $\hat{c}_{i,j}^B$ に加えられる。

【 0 0 5 1 】

d は、保管及び遅延ユニットにおける先読みウィンドウの深さである。

【 0 0 5 2 】

p は、第 2 パスエンコーダのパイプラインの深さである。

【 0 0 5 3 】

e_u は、タイムスロット $u - p$ からタイムスロット $u - 1$ までにエンコードされたピクチャの数である。

【 0 0 5 4 】

$m_{i,u}$ は、タイムスロット u における推定されたバッファフルネス $B_{i,u}$ と実際のバッ

ファフルネス $\hat{B}_{i,u}$ の間の誤差を反映する小さな「不整合」項である。

【 0 0 5 5 】

R_i は、出力ビデオストリーム i にとってのピクチャ時間毎の総ターゲットビット数である。出力ビデオストリーム i が CBR である場合には、 R_i は定数である。

【 0 0 5 6 】

β_i は、シングルチャンネルエンコーダの仮想バッファのフルネスが調節されるスピードにとっての感度因子である。それは以下に説明される。

10

【 0 0 5 7 】

$T_{i,j}^0$ は、ビデオストリーム i のピクチャ j のエンコードされたバージョンの当初のターゲットサイズである。

【 0 0 5 8 】

$T_{i,j}$ は、ビデオストリーム i のピクチャ j の調節されたターゲットサイズである。

【 0 0 5 9 】

V_u は、タイムスロット $u - p$ からタイムスロット $u - 1$ までにエンコードされたピクチャの総ビットレートターゲットの集合である。

20

【 0 0 6 0 】

$w_{i,j}$ は、ビデオストリーム i のピクチャ j の第 1 パス複雑値 $c_{i,j}$ を第 2 パス複雑値

$\hat{c}_{i,j}$ になる近似値に変換するために使用される重み付けである。この変換は線形ではな

く、従って、 $w_{i,j}$ は、それがコンテキストに依存するのでタイムスロット毎にアップデートされる。より詳細な説明は以下の式 2, 3, 4 において与えられる。

【 0 0 6 1 】

ビデオストリーム i のピクチャ j が I タイプであるときは $w_{i,j}^I = w_{i,j}$ であるが

30

、それ以外の場合には $w_{i,j}^I$ は未定義である。同様の定義が $w_{i,j}^P$ と $w_{i,j}^B$ に加えられる。

【 0 0 6 2 】

$w_{i,j}^I, w_{i,j}^P, w_{i,j}^B$ は、 $w_{i,j}$ の算出に使用される重み付け因子である。

【 0 0 6 3 】

$S_{i,k}$ は、先読みウィンドウ内の全てのピクチャの重み付けされた第 1 パス複雑度の総和である。総和 $S_{i,k}$ は、(第 2 パスエンコーダによって予めエンコードされたピクチャに関連する) 移動平均ウィンドウとは異なる第 1 パスエンコーダの先読みウィンドウに亘って実施されることに特に言及する。

40

【 0 0 6 4 】

N は、移動平均ウィンドウ内のピクチャの数である。 $N^I \cdots N^P, N^B$ は、同様の定義を有する。

【 0 0 6 5 】

ピクチャレベルレートコントロールアルゴリズムは、各ピクチャ k から $k + (d - 1)$ の第 1 パスエンコーディングから先読みウィンドウに記憶されたデータと移動平均ウィンドウに記憶された、第 2 パスエンコーダによって前もってエンコードされていた他のピクチャからの第 2 パスデータを用いて、タイムスロット u の始めに第 2 パスエンコーダに入る各ピクチャ k に対する当初のターゲットビット数を算出する。当初のターゲットビット数は、式 1 によって算出される。

50

【 0 0 6 6 】

【 数 1 】

$$T_{i,k}^o = \frac{w_{i,k} c_{i,k}}{S_{i,k}} \cdot \left(d \cdot R_i + \sigma_{i,k} \left(\frac{1}{2} |B_i| - B_{i,u} + m_{i,u-1} \right) \right) \text{ for } k \in (U_u) \quad (\text{式 1})$$

【 0 0 6 7 】

重み付け $w_{i,k}$ は、ビデオストリーム i の移動平均ウィンドウ内のピクチャに関する以前の第 1 パス複雑度と以前の第 2 パス複雑度の関数である。図 2 に示されたもののようなパイプライン実装において、エンコーディングのまだ終わっていないピクチャの第 2 パス複雑さは利用することができないことに特に言及する。

10

【 0 0 6 8 】

3つの異なる重み付け関数 W_i^{type} が存在するが、各ピクチャタイプ (I, P, B) に対して 1 つのものである。

それらは、全て一般式を有する。

【 0 0 6 9 】

【 数 2 】

$$w_{i,k}^I = W_i^I \left(\{c_{i,j}^I \mid j \leq k\}, \{\hat{c}_{i,j}^I \mid j \leq k - d - p\} \right) \quad (\text{式 2 a})$$

【 0 0 7 0 】

20

【 数 3 】

$$w_{i,k}^P = W_i^P \left(\{c_{i,j}^P \mid j \leq k\}, \{\hat{c}_{i,j}^P \mid j \leq k - d - p\} \right) \quad (\text{式 2 b})$$

【 0 0 7 1 】

【 数 4 】

$$w_{i,k}^B = W_i^B \left(\{c_{i,j}^B \mid j \leq k\}, \{\hat{c}_{i,j}^B \mid j \leq k - d - p\} \right) \quad (\text{式 2 c})$$

【 0 0 7 2 】

重み付け関数 W_i^I の簡略式は、適切なピクチャタイプの以前の第 2 パス複雑度を同じピクチャタイプの以前の第 1 パス複雑度によって割った比の移動平均である。例えば、I タイプピクチャである場合、

30

【 0 0 7 3 】

$$W_i^I = \frac{\sum_{j \in I_set} \frac{\hat{c}_{i,j}^I}{c_{i,j}^I}}{N^I} \quad (\text{式 3})$$

であり、ここで、I 集合は N^I サイズであり、その元は式 2 a の $w_{i,j}^I$ の一般式を満足するもっとも大きな整数である。

40

【 0 0 7 4 】

N^I が移動平均ウィンドウ内の I タイプのピクチャの数であるけれども、スタートアップ時に、はじめの $p + d$ 個のピクチャに対しては第 2 パス複雑度が利用できず、したがって、 $w_{i,k}$ は第 2 パスピクチャとデシメートされた第 1 パスピクチャの複雑さの間が一定の比率となることに特に言及する。

【 0 0 7 5 】

一般的に、最も近接するピクチャは他よりもより重く重み付けされるべきであり、

I タイプのピクチャの重み付けに関する重み付け関数 W_i^I のより複雑な実施の形態は
【 0 0 7 6 】

【 数 5 】

$$W_{ik}^I = \frac{\Omega^I \left(\frac{\hat{c}_{i,k_0}^I}{c_{i,k_0}^I} \right) + \sum_{j \in I_set} \frac{\hat{c}_{i,k}^I}{c_{i,k}^I}}{N^I + \Omega^I - 1}, \quad k_0 = k - d - p \quad (\text{式 4})$$

10

であり、ここで、 $I_set' = I_set \setminus \{K_0\}$ である。P_set 及び B_set に関して適切な定義を有する P ピクチャ及び B ピクチャに対しても同じ式が存在する。

【 0 0 7 7 】

従って、式 1 に関連して、ピクチャ j に関し重み付け $w_{i,j}$ と第 1 パス複雑度 $c_{i,j}$ の積が、ピクチャ j が第 2 パスエンコードによって受信される前に、ピクチャ j

の第 2 パス複雑度 $\hat{c}_{i,j}$ の近似値を与える。この近似値はスケーリング因子 $S_{i,k}$ によって割られる。

20

【 0 0 7 8 】

【 数 6 】

$$S_{i,k} = \sum_{j=k}^{k+d-1} (w_{i,j \bmod d} c_{i,j}) \quad (\text{式 5})$$

【 0 0 7 9 】

ここで、d は、先読みウィンドウのサイズを表している。複雑比 $w_{i,k} c_{i,k} / S_{i,k}$ は、集合 U_u の他のピクチャの近似された第 2 パス複雑度に比較した、ピクチャ k の近似された第 2 パス複雑度の測定値を表す。

30

【 0 0 8 0 】

ピクチャレベルレートコントロールアルゴリズムがビデオストリーム i のピクチャ k のエンコードされたバージョンに対して $T_{i,k}$ の適切な値を選択するために、該アルゴリズムはエンコードの仮想バッファの現状のフルネスを明らかにする必要がある。

【 0 0 8 1 】

理想的には、ピクチャレベルレートコントロールアルゴリズムは、タイムスロット u

の実際のバッファフルネス $\hat{B}_{i,u}$ を利用する。しかし、p 個のタイムスロットのエンコー

40

ダのパイプライン遅延が u + p 個のタイムスロットまで実際のバッファフルネス $\hat{B}_{i,u}$ を分からないものとする。

【 0 0 8 2 】

従って、従前のタイムスロットからの情報に基づいて、推測された仮想バッファのフルネス $B_{i,u}$ が、式 1 の実際の仮想バッファのフルネスに代って使用される。

【 0 0 8 3 】

【数 7】

$$B_{i,u} = B_{i,u-p} + \sum_{k \in V_u} T_{i,k} - e_u R_i \quad (\text{式 6})$$

【0084】

定義により、 V_u と e_u は、タイムスロット($u-p$)からタイムスロット($u-1$)までにおけるターゲットとピクチャに関連する。式6は、大部分、各種の圧縮規格において定義された仮想バッファのフルネス計算に対応しているが、以前の計算よりもむしろ仮想バッファのフルネスの以前の推定を使用する。

【0085】

10

以前のタイムスロットの仮想バッファのフルネスの推定における誤差を補償するために、不整合項 $m_{i,u}$ が、式1に用いられる。不整合項 $m_{i,u}$ は、仮想バッファの実際のフルネス $\hat{B}_{i,u-p}$ にとって利用可能な最もアップツードータな値と推定されたフルネス $B_{i,u-p}$ の間の差異に対応する。

【0086】

【数 8】

$$m_{i,u} = \frac{(\hat{B}_{i,u-p} - B_{i,u-p})}{p} \quad (\text{式 7})$$

20

【0087】

所定のピクチャの相対的な複雑性又は所定のタイムスロットにおけるバッファのフルネスにかかわらず、非マイナスのビット数が各ピクチャに割り振られなければならない。所定のピクチャ k の相対的な複雑性が決してマイナスとはならないので、式1の考察から、以下のような関係がピクチャ k に対して真でなければならないということになる。

【0088】

【数 9】

30

$$d \cdot R_i \geq -\sigma_{i,k} \left(\frac{1}{2} |B_i| - B_{i,u} + m_{i,u-1} \right) \quad (\text{式 8})$$

【0089】

演算において、不整合項 $m_{i,u-1}$ は、比較的によく小さく、式8に対して十分な効果を有するものではない。従って、式8を解析するために、不整合項 $m_{i,u-1}$ は無視されうる。式8によって定義されたルールは、仮想バッファが満杯である時、 $B_{i,u} = |B_i|$ の時、に破られる最も大きな危険性があり、結果的に、

【0090】

40

【数 10】

$$d \cdot R_i \geq \frac{1}{2} \sigma_{i,k} |B_i| \quad (\text{式 9})$$

となる。

【0091】

当初のターゲットビット割り当てを非マイナスにするように単純に要求するよりはむしろ、本発明のより複雑な実施の形態は、最小のビット割り当て値 $\sigma_{i,k}$ を割り当てが、それは、ピクチャ k のピクチャタイプに関して、どのようなビデオコード規格が第2

50

パスエンコーダによって実現されたとしても必要な最小のビット数に比例する。

【 0 0 9 2 】

従って、式 9 は、

【 0 0 9 3 】

【 数 1 1 】

$$d \cdot R_i - \frac{1}{2} \sigma_{i,k} |B_i| > \varepsilon_{i,k} \quad (\text{式 1 0})$$

のように書き換えられる。

10

【 0 0 9 4 】

ピクチャ特性とは独立してピクチャレベルレートコントロールアルゴリズムによってピクチャからピクチャに調整されうる式 1 0 の成分が $\sigma_{i,k}$ のみであり、 $\sigma_{i,k}$ の適切なピクチャタイプ依存値は $\sigma_{i,k}$ について式 1 0 を解決することによって見つけることが可能である。

【 0 0 9 5 】

【 数 1 2 】

$$\sigma_{i,k} = \frac{2(d \cdot R_i - \varepsilon_{i,k})}{|B_i|} \quad (\text{式 1 1})$$

20

【 0 0 9 6 】

しかし、不整合項 $m_{i,k}$ により、式 1 によって与えられた当初のターゲットビット割り当て $T_{i,k}^0$ は、ピクチャ k のピクチャタイプにとって使用しうるビデオ規格により必要な最小のビット数以下の結果を得ることが可能である。

従って、当初のターゲットビット割り当ては、最小ビット数と比較され、より大きな数がピクチャ k に対して最終ビット割り当て $T_{i,k}$ として使用される。例として、H. 264 の場合、

30

【 0 0 9 7 】

【 数 1 3 】

$$T_{i,k} = \max(T_{i,k}^0, \alpha_{i,k} R_i) \quad (\text{式 1 2})$$

であるが、ここで、 $\alpha_{i,k}$ はビットレート及び規格依存の重み付けであり、I タイプ、P タイプ、B タイプのピクチャに対して参照テーブルによって表される。

【 0 0 9 8 】

【 数 1 4 】

40

$$a_{i,k} = a_{i,k}^I [\min(t^* \text{ ヲトレート}), \dots, \max(t^* \text{ ヲトレート})] \quad (\text{式 1 3 a})$$

$$a_{i,k} = a_{i,k}^P [\min(t^* \text{ ヲトレート}), \dots, \max(t^* \text{ ヲトレート})] \quad (\text{式 1 3 b})$$

$$a_{i,k} = a_{i,k}^B [\min(t^* \text{ ヲトレート}), \dots, \max(t^* \text{ ヲトレート})] \quad (\text{式 1 3 c})$$

【 0 0 9 9 】

図 2 に関連して、上記のピクチャレベルレートコントロールアルゴリズムを利用する

50

ことによって、シングルチャンネルデュアルパスエンコーダ5のコントロールユニットは、十分な精度を有して第2パスエンコーダの仮想バッファのフルネスを予測することができ、第2パスエンコーダの平行ルアーキテクチャに固有のパイプライン遅延にかかわらず、ピクチャレベルレートコントロールアルゴリズムが各ピクチャに対してビット数を割り当てることが可能である。

【0100】

図2の実線で示されたエンコーダは、シングルチャンネルエンコーダであり、従って、インデックス*i*は指定されない。図4に図示され、そして、図2に点線で示された本発明の他の実施の形態において、*n*個のシングルチャンネルエンコーダ5 - *i* (*i* = 1 . . . *n*) は、マルチチャンネルエンコーダ32内に配置される。マルチチャンネルエンコーダ32は、入力として*n*個の非エンコードビデオストリームIN - *i* (*i* = 1 . . . *n*)を受信し、そして、*n*個のSPTSである出力OUT - *i* (*i* = 1 . . . *n*)を所望の可変ビットレートでそれぞれ送信する。必要であれば、異なるビデオコード規格がチャンネル毎に使用されうる。エンコーダ32は、マスタコントロールユニット28と、各ビデオストリーム*i*の各ピクチャ*j*に関連して、第1パス複雑値 $c_{i,j}$ が、デュアルパスエンコーダ5 - *i*からマスタコントロールユニット28に送信される。 $c_{i,j}$ が受信された時点で、マスタコントロールユニット28は、ターゲット値 $T_{i,j}$ を算出し、このターゲット値をデュアルパスエンコーダ5 - *i*に送り返す。ターゲット値 $T_{i,j}$ を受信した後で、デュアルパスエンコーダ5 - *i*はピクチャ*j*をエンコードしてそのエンコードされたバージョンを出力OUT - *i*の一部として送信する。そして、SPTSである出力OUT - *i*は、マルチチャンネルエンコーダ32から送信される。図4の図示された実施の形態のためのピクチャレベルレートコントロールアルゴリズムは、*i* = 1 . . . *n*であること除いて、上記のピクチャレベルレートコントロールアルゴリズムと同じである。

【0101】

図4において点線で示された本発明の他の実施の形態において、*n*個のSPTSである出力OUT - *i*は、マルチプレクサユニット36によって一緒に統計的に多重送信されて総計固定ビットレート(CBR)を有するシングルのマルチプログラムトランスポートストリーム(MPTS)M__OUTを形成する可変ビットレートストリームである。出力M__OUTが、図示しない、下流のマルチチャンネルデコーダに送られ、そこで、M__OUTがバッファに保存され、デマルチプレックスされ、そして、デコードされる。*n*個の入力ビデオストリームが大規模なマルチチャンネルデコーダバッファに結局全て一緒に保存されるので、マルチチャンネルエンコーダは*n*個のシングルチャンネルエンコーダすべてにとって共有する仮想バッファモデルを使用し、そして、ピクチャレベルレートコントロールアルゴリズムは、*n*個のビデオストリームに対して集団的にジョイントターゲットビット割り当て操作を実行しなければならない。上記の実施の形態におけるのと同様に、出力OUT - *i*は、異なるビデオコーディング規格を用いてエンコードされうる。図5に関連して、シングルチャンネルエンコーダの出力ストリームOUT - *i*が多重送信されてCBR MPTSになるようにしている場合において、第1パスエンコーダ8への入力は、デシメーションユニット24と第1パスエンコーダ8の間に配設された対応する時間ベースコレクタ(TBC)ユニット40によって、時間ベースに共通のビデオクロックに対して訂正される。レートコントロールアルゴリズムは、*n*個のビデオストリームが共通のクロックに対して同期されない限り、*n*個全てのビデオストリームに対してジョイントターゲットビットレート割り当てを正確に行うことはできない。しかし、第2パスエンコーダ12の入力に対してビデオストリームを同様に同期することは必要ではない。第2パスエンコーダ12に入力されている訂正されていないクロックが第1パスエンコーダ8に入力する訂正されたクロックとはるかにずれているとき、時には2つのピクチャシーケンスの間に矛盾が生じることもある。このような矛盾は、第2パスエンコーダの入力に関し第1パスエンコーダの入力からピクチャの脱落となるか、又は、その逆となる。第2パスエンコーダ12は、容易にこのような矛盾を処理する。前者の場合、同じピクチャタイプの以前のピクチャをエンコードするのに使用されるパラメータは、現状のピクチャ

アをエンコードするために使用される。後者の場合、第2パスエンコーダ12は単純に現状のピクチュアを無視する。

【0102】

充填ビットを用いてシングルチャンネルエンコーダのC B R出力に埋め込む場合と同様に、マルチチャンネルエンコーダのM P T S出力に埋め込んでエンコーダの共有仮想バッファのフルネスがマイナスにならないようにすることも時には必要である。充填ビットの代わりに、マルチトランスポートストリームパッケージがシングルチャンネルエンコーディングのために上記と同様の方法で挿入されるが、その差は、 n 個のS P T Sと一緒に多重送信された後で挿入が起こることである。マルチプレクサ36から出力された後で、出力M__O U Tはマルチチャンネルマルチパッケージインサータ46によって受信される。圧縮された出力ストリームが少なくとも $\max(p_1, p_2, \dots, p_n)$ 個のタイムスロットの間バッファに保存され、正確な数のマルチトランスポートストリームパッケージが算出され加えられる。ここで、 p_i ($i = 1, 2, \dots, n$)はストリーム*i*に対する第2パスエンコーダのパイプラインの深さである。

10

【0103】

図4に示す実施の形態において、シングルチャンネルエンコーダユニット5 - i は、図5の点線で示された適応プレフィルタ42を具備することができる。第1パスエンコーダ8から得られたピクチュアレベル複雑系統計が使用されて、第2パスエンコーディングの前で現状のGOPに対するフィルタ42の強度を制御する。適応プレフィルタ42は、ローパスフィルタであり、その出力は所定のピクチュアのエンコードバージョンのサイズを減少するために選択される。第2パスエンコーダには、入力として、元の非圧縮ビデオソースか又は適応プレフィルタ42の出力が与えられる。ピクチュアレベルレートコントロールアルゴリズムは、複雑系統計に基づいてこの決定を行う。たとえば、出願中の米国特許出願第10/316,483号を参照してください。

20

【0104】

複数の個別のシングルチャンネルエンコーダのV B R出力が組み合わされて多重送信C B Rストリームを形成する、統計的なマルチプレクシングを伴うマルチチャンネルエンコーディングの場合のピクチュアレベルレートコントロールアルゴリズムは、上記のピクチュアレベルレートコントロールアルゴリズムよりも若干より複雑である。統計的なマルチプレクシングを伴うピクチュアレベルレートコントロールのために、中央のマスタレートコントロールが使用される。

30

【0105】

各チャンネルは、異なるビデオ規格(M P E G - 2, H . 2 6 4, S M P T E V C - 1など)を用いてエンコードされる。それぞれのシングルチャンネルエンコーダのアーキテクチャは同じであるが、ピクチュアレベルターゲットビット割り当て式は式1とは若干異なる。

【0106】

以下の記号と変数のリストは、統計的多重送信(マルチプレクシング)によるマルチチャンネルエンコーダ用のピクチュアレベルレートコントロールアルゴリズムを記述するときの明確化のために提供される。

40

【0107】

n は、マルチチャンネルエンコーダが受信することのできるチャンネルの数である。

【0108】

R は、多重送信された出力ストリームM__O U Tのためのピクチュア毎の一定のビットレートである。

【0109】

$|B|$ は、 n 個のシングルチャンネルエンコーダによって使用される共有仮想バッファのサイズである。

【0110】

B_u は、タイムスロットuにおける共有仮想バッファの推定されるフルネスである。

50

【 0 1 1 1 】

\hat{B}_u は、タイムスロット u における共有仮想バッファの実際のフルネスである。

【 0 1 1 2 】

m_u は、上記 $m_{i,u}$ と同様の不整合項である。

【 0 1 1 3 】

ピクチュアレベルレートコントロールアルゴリズムは、以下の式にしたがって、各タイムスロット u の間ビデオストリーム i の各ピクチュア k に対し、当初のターゲットビット数を算出する。

【 0 1 1 4 】

【 数 1 5 】

$$T_{i,k}^o = \frac{w_{i,k} c_{i,k}}{S_k} \cdot \left(d \cdot R + \sigma_{i,k} \left(\frac{1}{2} |B| - B_u + m_{u-1} \right) \right) \text{ for } k \in (U_u) \quad (\text{式 1 4})$$

【 0 1 1 5 】

S_k は、トータルな複雑度がこの場合、 n 個のビデオチャンネルの全ての複雑度に依存しているので、式 5 と比べて異なって定義される。

【 0 1 1 6 】

【 数 1 6 】

$$S_k = \sum_{i=1}^n \sum_{j=k}^{k+d-1} (w_{i,j \bmod d} c_{i,j}) \quad (\text{式 1 5})$$

同じく、 m_u は、 $m_{i,u}$ とは異なる定義を有する。

【 0 1 1 7 】

【 数 1 7 】

$$m_u = \frac{(\hat{B}_{u-p} - B_{u-p})}{p} \quad (\text{式 1 6})$$

【 0 1 1 8 】

上記のように、 n 個のチャンネルは、異なるビデオコード規格を用いてエンコードされる。この場合、 $c_{i,j}$ は、規格に依存しているが、 $w_{i,j}$ は、第 1 パス複雑系から第 2 パス複雑系へ変換するのみならず、異なるビデオコーディング規格間の複雑度を共通のベース規格に標準化する。特定のピクチュア k へのビットレート割り当ては、先読みウィンドウの深さ d とチャンネル数 n と各チャンネルの先読みウィンドウにおける全てのピクチュアのトータルな複雑度の組み合わせにより決定される。多重送信された出力 M_OUT が CBR であるので、各 $SPTS_OUT - i$ は $VBRSPTS$ としてエンコードされる。ピクチュアレベルレートコントロールアルゴリズムは、 n 個の $SPTS$ のビットレートを組み合わせて所望の CBR を形成することを確実にする。

【 0 1 1 9 】

本発明の上記実施の形態の全てにおいて、シングルチャンネルエンコーダのデュアルパス特性がシングルチャンネルエンコーダの第 2 パスエンコーダにおける MB レベルレートコントロールアルゴリズムを実現するのに使用される。ピクチュアレベルレートコントロールアルゴリズムによって提供されたターゲットビット割り当て $T_{i,k}$ は、ピクチュア k の MB に分割されねばならない。個々の MB の複雑度が MB から MB へ変化するので、例えば $T_{i,k}$ をピクチュア k の MB の数で割るような均一な割り当ては必ずしも最適な解決ではない。

10

20

30

40

50

【 0 1 2 0 】

第 1 パスエンコーダは、エンコーディング処理中マクロブロックの複雑度を測定する。第 1 パスエンコーダの分解能とビットレートが第 2 パスエンコーダのそれらとは異なるので、そして、第 2 パスエンコーダのモード決定が第 1 パスエンコーダにおいてなされるモード決定とは独立しているので、第 2 パスエンコーダによって測定されるであろうほどには、MB の複雑度の真の値は第 1 パスエンコーダの MB 複雑度の計測に基づいて正確には算出され得ない。しかし、重要なことは MB レベルの複雑度の真の値ではなく、マクロブロックの相対的な複雑度である。すなわち、MB から MB へどのくらい複雑度が変動するかである。第 1 パスエンコーダがピクチャの第 1 の領域をそのピクチャの第 2 の領域に関連して単純化 / 複雑化されるものとして測定すると、第 2 パスエンコーダは、同様に、そのピクチャの第 1 の領域をそのピクチャの第 2 の領域に関連して単純化 / 複雑化されるものとして測定する。このマッピング関係は、以下に図 5 及び図 6 に関連して詳細に説明される。MB レベルレートコントロールアルゴリズムは、この関係を有利に用いている。

10

【 0 1 2 1 】

本発明の上記実施の形態の全てにおいてマクロブロックレベルレートコントロールアルゴリズムは同じである。MB レベルレートコントロールの目的は、a) 第 2 パスエンコーダによって生成されたときのピクチャ j のエンコードされたバージョンの実際のサイズがピクチャレベルレートコントロールアルゴリズムによって割り当てられたターゲットサイズ $T_{i,j}$ に近づくことを確実にすることと、b) 利用可能なビットが相対的な MB 複雑度に基づいて各 MB に分配されることを確実にすることである。上記のように、MB レベルのビット割り当てを決定するときには、MB の複雑度の実際の値は特に重要なことではない。特定の MB の相対的な複雑度の方がはるかに意味のあることである。すなわち、MB から MB へ複雑度がどの程度変動するかである。ピクチャに対する MB から MB への第 2 パス複雑度の変動は、第 2 パス MB を対応する第 1 パス MB にマッピングして関連のある第 1 パス複雑度の変動を比較することによって予測されうる。

20

【 0 1 2 2 】

例示のために話を簡単にすると、デュアルパスエンコーダによって受信されたピクチャが 8×8 アレイの 64 個の MB からなっているものと仮定する。第 1 パスエンコーダへの入力はデシメートされ、第 1 パスエンコーダに入るピクチャの解像度を水平方向と垂直方向の両方向において 2 分の 1 に減らす。従って、第 1 パスエンコーダに入るピクチャは、図 6 に示すような 4×4 アレイの 16 個の MB からなる。現実には、ピクチャはその解像度に応じて何百又は何千すらの MB を有することも可能であるが、しかし、この簡単な例から導き出される結論や相関関係はいずれの解像度にも適用される。この単純化は欠くべからざるものにあらず。それは正に例示のためである。図 6 はピクチャ j の 2 つのバージョンを示している。1 つのバージョンにおいて、ピクチャ j は、第 1 パスエンコーダによって、 4×4 アレイのマクロブロックに分解される。第 2 のバージョンにおいて、第 2 パスエンコーダによって、 8×8 アレイのマクロブロックに分解される。第 1 のバージョンにおいて、各 MB は単独の大文字 (A - P) によって示され、第 2 のバージョンにおいて、各 MB は小文字 (a - p) と接尾数字 (1 - 4) によって示される。以下の記号と変数のリストは、図 6 に関連して MB レベルレートコントロールアルゴリズムを記述するときの明確化のために提供される。

30

40

【 0 1 2 3 】

N_1 は、第 1 パスエンコーダによって生成された所定のピクチャのエンコードされたバージョンにおけるマクロブロックの数である。図 6 に図示された例においては、 N_1 は 16 である。

【 0 1 2 4 】

k は、第 1 パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの順序数のインデックスとして使用される。マクロブロックの順序数は、ゼロで始まり、すなわち、 $k = 0 \cdots (N_1 - 1)$ であり、図 6 に図示され

50

た例においては、 $k = 0 \cdots 15$ である。

【0125】

v は、第1パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの行数である。 $H \cdot 264$ のようなある種のコーディング規格において必要とされるものではないが、マクロブロックが均等なサイズであると仮定すると、 v はマクロブロックの行数に対応する。図6に図示された例においては、 $v = 4$ である。

【0126】

i は、第1パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの順序数の行インデックスである。水平方向の順序数はゼロで始まり、すなわち、 $i = 0 \cdots (v - 1)$ である。

10

【0127】

h は、第1パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの列数である。 $H \cdot 264$ のようなある種のコーディング規格において必要とされるものではないが、マクロブロックが均等なサイズであると仮定すると、 h はマクロブロックの列数に対応する。図6に図示された例においては、 $h = 4$ である。

【0128】

j は、第1パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの順序数の列インデックスである。垂直方向の順序数はゼロで始まり、すなわち、 $j = 0 \cdots (h - 1)$ である。

20

【0129】

$N2$ は、第2パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの数である。図6に図示された例においては、 $N2$ は64である。

【0130】

\hat{k} は、第2パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの順序数のインデックスとして使用される。マクロブロックの順序数は、ゼロで始まり、すなわち、 $\hat{k} = 0 \cdots (N2 - 1)$ であり、図6に図示された例においては、 $\hat{k} = 0 \cdots 63$ である。

30

【0131】

\hat{i} は、第2パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの行インデックスである。

【0132】

\hat{j} は、第2パスエンコーダによって生成されたピクチャのエンコードされたバージョンにおけるマクロブロックの列インデックスである。

40

【0133】

m_h と m_v は、それぞれ水平方向と垂直方向における整数の乗数であるが、それらは第1パスエンコーダと第2パスエンコーダの間の解像度の相違を表す。 $m_h m_v = N2 / N1$ である。図6に図示された例においては、第2パスエンコーダの解像度は水平方向と垂直方向の両方向において第1パスエンコーダの解像度の2倍であり、 $m_h = 2$ と $m_v = 2$ である。

【0134】

g_k は、第1パスエンコーダによって測定された時のマクロブロック k の複雑度である。それは、マクロブロック k の第1パス複雑度としても知られている。

50

【 0 1 3 5 】

k は、マクロブロック 0 乃至 k の累積された第 1 パス複雑値である。

【 0 1 3 6 】

u_k は、所定のピクチャのマクロブロック数 k 未満のマクロブロックをエンコードした後で第 1 パスエンコーダによって生成されたビットの数である。

【 0 1 3 7 】

\hat{g}_k は、所定のピクチャのマクロブロック k のマップされた第 2 パス複雑度である

。 $\hat{\gamma}_k = \hat{g}_k + \hat{\gamma}_{k-1}$ である。

10

【 0 1 3 8 】

$\hat{\gamma}_k$ は、マクロブロック 0 乃至 k のマップされた累積第 2 パス複雑値であり、以下に詳述される。

【 0 1 3 9 】

\hat{u}_k は、所定のピクチャのマクロブロック数 k 未満のマクロブロックをエンコードした後で第 2 パスエンコーダによって生成されたビットの数である。

【 0 1 4 0 】

a_I は、I タイプのピクチャの MB レベルコントロール感応度を調整する増幅因子である。

20

【 0 1 4 1 】

a_P は、P タイプのピクチャの MB レベルコントロール感応度を調整する増幅因子である。

【 0 1 4 2 】

a_B は、B タイプのピクチャの MB レベルコントロール感応度を調整する増幅因子である。

【 0 1 4 3 】

Q_k は、第 2 パスエンコーダによるマクロブロック k のために使用される量子化器である。

30

【 0 1 4 4 】

q^{\max} は、量子化器の最大値である。それは規格に依存する。H. 264 の場合、 q^{\max} は 51 である。

【 0 1 4 5 】

v_k は、マクロブロック数 k 未満のマクロブロックをエンコードした後で第 2 パスエンコーダの仮想バッファの累積されたフルネスである。

【 0 1 4 6 】

v_f は、第 2 パスエンコーダの仮想バッファの当初のフルネスである。

【 0 1 4 7 】

第 2 パスエンコーダがマクロブロック \hat{k} をエンコードしているとき、マクロブロックレベルレートコントロールアルゴリズムは、対応する第 1 パスマクロブロック k の複雑さの情報を必要とする。

40

【 0 1 4 8 】

しかし、図 6 に示されているように、第 1 パスエンコーダによって受信されたデシメートされたビデオストリームは、第 2 パスエンコーダによって受信されたオリジナルのビデオストリームとは異なってエンコードされ、所定のピクチャのそれぞれのエンコードされたバージョンのマクロブロックは同じものではない。

【 0 1 4 9 】

従って、第2パスエンコーダが第1パスエンコーダによって生成された複雑さの情報を用いるためには、マクロブロックレベルレートコントロールアルゴリズムは、第2パ

スマクロブロック \hat{k} を第1パスマクロブロックにマップして、対応する第1パスマクロブロック k の順序を決定する。このことは、第1パスエンコーダと第2パスエンコーダの間の水平方向と垂直方向の解像度における既知の差を用いて、実現される。

【0150】

【数18】

$$\hat{i} = \left\lfloor \frac{\hat{k}}{m_h h} \right\rfloor \quad (\text{式17}) \quad 10$$

【0151】

【数19】

$$\hat{j} = \text{mod}(\hat{k}, m_h h) \quad (\text{式18})$$

式17と18は、それぞれ第2パス行列順序を返す。そして、対応する第1パス行列順序は、

【0152】

【数20】

$$i = \left\lfloor \frac{\hat{i}}{m_v} \right\rfloor \quad (\text{式19}) \quad 20$$

【0153】

【数21】

$$j = \left\lfloor \frac{\hat{j}}{m_h} \right\rfloor \quad (\text{式20}) \quad 30$$

のように算出され、そして、2パスマクロブロック \hat{k} に対応する第1パスマクロブロックの順序 k が算出される。

【0154】

【数22】

$$k = h \cdot i + j \quad (\text{式21})$$

再び図6に言及して、MBがn2の場合、

【0155】

40

【数 2 3】

$$\hat{k}_{n2} = 51$$

$$\hat{i} = \left\lfloor \frac{51}{2 \cdot 4} \right\rfloor = \left\lfloor \frac{51}{8} \right\rfloor = 6$$

$$\hat{j} = \text{mod}(51, 2 \cdot 4) = 3$$

10

$$i = \left\lfloor \frac{6}{2} \right\rfloor = 3$$

$$j = \left\lfloor \frac{3}{2} \right\rfloor = 1$$

$$k = 4 \cdot 3 + 1 = 13 = k_N \quad \text{である。}$$

20

【0 1 5 6】

式 1 7 乃至 2 1 は、組み合わせられて直接に \hat{k} を k の適切な値にマップするための単独の関数 $\text{map}()$ にすることができる。

【0 1 5 7】

【数 2 4】

$$k = \text{map}(\hat{k}) = h \left\lfloor \frac{\left\lfloor \frac{\hat{k}}{m_h h} \right\rfloor}{m_v} \right\rfloor + \left\lfloor \frac{\text{mod}(\hat{k}, m_h h)}{m_h} \right\rfloor \quad (\text{式 2 2})$$

30

【0 1 5 8】

第 2 パス M B 複雑度変動は、それによって、対応する第 1 パス M B の累積された複雑度を使って、予測されうる。

【0 1 5 9】

【数 2 5】

$$\hat{\gamma}_{\hat{k}} = \sum_{x=0}^{\hat{k}} \hat{g}_x = \sum_{x=0}^{\hat{k}} g_{\text{map}(x)} \quad (\text{式 2 3})$$

40

【0 1 6 0】

ここで x は、単純にインデックスである。例として、第 1 パス M B である A , B , C , D ($k = 0, 1, 2, 3$) が、それぞれ、1 8 4 と 6 6 7 と 1 9 0 と 1 2 2 の第 1 パス複雑値 g_k を有すると仮定する (これらの数値がサンプライメージからとられた標準化された複雑値であり、そして、例としてのみ与えられたものであることに特に言及する) 。

従って、これらの M B に対する累積された第 1 パス複雑度は、

【0 1 6 1】

【数 2 6】

$$\gamma_0 = g_0 = 184$$

$$\gamma_1 = g_0 + g_1 = 851$$

$$\gamma_2 = g_0 + g_1 + g_2 = 1041$$

$$\gamma_3 = g_0 + g_1 + g_2 + g_3 = 1163$$

である。

【 0 1 6 2】

【数 2 7】

10

第2パスMBの場合、式23がa1 ($\hat{k}_{a1} = 0$) と a2 ($\hat{k}_{a2} = 1$) を代入すると、

$$\hat{\gamma}_0 = \sum_{x=0}^0 g_{map(x)} = g_{map(0)} = g_0 = 184$$

$$\hat{\gamma}_1 = \sum_{x=0}^1 g_{map(x)} = g_{map(0)} + g_{map(1)} = g_0 + g_0 = 184 + 184 = 366 \quad \text{となる。}$$

【 0 1 6 3】

20

残りの第2パスMBのマッピングされた累積複雑度は同様に算出される。行 $\hat{j} = 0$ と行 $\hat{j} = 1$ の第2パスMBのマッピングされた累積複雑度が表3に示されている。

【 0 1 6 4】

【表 3】

$\hat{j} = 0$			$\hat{j} = 1$		
MB	\hat{k}	$\hat{\gamma}_{\hat{k}}$	MB	\hat{k}	$\hat{\gamma}_{\hat{k}}$
a1	0	184	a3	8	2510
a2	1	368	a4	9	2694
b1	2	1035	b3	10	3361
b2	3	1702	b4	11	4028
c1	4	1892	c3	12	4218
c2	5	2082	c4	13	4408
d1	6	2204	d3	14	4530
d2	7	2326	d4	15	4652

【 0 1 6 5 】

図 7 は、上記の例における、累積された第 1 パス複雑度とマップされた累積第 2 パス複雑度の間の関係を示している。

【 0 1 6 6 】

ピクチャ j の場合、 T_j がピクチャレベルレートコントロールアルゴリズムによって決定されたときのピクチャ j のエンコードされたバージョンのターゲットサイズである場合には、マクロブロック $k = 0 \cdots N_2 - 1$ のための支配方程式は、ピクチャタイプに応じて、以下のようになる。

【 0 1 6 7 】

【数 2 8】

$$V_k = V_f + a_I \left(\hat{u}_k - T_j \left(\frac{\hat{\gamma}_k}{\hat{\gamma}_{N_2}} \right) \right) \quad (\text{式 2 4 a})$$

【 0 1 6 8 】

【数 2 9】

$$V_k = V_f + a_P \left(\hat{u}_k - T_j \left(\frac{\hat{\gamma}_k}{\hat{\gamma}_{N_2}} \right) \right) \quad (\text{式 2 4 b})$$

10

20

30

40

50

【 0 1 6 9 】

【 数 3 0 】

$$V_k = V_f + a_B \left(\hat{u}_k - T_j \left(\frac{\hat{\gamma}_k}{\hat{\gamma}_{N2}} \right) \right) \quad (\text{式 2 4 c})$$

である。

【 0 1 7 0 】

量子化器 Q_k は、

【 0 1 7 1 】

【 数 3 1 】

$$Q_k = \text{Clip3} \left(0, q^{\max}, \frac{q^{\max} V_k + \frac{1}{2} |B|}{|B|} \right) \quad (\text{式 2 5})$$

によって算出されるが、ここで、

【 0 1 7 2 】

【 数 3 2 】

$$\text{Clip3}(x, y, z) = \begin{cases} x; & z < x \\ y; & z > y \\ z; & \text{otherwise} \end{cases} \quad (\text{式 2 6})$$

である。

【 0 1 7 3 】

定義により、パイプライン遅延のために、エンコーダバッファフルネスの真の値が直ちには利用できないので、 V_f が式 6 によって与えられたエンコーダバッファフルネスの推定値である $B_{i,u}$ のコピーである。MB レベルアルゴリズムは、(シングルプロセッサアーキテクチャの) シーケンシャルエンコーダ設計にも (マルチプロセッサアーキテクチャの) パラレルエンコーダ設計にも適用する。シーケンシャルレートコントロールの場合、パイプライン遅延はゼロとなり V_f は全てのピクチャのエンコード後に直ちに利用可能である。

【 0 1 7 4 】

本発明は、記載された特定の実施の形態に制限されるものではないことは明らかであり、そして、添付の特許請求の範囲に記載された発明の範囲及びその均等の範囲を逸脱することなく改作が行われることも明らかである。例えば、図 2 のデシメーションユニットと第 1 パスエンコーダが、MPEG 2 と H. 264 のような第 2 パスエンコーダによって実現されたものとは別の比較的単純なビデオコード規格を実現するように設計された第 1 パスエンコーダに置換されうことは明らかである。このような場合、ピクチャレベルレートコントロールアルゴリズムは、上記の異なる解像度の場合とは反対に、異なるコード規格により生じたピクチャ複雑度における差異を補償する必要があるでしょう。それとは別に、非圧縮ビデオストリームを受信するエンコーダよりもむしろ、上記と同じアーキテクチャとアルゴリズムがトランスコーディング能力の点で採用されるが、シングルチャンネルトランスコードは第 1 のエンコード規格によりエンコードされたビデオストリームを受信し、上記実施の形態の第 1 パスエンコーダがエンコードされた入力ビデオストリームをデコードして、レートコントロールアルゴリズムによって必要な複雑さの情報を発生する第 1 パスデコーダに置換され、そして、上記の方法でデコードによって集められた複雑さの情報を利用することにより、第 2 パスエンコーダが第 2 のビデオコード規格に

10

20

30

40

50

よりデコードされたビデオストリームをエンコードする。

【図面の簡単な説明】

【0175】

【図1】デュアルパスエンコーダのブロック図である。

【図2】本発明の実施の形態であるシングルチャンネルデュアルパスエンコーダのブロック図である。

【図3】本発明の実施の形態であるデュアルパスエンコーダにおける第1のパスエンコーダと第2のパスエンコーダの間の関係を示すタイミング図である。

【図4】本発明の実施の形態であるマルチチャンネルエンコーダのブロック図である。

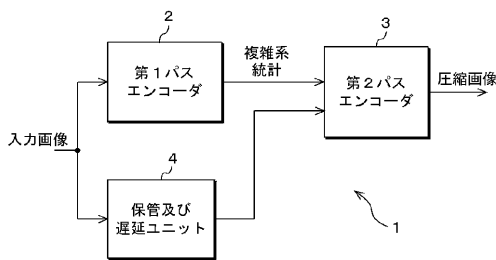
【図5】図4のマルチチャンネルエンコーダに使用される本発明の実施の形態であるシングルチャンネルエンコーダユニットのブロック図である。

【図6】二つの異なるやり方で、単独のピクチャのマクロブロックへの分割を図示する単独のピクチャの二つのバージョンの略図である。

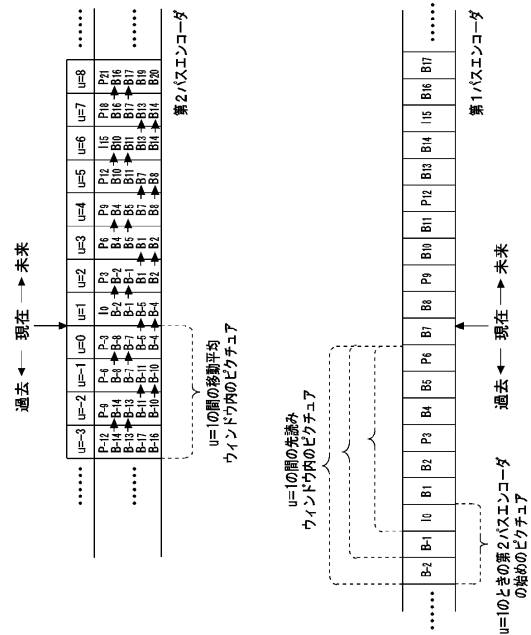
【図7】累積された第1パス複雑度とマップされた累積第2パス複雑度との間の関係を示すグラフ群である。

10

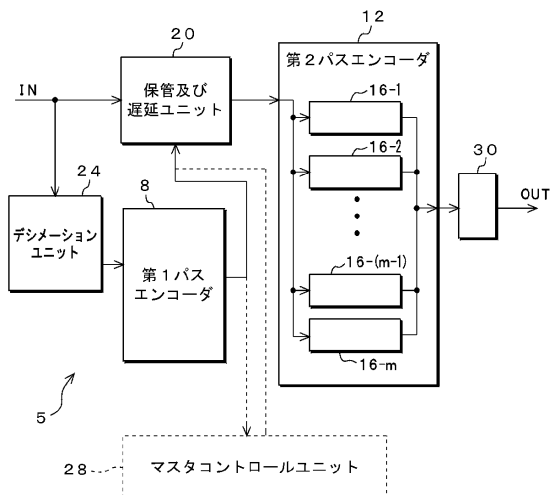
【図1】



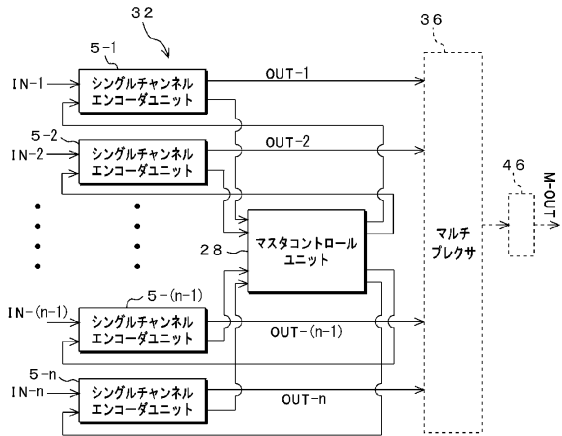
【図3】



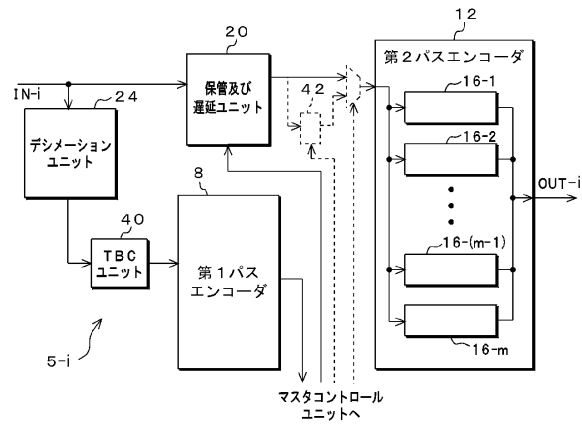
【図2】



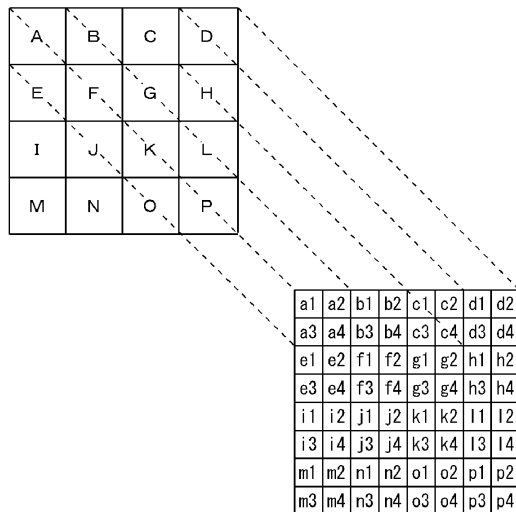
【図 4】



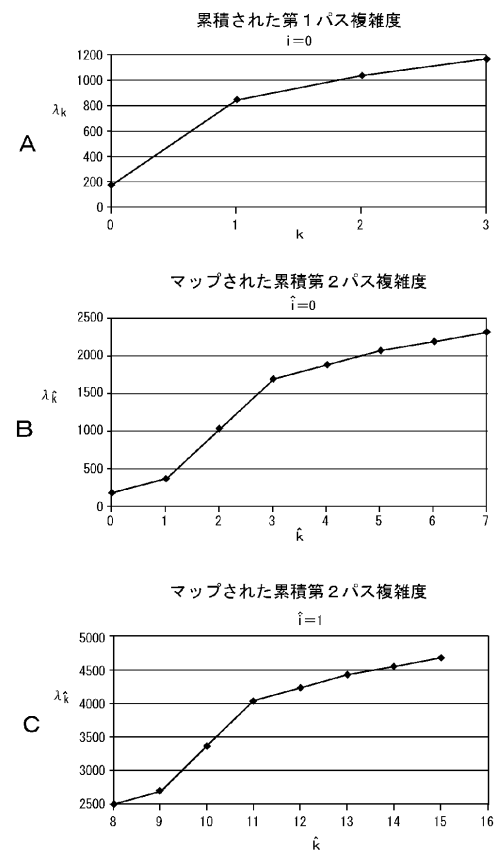
【図 5】



【図 6】



【図 7】



フロントページの続き

- (72)発明者 ユウ・グォヤオ
アメリカ合衆国, オレゴン州 9 7 2 2 9 , ポートランド, ノースウエスト・ミルフォード・スト
リート 1 2 7 1 5
- (72)発明者 シュワルツ・メイヤー・ディ
アメリカ合衆国, オレゴン州 9 7 2 2 5 , ポートランド, サウスウエスト・メルノア・ストリー
ト 1 0 2 6 0
- (72)発明者 ヴァンデューセン・チャールズ・エイチ
アメリカ合衆国, オレゴン州 9 7 0 0 7 , ビーバートン, ピーオーボックス 8 6 8
- (72)発明者 ボルグウォード・ピーター
アメリカ合衆国, オレゴン州 9 7 2 1 9 , ポートランド, サウスウエスト・ネヴァダ・コート
3 9 2 0

審査官 伊東 和重

- (56)参考文献 特開2002-232882(JP, A)
特開平10-191331(JP, A)
特開平07-284096(JP, A)
特開平09-018873(JP, A)
特開平07-264580(JP, A)

- (58)調査した分野(Int.Cl., DB名)
H04N 7/26