

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第3630686号  
(P3630686)

(45) 発行日 平成17年3月16日(2005.3.16)

(24) 登録日 平成16年12月24日(2004.12.24)

(51) Int. Cl.<sup>7</sup>

F I

HO 4 J	3/00	HO 4 J	3/00	M
HO 4 B	14/04	HO 4 B	14/04	Z
HO 4 J	3/16	HO 4 J	3/16	Z
HO 4 N	7/08	HO 4 N	7/13	Z
HO 4 N	7/081	HO 4 N	7/08	Z

請求項の数 3 (全 15 頁) 最終頁に続く

(21) 出願番号 特願平7-527604  
 (86) (22) 出願日 平成6年4月22日(1994.4.22)  
 (65) 公表番号 特表平9-512399  
 (43) 公表日 平成9年12月9日(1997.12.9)  
 (86) 国際出願番号 PCT/US1994/004420  
 (87) 国際公開番号 W01995/029540  
 (87) 国際公開日 平成7年11月2日(1995.11.2)  
 審査請求日 平成13年4月20日(2001.4.20)

(73) 特許権者  
 トムソン コンシューマ エレクトロニクス インコーポレイテッド  
 アメリカ合衆国 インディアナ州 インディアナポリス ノース・メリディアン・ストリート 10330

(74) 代理人  
 弁理士 谷 義一

(74) 代理人  
 弁理士 阿部 和夫

(72) 発明者  
 オヅカン, メヘメット, ケマル  
 アメリカ合衆国 46032 インディアナ州 カーメル ラーク ドライブ 420 アpartment 7

最終頁に続く

(54) 【発明の名称】 パラメータ・サンプリング装置

(57) 【特許請求の範囲】

【請求項1】

パラメータ・サンプリング装置であって、  
 各データ信号がシーケンスに並んだデータ・グループを含み、前記データ信号のすべての中の各グループは同じ一定の、あらかじめ決められた持続時間を持ち、シーケンスに並んだデータ・ブロックを含んでいる複数のデータ信号ソースと、  
 各々が、該データ信号のそれぞれ1つにตอบสนองして、該データ信号のそれぞれ1つの中の前記シーケンスのブロックの各々と関連するパラメータを表す値をもつシーケンスの信号を出力する複数のパラメータ決定回路と、  
 各々が、前記パラメータを表す信号のそれぞれ1つにตอบสนองして、該データ信号の前記それぞれ1つの中のシーケンスのブロックに対するパラメータ値の合計を表す各信号を供給する複数のアキュムレータと、  
 前記複数のアキュムレータに結合されていて、該アキュムレータの各々からの前記合計をデータ・グループの前記持続時間内の時間で同時にサンプリングするデータ・サンブラとを備え、  
 前記データ信号ソースの各々はビデオ信号を発生する手段を含み、各データ・グループはあらかじめ決められた個数のシーケンスのビデオ・フレームを含み、前記データ・ブロックは前記ビデオ・フレームからなり、前記持続時間は前記グループの1つにおけるシーケンスのビデオ・フレームのあらかじめ決められた個数の持続時間と等しく、  
 前記パラメータ決定回路の各々は前記ビデオ信号の前記1つの各フレームの符号化複雑度

10

20

を決定する回路を含んでいることを特徴とする装置。

【請求項2】

請求項1に記載の装置において、前記複数のアキュムレータと前記データ・サンブラは、それぞれのパラメータ決定回路に結合された複数の入力端子をもつコンピュータ・システムに構成されていることを特徴とする装置。

【請求項3】

請求項1に記載の装置において、

各データ・グループは、あらかじめ決められた個数の前記シーケンスのデータ・ブロックを含んでおり、

前記複数のアキュムレータの各々は、

各データ・グループに含まれる前記あらかじめ決められた個数のシーケンスのデータ・ブロックの信号パラメータ値を、最新のものから最も古いものの時間順にストアしておくメモリと、

累積パラメータ値としての前記合計を保持している回路と、

信号パラメータを表す信号の値を該累積パラメータ値に加算し、前記最も古い信号パラメータ値を該累積パラメータ値から減算する回路と、

該最も古い信号パラメータ値を前記メモリから破棄し、前記信号パラメータを表す信号の値を最新の信号パラメータ値として前記メモリにストアする回路とを含んでいる

ことを特徴とする装置。

【発明の詳細な説明】

本発明は、複数のデータ信号の各々の信号のパラメータを表している信号をサンプリングする装置に関する。

発明の背景

複数のデータ信号の各々の信号の信号パラメータ（例えば、符号化複雑度）をサンプリングすることが必要なことがある。すべてのデータ信号が相互に同期している場合には、これは、測定装置により、データ信号に回答してデータ信号の信号パラメータを表すそれぞれの信号を生成し、これらのパラメータを表す信号を適切な時刻にサンプリングすることによって行われるだけである。しかるに、データ信号が相互に同期していないが、データ信号のすべての処理を、それぞれの信号パラメータの相対値に基づいて制御することが望ましい場合には、信号パラメータをデータ信号のすべてで実質的に同時にサンプリングすることが必要である。その場合、信号パラメータのサンプリングが問題となっている。

あるシステム例では、それぞれのチャンネル（これらはテレビジョン・ネットワーク供給（television network feed）やテレビジョン・ステーション、その他のビデオ・ソースである場合がある）からの複数のビデオ（映像）信号は衛星リンクを経由して伝送され、消費者の家庭のそれぞれのテレビジョン受信装置にブロードキャストされている。各ビデオ信号は、シーケンスに並んだデータ・グループからなる信号に符号化され、各グループは一定数の連続ビデオ・ピクチャまたはフレームを表すデータを含んでおり、ピクチャ・グループ（group of pictures - GOP）と呼ばれている。データ信号のすべてにおける各グループは、同じ一定の、あらかじめ決められた持続時間（グループ内のビデオ・ピクチャまたはフレームの個数の持続時間に等しい）をもっているが、異なるビデオ信号内のグループのタイミングは同期していない。

上記のようなシステムでは、異なるチャンネルからのデータは、ヘッドエンド・ステーション（head end station）で単一のデータ・ストリームに結合、つまり、多重化されている。多重化データ・ストリームは、ワイヤや光ファイバ、無線リンクなどの伝送リンクを経由してバックエンド・ステーション（back end station）に伝送され、そこで多重化データ・ストリームからのデータのチャンネルが分離、つまり、デマルチプレクスされ、目的の受信人へ供給されている。衛星リンクの例としては、毎秒24メガビット（Mbps）の伝送能力をもつデジタル伝送経路がある。この種のリンクの効率と利用効率を最大限にするためには、複数のビデオ信号がリンクを共有する必要がある。例えば、上記の衛星伝送リンクを少なくとも6つのビデオ信号チャンネル間で共有することが望ましい場合がある。

10

20

30

40

50

## 発明の概要

衛星リンク経由で搬送される多重化信号の総ビットレートを、チャンネルで伝送されるイメージ（画像など）の現在の符号化複雑度（coding complexity）に基づいて、異なるチャンネルに動的に割り当てることが提案されている。すべてのチャンネル用に現在伝送されているイメージの符号化複雑度はGOP単位で計算され、実質的に同時にサンプリングされている。伝送リンクの総ビットレートの割り分は、各チャンネルの符号化複雑度とすべてのチャンネルの総符号化複雑度との関係に基づいて、各チャンネルに割り当てられている。各チャンネルの次のGOPの符号化はそのチャンネルに動的に割り当てられたビットレートに基づいて行われる。

本願の発明者は、各GOPが複数のシーケンスのデータ・ブロックを含み、各ブロックが伝送イメージの1つのフレームまたはピクチャを表していること、および複雑度を表す信号が各ブロックごとに生成できることを認識した。さらに、発明者は、GOPの複雑度を表している信号は、GOP内の各ブロック（つまり、フレームまたはピクチャ）の複雑度を表す値を累算することによって生成できることを認識した。また、発明者は、GOP複雑度値を生成するとき、各ビデオ信号内の同数のブロック（つまり、フレームまたはピクチャ）の複雑度値が累算され、アキュムレータが各サンプリングのあとでリセットされる限り、これらのブロックがすべて同一GOP内に属していなければ、GOP複雑度値に不利に影響しないことも認識した。この同一原理は、一般的に、これらの特性をもつどのデータ・パラメータ・サンプリング・システムにも応用可能である。

本発明の原理によれば、複数のデータ信号の中の信号パラメータをサンプリングする装置は複数のデータ信号ソース（発生源）を含んでおり、各データ信号はシーケンスに並んだデータ・グループを含み、各データ信号内の各データ・グループは同じ一定の、あらかじめ決められた持続時間をもち、シーケンスに並んだデータ・ブロックを含んでいる。複数のパラメータ決定回路（parameter determining circuit）の各々は、データ信号のそれぞれ1つの応答し、それぞれのデータ信号内のシーケンス・データ・ブロックの各々の信号パラメータを表す値をもつシーケンスに並んだ信号を出力する。複数のアキュムレータの各々は、パラメータを表す信号のそれぞれ1つに応答し、データ信号の累積パラメータ値を表す信号を出力する。データ・サンプラ（data sampler）は、複数のアキュムレータに応答して、アキュムレータのすべてからの累積パラメータを表す信号を、一定のあらかじめ決められた持続時間に等しい時間インターバルで実質的に同時にサンプリングする。本発明によるシステムは、複数のデータ信号からの信号パラメータを実質的に同時にサンプリングすることができ、しかも、データ信号が同期していることを必要とせず、単に一定のあらかじめ決められた持続時間をもつデータ・グループだけからなることを必要としている。上述の例では、本発明によるサンプリング装置を使用することにより、瞬間の符号化複雑度（coding complexity）に基づいた複数のビデオ信号中の多重化データ・ストリームにおけるビットレートのダイナミック割当てが可能となる。

### 【図面の簡単な説明】

図1は、本発明によるマルチプレクサ・システムを示すブロック図である。

図2は、図1に示すマルチプレクサ・システムで使用できるチャンネル・プロセッサを示すブロック図である。

図3は、図2に示すチャンネル・プロセッサで使用できるMPEGエンコーダ（符号器）の一部を示すブロック図である。

図4は、図1に示すマルチプレクサ・システムで使用できるビットレート・アロケータを示すブロック図である。

図5は、図2に示すチャンネル・プロセッサで使用できる複雑度アナライザを示す詳細ブロック図である。

図6、図7、および図8は複雑度情報のサンプリングを示すタイミング図である。

### 好ましい実施例の詳細な説明

図1は、本発明に従ったマルチプレクサ・システムを示すブロック図である。図1において、すべての信号経路は単一の信号ラインで示されている。しかし、この分野の精通者な

10

20

30

40

50

らば理解されるように、図示の信号経路はマルチビット・デジタル信号をパラレル（並列）で伝達することも、シリアル（直列）で伝達することも可能である。パラレルで伝達する場合は、信号経路は複数の信号ラインで構成されることになり、シリアルで伝達する場合は、信号経路は単一データ・ラインにすることも、および/またはデータ・クロック信号ラインを含めることも可能である。本発明を理解することと密接な関係のない他の制御信号経路とクロック信号経路は、図を簡略化するために図には示されていない。

図1に示すように、複数の入力端子5はビデオ信号（CHANNEL 1 - CHANNEL K）のソース（図示せず）に結合され、これらの信号はデータ・リンクを経由して一緒に伝送されるものである。複数の入力端子5は、対応する複数のチャンネル・プロセッサ10のそれぞれのデータ入力端子に結合されている。複数のチャンネル・プロセッサ10のそれぞれのデータ出力端子はマルチプレクサ（MUX）20の対応するデータ入力端子1 - Kに結合されている。マルチプレクサ20のデータ出力端子はマルチプレクサ・システムの出力端子15に結合されている。出力端子15は、多重化されたデータ・ストリーム（MUX'ED DATA）を送信リンク経路で伝送するために利用回路（図示せず）に結合されている。

複数のチャンネル・プロセッサ10の各々はさらに、複雑度出力端子と制御入力端子も備えている。複数のチャンネル・プロセッサの各々のそれぞれの複雑度出力端子はビットレート・アロケータ（bit rate allocator）30の対応する複雑度入力端子に結合されており、ビットレート・アロケータ30のそれぞれの割当量出力端子（quota output terminal）は複数のチャンネル・プロセッサ10の対応する制御入力端子に結合されている。

動作時には、各チャンネル・プロセッサは、次の割当量周期（quota period）の間に割り当てられたビットレートを表す信号をその制御入力端子から受信する。次に、チャンネル・プロセッサは、次の割当量周期の間に、そのデータ入力端子に現れた信号を割り当てられたビットレートでデジタル符号化信号に符号化する。この符号化データ信号はマルチプレクサ20の対応する入力端子に入力される。マルチプレクサ20は公知のように動作し、すべてのチャンネル・プロセッサからの信号を結合して多重化データ・ストリームにする。多重化データ・ストリームは、データ・リンクを構成する回路に入力され、これも公知のように伝送される。

符号化プロセス期間に、チャンネル・プロセッサ10は符号化される信号の符号化複雑度を表す信号をその複雑度出力端子から発生する。ビットレート・アロケータ30はチャンネル・プロセッサ10の複雑度出力端子から信号を受信し、複雑度信号のすべてに基づいて、次の割当量周期の間のビットレート割当量を複数のチャンネル・プロセッサ10間で動的に調整する。好ましい実施例では、複雑な信号には、複雑でない信号よりも相対的に高いビットレートが動的に割り当てられるようになっている。ビデオ信号の複雑度を判断し、その複雑度に基づいてビットレートを割り当てるいくつかの方法については、以下で説明する。

図2は、図1に示すマルチプレクサ・システムで使用できるチャンネル・プロセッサを示すブロック図である。図2において、図1に示すエレメントと類似のエレメントは同一参照符号で示し、以下では詳しく説明することは省略する。図2に示すように、データ入力端子5はビデオ信号ソース（図示せず）に結合されている。データ入力端子5は一定ビットレート・エンコーダ（constant bit rate encoder - CBR）14のデータ入力端子と複雑度アナライザ（complexity analyzer）16に結合されている。CBRエンコーダ14のデータ出力端子はマルチプレクサ（MUX）20（図1）の入力端子に結合されている。チャンネル・プロセッサ10の制御入力端子（CONTROL）はCBRエンコーダ14の割当量入力端子Qに結合されている。複雑度アナライザ16の出力端子はチャンネル・プロセッサ10の複雑度出力端子（COMPLEXITY）に結合されている。

動作時には、複雑度アナライザ16はデータ入力端子5のビデオ信号の複雑度を分析する。複雑度アナライザ16の出力端子に生成される信号は、入力信号の複雑度を表している。複雑度を表す信号はビットレート・アロケータ30（図1）に入力される。この複雑度信号（および他のチャンネル・プロセッサ10の複雑度信号）にตอบสนองして、ビットレート・アロケータ30は、そのチャンネル・プロセッサ10（および他のチャンネル・プロセッサ10）の制御入力端子（CONTROL）へ信号を供給し、この信号はそのチャンネル・プロセッサ10に割り当てら

10

20

30

40

50

れたビットレートを表している。CBRエンコーダ14は、そのデータ入力端子とデータ出力端子間にデータ経路をもち、一定ビットレートで符号化された出力信号を出力する。一定ビットレートは、ビットレート・アロケータ30からのチャンネル・プロセッサ10の制御入力端子（CONTROL）からの、割当量入力端子Qにおける信号にตอบสนองしてセットされる。CBRエンコーダ14内の回路は、その分析を行う際に複雑度アナライザ16によって利用することも可能である。そのような場合には、図2に破線で示すように、データはCBRエンコーダ14内部から直接に複雑度アナライザ16に渡される。CBRエンコーダ14からのこのようにデータは、入力端子5からのデータを補足することも、そのデータと完全に入れ替わることできるが、後者の場合には、複雑度アナライザとデータ入力端子5とは直接に接続されない。

10

好ましい実施例では、各CBRエンコーダ14は、動画専門家グループ（Moving Pictures Expert Group - MPEG）が公表した標準に従ってビデオ信号を圧縮・符号化するエンコーダであり、MPEGエンコーダと呼ばれている。図3は、MPEGエンコーダ14の一部を示すブロック図である。MPEGエンコーダ14の公知コンポーネントは以下で詳しく説明することは省略する。MPEGエンコーダには他のエレメントもあるが、これらは本発明を理解することとは無関係であるので、図面を簡略化するために図示されていない。

図3に示すように、MPEGエンコーダ14のデータ入力端子5（DATA IN）は圧縮・符号化しようとするビデオ信号のソース（図示せず）に結合されている。入力端子5はフレーム・バッファ41の入力端子に結合されている。フレーム・バッファ41は複数のフレーム周期バッファまたはディレイラインと、異なっているが時間的に隣り合うフレームまたはピクチャの部分を表す、それぞれの信号を出力するための複数の出力端子とを備えている。フレーム・バッファ41の複数の出力端子は動き予測器（motion estimator）42の対応する入力端子に結合されている。動き予測器の出力端子は離散コサイン変換（discrete cosine transform - DCT）回路43に結合されている。DCT回路43の出力端子は可変量子化回路（variable quantizer - Qu）回路46のデータ入力端子に結合されている。可変量子化回路46の出力端子は可変長コーダ（variable length coder - VLC）47の入力端子に結合されている。VLC47の出力端子は出力バッファ48の入力端子に結合されている。出力バッファ48のデータ出力端子はMPEGエンコーダ14のデータ出力端子（DATA OUT）に結合されている。MPEGエンコーダ14のデータ出力端子（DATA OUT）はマルチプレクサ20（図1の）の対応する入力端子に結合されている。

20

30

出力バッファ48のステータス出力端子はビットレート・レギュレータ49のステータス入力端子に結合されている。ビットレート・レギュレータ49の制御出力端子は可変量子化器（variable quantizer）46の制御入力端子に結合されている。MPEGエンコーダ14の割当量入力端子Qはビットレート・アロケータ30の対応する割当量出力端子に結合されている。MPEGエンコーダ14の割当量入力端子Qはレギュレータ49の制御入力端子に結合されている。動作時には、MPEGエンコーダ14は公知のように動作して、その入力端子に現れたビデオ信号を、そのQ入力端子に現れた信号によって決まるビットレートで次の割当量周期の間に圧縮・符号化する。以下の例では、MPEGエンコーダが12ピクチャまたはフレームからなるグループ（GOP）に分割されたビデオ信号を符号化する場合について説明する。なお、当然に理解されるように、GOPにおけるピクチャまたはフレーム数は可変である。また、以下の例では、各MPEGエンコーダ用のビットレート割当ては各GOPごとに一度更新されるものと想定している。つまり、割当量周期がGOP期間であるものと想定している。なお、この場合も当然に理解されるように、割当量周期は異なる場合があり、時間の経過と共に変化する場合もある。

40

フレーム・バッファ41は、例示のGOP内の12フレームのうち現在符号化されている部分を表しており、動き予測を行うために必要なデータを受信し、ストアする。その方法については、以下で説明する。このデータは動き予測器42に渡される。好ましい実施例では、12フレームまたはピクチャの最初の1つは参照フレーム（Iフレーム）として使用され、動き予測器を経由してDCT回路43へ渡される。残りのフレームについては、各ピクチャまたはフレームに含まれる複数の16ピクセル×16ライン・ブロックの各々ごとに、動きベクト

50

ル (motion vector) が先行フレーム (Pフレーム) 単独から、あるいは先行フレームと後続フレーム (Bフレーム) の両方からインタポレートされたものから、動き予測器42で生成される。なお、このブロックはMPEG標準ドキュメントではマクロブロック (macroblock) と名づけられている。上述したように、フレーム・バッファ41は、動き予測器が先行フレームまたは先行フレームと後続フレームからインタポレートしたのものから予測を行うとき必要になるデータを格納している。特定フレームの生成された動きベクトルは、そのあと、予測しようとするフレームに含まれる実際のデータと比較され、動き差信号 (motion difference signal) が生成され、DCT回路43に渡される。

DCT回路43では、Iフレームからの空間的データの16ピクセル×16ライン・マクロブロックと、およびPフレームとBフレームとからの動き差信号とは、6個の8ピクセル×8ライン・ブロック (4個のルミナンス・ブロック (luminance block) と、サブサンプリングされた2個のクロミナンス・ブロック (chrominance block) ) に分割される。なお、以下の説明では、これらのブロックはMPEG標準ドキュメントに従ってマクロブロックと呼ぶことにする。離散コサイン変換が各マクロブロックについて行われる。その結果得られたDCT係数の8×8ブロックは可変量子化器46に渡される。係数の8×8ブロックは量子化され、ジグザグ順にスキャンされ、VLC47に渡される。量子化されたDCT係数、およびGOPを表す他のサイド情報 (符号化GOPのパラメータに関するもの) はVLC47でランレングス符号化 (run-length coding) によって符号化され、出力バッファ48に渡される。

VLC47の出力ビットレートを制御し、もってMPEGエンコーダ14用に割り当てられた一定ビットレートを保つ最も直接的方法は、可変量子化器46でDCT係数の各ブロックを量子化する

20

とき使用される量子化レベル数 (言い換えれば、量子化ステップ・サイズ) を制御することであることは知られている。ビットレート・レギュレータ49から可変量子化器46に渡された制御信号Qはこの制御機能を実行する。ビットレート・アロケータ30 (図1) からの連続的ビットレート割当量更新信号Q相互間の期間である割当量周期内に、ビットレート・レギュレータ49は制御信号を可変量子化器46へ送り、この制御信号はGOP内の各16×16マクロブロックが量子化されるレベル数を変更して、割り当てられたビットレートをその割当量周期の間維持するようにするが、これは公知である。

この例では、ビットレート・レギュレータ49のビットレート割当ては、各チャンネルにおけるビデオ信号の符号化複雑度値に応じて、各GOP期間ごとに変化するが、これについては以下で説明する。

30

好ましい実施例では、ビットレート・アロケータ30 (図1) は、複数のチャンネル・プロセッサ10内の種々回路コンポーネントに結合された接続個所をもつコンピュータ・システムである。図4はビットレート・アロケータ30を構成するハードウェアを示すブロック図である。図4に示すように、マイクロプロセッサ (μP) 31は、コンピュータ・システム・バス35を介して読み/書きメモリ (RAM) 32、リードオンリメモリ (ROM) 33および入出力 (I/O) コントローラ34に結合されている。コンピュータ・システムには、大量記憶デバイスやユーザ端末などの他のコンポーネントもあるが、これらは図面を簡単にするために図示されていない。入出力コントローラ34は、複数のチャンネル・プロセッサ10 (図1) の対応する複雑度出力端子に結合された複数の入力端子 (COMPLEXITY) と、複数のチャンネル・プロセッサ10の対応する割当量入力端子に結合された複数の出力端子 (QUOTA) とをも

40

っている。

マイクロプロセッサ31、RAM32、ROM33および入出力コントローラ34は公知のようにコンピュータ・システムとして動作して、ROM33に格納されたプログラムを実行し、データをRAM33にストアし、そこからデータを取り出し、入出力コントローラ34に接続されたデバイスとの間でデータを送受信する。複数のチャンネル・プロセッサ10 (図1) で符号化されるビデオ信号の現在の符号化複雑度を表しているデータは、これらのチャンネル・プロセッサ34の対応する出力端子からCOMPLEXITY入力端子を経由して入出力コントローラ34で受信される。なお、これについては以下で説明する。このデータを受信したことは、例えば、ポーリングや割込みなどの公知の方法でマイクロプロセッサ31に通知される。マイクロプロセッサ31はこれらの信号を入出力コントローラ34からコンピュータ・システム・バス35経由

50

で取り出し、エンコーダの各々ごとに次の割当量周期の間のビット割当量を判断し、その割当量を表している信号を、次の割当量周期の間にQUOTA出力端子を經由して複数のチャネル・プロセッサ10へ渡す。

MPEGエンコーダ14(図3)によって符号化されるビデオ信号の符号化複雑度を判断する好ましい方法では、各 $16 \times 16$ マクロブロックの量子化スケール係数(quantization scale factor:  $Q_{MB}$ と呼ぶ)とそのマクロブロックを符号化するために使用されたビット数( $T_{MB}$ と呼ぶ)が、GOPの各ピクチャまたはフレームに含まれるすべてのマクロブロックについて使用されている。図5はMPEGエンコーダ14(図3)のビットレート・レギュレータ49と、符号化複雑度を表す信号をこの方法に従って生成する複雑度アナライザ16(図2)とを示すブロック図である。種々のクロック信号と制御信号は図面を簡略化するために図6には示されていない。なお、どのような信号が必要であるか、さらに、これらの信号の必要なタイミングと電圧特性は自明であるので、説明は省略する。

図5に示す複雑度アナライザ16は、図2に破線で示すようにCBRエンコーダ14からの情報だけを利用する複雑度アナライザの例である。図5に示すように、ビットレート・レギュレータ49は出力バッファ48(図3)のステータス出力端子に結合されたステータス入力端子 $T_{MB}$ をもっている。ビットレート・レギュレータ49の制御出力端子 $Q_{MB}$ は可変量子化器46(図3)の制御出力端子に結合されている。レギュレータ49は、さらに、ビットレート・アロケータ30(図1)の対応する割当量出力端子に結合された制御出力端子(Q)をもっている。

ビットレート・レギュレータ49のステータス入力端子 $T_{MB}$ は第1加算器92の第1入力端子にも結合されている。第1加算器92の出力端子は第1ラッチ93の入力端子に結合されている。第1ラッチ93の出力端子は乗算器94の第1入力端子と第1加算器92の第2入力端子に結合されている。乗算器94の出力端子は第2ラッチ95の入力端子に結合されている。第2ラッチ95の出力端子は符号化複雑度出力端子 $X_{pic}$ に結合されている。複雑度出力端子 $X_{pic}$ はビットレート・アロケータ30(図1)の対応する複雑度入力端子に結合されている。ビットレート・レギュレータ49の制御出力端子 $Q_{MB}$ は第2加算器96の第1入力端子にも結合されている。第2加算器96の出力端子は第3ラッチ97の入力端子に結合されている。第3ラッチ97の出力端子は除算器(divider)98の分子入力端子Nと第2加算器96の第2入力端子に結合されている。除算器98の出力端子は乗算器94の第2入力端子に結合されている。レジスタ99は除算器98の分母入力端子Dに結合された出力端子をもっている。

動作時には、各マクロブロックごとに、ビットレート・レギュレータ49は可変量子化器46の量子化スケール係数信号 $Q_{MB}$ を、現在のビットレート割当量と先行ピクチャを符号化するために使用されたビット数に基づいて公知のように生成し、そのあと、そのマクロブロックを符号化するために使用されたビット数 $T_{MB}$ を示している信号を出力バッファ48から受信する。可変量子化器46(図3)は、各マクロブロックに含まれるDCT係数を量子化スケール係数 $Q_{MB}$ に従って量子化する。量子化スケール係数 $Q_{MB}$ は量子化ステップ・サイズ、つまり、各量子化レベルにおいてDCT係数が全ダイナミックレンジに占めるパーセンテージを表している。 $Q_{MB}$ の値が高いときは、大きな量子化ステップ・サイズが存在し、従って、量子化レベルが少ないことを意味する。逆に、 $Q_{MB}$ の値が低いときは、小さな量子化ステップ・サイズが存在し、従って、量子化レベルが多いことを意味する。好ましい実施例では、 $Q_{MB}$ は5ビット整数(1と31の間の値をもつ)になっている。

次に、完全ピクチャまたはフレームに含まれるすべてのマクロブロックの平均量子化スケール係数(

$Q_{pic}$

と呼ぶ)が以下のように計算される。各フレームまたはピクチャの始まりで、ラッチ93と97はクリア信号(図示せず)にตอบสนองしてゼロにクリアされる。第2加算器96と第3ラッチ97の組合せはアキュムレータとして動作して、ビットレート・レギュレータ49からのマクロブロック量子化スケール係数 $Q_{MB}$ を連続的に加算する。これと同時に、第1加算器92と第1ラッチ93の組合せはアキュムレータとして動作して、フレームまたはピクチャを符号化するためにそれまでに使用されたビット数を連続的に加算する。

10

20

30

40

50

フレームまたはピクチャ内のマクロブロックのすべて（その数は $N_{MB}$ と呼ぶ）が処理されたとき、ラッチ97にはビットレート・レギュレータ49から得たマクロブロック量子化スケール係数 $Q_{MB}$ のすべての和が入っており、ラッチ93にはピクチャまたはフレームを符号化するために使用されたビットの全ての和 $T_{pic}$ が入っている。除算器98はピクチャまたはフレームに含まれる全マクロブロック量子化スケール係数 $Q_{MB}$ の総和を、ピクチャまたはフレームに含まれるマクロブロック数 $N_{MB}$ で除した商を出力する。この商は、そのフレームまたはピクチャの平均量子化スケール係数

$$Q_{pic}$$

である。乗算器94は

$$Q_{pic}$$

と $T_{pic}$ の積を出力し、これはそのピクチャの符号化複雑度（coding complexity: $X_{pic}$ と呼ぶ）である。つまり、

$$X_{pic} = T_{pic} \cdot Q_{pic}$$

である。ピクチャまたはフレームの終わりで、符号化複雑度信号 $X_{pic}$ はクロック信号（図示せず）に 응답して第2ラッチ95符号ラッチされる。上述したサイクルは、符号化されるビデオ信号内の各フレームまたはピクチャごとに繰り返される。

次に、符号化複雑度 $X_{pic}$ はラッチ95からビットレート・アロケータ30（図4）の入出力コントローラ34の複雑度入力端子へ送られ、そこで残余の処理が行われて、GOPの符号化複雑度が得られる。GOPの符号化複雑度（ $X_{GOP}$ と呼ぶ）はそのGOPに含まれる全ピクチャの $X_{pic}$ の総和である（式（1）を参照）。 $\mu P31$ はアキュムレータの働きをして、各 $X_{pic}$ 値を入出力コントローラ34から取り出し、GOPに含まれる全フレームまたはピクチャにわたってその総和をとる。

$$X_{GOP} = \sum_{\text{all pics in GOP}} T_{pic} \cdot Q_{pic} = \sum_{\text{all pics in GOP}} X_{pic} \quad (1)$$

GOP内のフレームまたはピクチャ数（ $N$ と呼ぶ）は一般に一定のままになっている。 $N$ が一定である間に、 $X_{GOP}$ は、最新ピクチャの符号化複雑度値 $X_{pic}$ を加算し、その符号化複雑度値をGOPの最古ピクチャから減算することによって、スライディング・ウィンドウ単位で計算される。この場合には、 $X_{GOP}$ の更新値は各フレームまたはピクチャのあとで得られる。なお、 $N$ は変化することもある。 $N$ が変化したときは、新たに定義されたGOPの $X_{GOP}$ は、式（1）に示すように、新たに定義されたGOP内の新しい数の先行ピクチャからの符号化複雑度値 $X_{pic}$ の総和をとることによって計算しなければならない。

上述したように、異なるチャンネルは異なるフレームまたはピクチャ・レートで動作することが起こり得る。例えば、標準ビデオ・フレーム・レート（米国の場合）は毎秒29.97フレームであり、フィルムイメージでは毎秒24フレーム、カートーンでは毎秒15フレームである。また、チャンネルが異なると、GOPに含まれるピクチャまたはフレーム数が異なることも起こり得る。従って、チャンネルが異なると、GOP時間周期が異なることが起こり得る。そのような条件下でチャンネルにビットを正確に割り当てるためには、そのようなことが起こったときの複数のチャンネルのGOP符号化複雑度値は、各チャンネルについて式（1）で得たGOP複雑度値を、そのチャンネルのGOP時間周期（ $GOP_{time}$ と呼ぶ）で除することにより、ビットレート・アロケータ30で時間正規化される（式（2）を参照）。正規化されたGOP符号化複雑度値（ $X_{norm\_GOP}$ と呼ぶ）は、

$$X_{norm\_GOP} = \frac{X_{GOP}}{GOP_{time}} \quad (2)$$

ビットを異なるチャンネル間で割り当てるために使用される。このようなシステムで複雑度値をサンプリングし、割当量を生成するタイミングについては、以下で詳しく説明する。図5に戻って説明すると、上述したように、各マクロブロックごとに、ビットレート・レギュレータ49は可変量子化器46に対して量子化スケール係数信号 $Q_{MB}$ を生成し、そのあと、そのマクロブロックを符号化するために使用されたビット数 $T_{MB}$ を示す信号を出力バッファ48から受信する。これらの信号は、別の方法として、ビットレート・アロケータ30（

10

20

30

40

50

図4)内の入出力コントローラ34へ直接に送ることも可能である。そのあと、 $\mu P31$ は該当の符号化複雑度測定値を内部で計算することができる(式(1)または式(1)と(2)から)。

さらに、伝送を単純化するために、各ピクチャの符号化複雑度値 $X_{pic}$ をスケールリングすることが可能である。好ましい実施例では、この値は、乗算器94のあとで8ビット数にスケールリングされている。このスケールリングされた値はビットレート・アロケータ30(図4)に渡される。コンピュータ・システムは、Nが変化した場合に符号化複雑度値の再計算を可能にするといった他の理由で、ピクチャ複雑度値 $X_{pic}$ のファイルを、例えば、大量記憶デバイス(図示せず)に保管しておくことが望ましい場合もある。一時間の8ビット $X_{pic}$ 値をストアするには、標準ビデオでは108キロバイト(KB)、フィルムでは86KBが必要になる。

以下の説明では、 $X^i$ は、i番目のチャンネル・プロセッサからの $X_{GOP}$ (すべてのチャンネルが同じGOP時間周期をもつ場合)または $X_{norm\_GOP}$ のうち現在利用可能であって、該当するものを表している。ビットレート・アロケータ30(図1)は、次の割当量周期の間の伝送リンクにおける利用可能ビット数の割当てを表しているそれぞれの割当量(Q)信号を、複数のチャンネル・プロセッサ10を構成するK個のチャンネル・プロセッサのすべてからの符号化複雑度値 $X^i$ に基づいて生成する。マルチプレクサ20(図1)の出力端子からの、あらかじめ決められた伝送リンク・ビットレート(Rと呼ぶ)は、i番目のプロセッサが $R^i$ と名づけられたビットレート割当てを受け取るように、複数のチャンネル・プロセッサ10間で割り当てられる。

伝送リンクにおけるビットレートを異なるチャンネルに割り当てる1つの方法は、複数のチャンネル・プロセッサ10(図1)のすべてについて(上述したように、スライディング・ウィンドウ単位で)先行GOP周期の現在利用可能な符号化複雑度 $X^i$ に基づいたリニア割当て(linear allocation)である。この方法では、各プロセッサiは、そのエンコードの符号化複雑度 $X^i$ がすべてのエンコードの総符号化複雑度と係わりがあるので、総ビット容量Rの同一割当量 $R^i$ を受け取る(式(3)を参照)。

$$R^i = \frac{X^i}{\sum_{j=1}^K X^j} R \quad (3)$$

なお、下限ビットレート割当て以下になると、再現イメージの品質が急激に低下することが分かっている。さらに、図示の実施例では、次の割当量周期の間のビットレート割当ては、先行GOPからの複雑度測定値に依存している。従って、シーンが単純なイメージから複雑なイメージに変化する場合は、新しいシーンの割当てが先行の単純シーンに基づいているので、新しい複雑なシーンを符号化するために割り当てられたビット数が不足することになる。

伝送リンクにおけるビットレートを異なるチャンネルに割り当てる別の方法によれば、各エンコードiへの最小限ビットレート割当て $RG^i$ が保証され、式(3)に示すように、残りのビットは線形的に割り当てられる(式(4)を参照)。

$$R^i = RG^i + \frac{X^i}{\sum_{j=1}^K X^j} [R - \sum_{j=1}^K RG^j] \quad (4)$$

各チャンネルに保証される最小限ビットレートは、そのチャンネル経由で伝送されるビデオの予想される総複雑度および/またはビデオ信号のプロバイダに対するチャンネルの値段に応じて異なる場合がある。

伝送リンクにおけるビットを異なるチャンネルに割り当てる、さらに別の方法によれば、各エンコードiに重み係数 $P^i$ を用意し、重み係数 $P^i$ で重み付けされた符号化複雑度値 $X^i$ に応じてビットを比例的に割り当てている(式(5)を参照)。

$$R^i = \frac{P^i X^i}{\sum_{j=1}^K P^j X^j} R \quad (5)$$

式(4)の最小限保証割当て方法と同じように、重み係数 $P^i$ はチャネル経由で伝送されるビデオ信号の予想される総複雑度および/またはビデオ信号のプロバイダに対するチャネルの値段によって左右される場合がある。

伝送リンクにおけるビットを異なるチャネルに割り当てる好ましい方法は、式(5)の重み付け割当て方法と式(4)の最小限保証割当て方法を組み合わせたものである。この方法では、各チャネルには最小限割当てが保証され、残余のビットは重付け按分法で割り当

10

$$R^i = R G^i + \frac{P^i X^i}{\sum_{j=1}^K P^j X^j} [R - \sum_{j=1}^K R G^j] \quad (6)$$

上述したように、最小限保証割当てと重み係数は、チャネル経由で伝送されるビデオ信号の予想される総複雑度および/またはビデオ信号のプロバイダに対するチャネルの値段によって左右される場合がある。

システムの他のパラメータに応じてビット割当て $R^i$ をさらに改善することが可能である。例えば、上限ビットレート割当て値を越えると、再現イメージの品質が改善しないことが分

かっている。従って、この上限割当て値を越えてビットを割り当てることは、伝送リンクにおけるビットを無駄に消費することになる。また、伝送リンクの運用者には、各チャネルごとに上記最大限ビットレート割当て $R_{max}$ (これは上記の上限ビットレート割当て値を反映する場合がある)および/または最小限ビットレート割当て $R_{min}$ が課される場合もある。さらに、ビットレート制御が変動する可能性を最小限にし、従って、ビットレート制御の安定性を最大限にするためには、あるチャネルの、ある割当量周期から次の割当量周期までのビットレート割当ての増加 および/または減少 の最大限の増分量(インクリメント)が課される場合がある。上述したように、上限ビットレート割当て値、最大限および最小限ビットレート割当て、および増減の最大限増分量の値はチャネルが異なると、異なる

ことがあり、そのチャネル経由で伝送されるビデオ信号の予想される総複雑度およびビデオ

20

30

信号のプロバイダに対するチャネルの値段によって左右されることがある。さらに、増減の最大限および最小限増分量は、チャネルのバッファの空き度または満杯度に応じて動的に変化することも起こり得る。

さらに、割り当てられるビットレートは、バッファ管理が行えるように、例えば、CBRエンコーダ10(図1)の出力バッファ48および対応する受信側デコーダ(図示せず)の入力バッファがオーバフローまたはアンダフローしないようにさらに改善することが可能である。エンコーダのバッファ・サイズ $E$ が不等式(7)に示すように制御される場合には、明示的なバッファ管理は不要である。式において、 $D$ はデコーダの固定バッファ・サイズである。エンコーダ側バッファ・サイズが不等式(7)に従って選択されていれば、ビット

40

$$E \leq D \frac{R_{min}}{R_{max}} \quad (7)$$

エンコーダ側とデコーダ側のどちらのバッファにもオーバフローまたはアンダフローを引き起こさないで、 $R_{min}$ から $R_{max}$ まで変化させることが可能である。なお、この方法によると、エンコーダ側バッファのサイズが不当に制限されるので、レート制御の柔軟性が不当に制限される。

別のバッファ管理方式は適応方式であり、固定パラメータ $R_{min}$ と $R_{max}$ ではなく、現在の瞬時ビットレートを使用してバッファ管理を行うものである。デコーダ側バッファ・サイズは最高レート $R_{max}$ で伝送されたデータを処理できるように選択されているので、ビットレート割当てはデコーダ側バッファにオーバフローを引き起こすことなく、常に増加(シス

50

テム最大値 $R_{max}$ まで)することが可能である。しかるに、エンコーダ側バッファにすでに存在するデータがそのデコード時以前にデコーダ側バッファに伝送されることを保証するには、維持しなければならない瞬時最小限ビットレートが存在する。従って、デコーダ側バッファにアンダフローが起こらないことを保証する最小限ビットレートは動的に計算しなければならない。

この最小限ビットレート割当てを動的に計算する際、ビットレート割当てを減少するとき、新たに判断されるエンコーダ側バッファ・サイズと、ある先行時間量の間にエンコーダ側バッファにすでに存在するデータ量の両方を考慮に入れなければならない。フレーム $n$ 用の新たに判断されるエンコーダ側バッファ・サイズ( $E_n$ と呼ぶ)は式(8)に従って判断される。

$$E_n = \Delta R_{new} = \frac{R_{new} D}{R_{max}} \quad (8)$$

ただし、 $E_n$ はシステムの遅延時間であり、これはビデオのフレームがエンコーダに到着した時からそのフレームがデコーダ側に現れる時までの一定の時間遅延である。 $R_{new}$ は新たに提案されたビットレート割当てである。このバッファ・サイズにすると、新ビットレート割当てで安定状態が得られるので、エンコーダ側とデコーダ側のバッファにオーバフローまたはアンダフローが起こらないことが保証される。

なお、上述したように、新たに提案されたビットレート割当てが減少された場合は、システム遅延時間 $E_n$ に等しい遷移期間が生じ、その期間にエンコーダ側バッファにすでに存在するビット数が多くなりすぎて、新しい減少レートでデコーダに正常に伝送できないことが起こる。新たに提案されたビットレート割当てを改善する1つの提案方法では、実際にはエンコーダ側バッファに置かれているビット数( $e$ と呼ぶ)(バッファ満杯)を最初に調べて、システム遅延時間 $E_n$ におけるフレーム数( $F$ と呼ぶ)を確かめている。次に、先行フレーム数 $F$ の最大バッファ満杯数( $e_{max, F}$ )は式(8)から求めた、新たに判断されたエンコーダ側バッファ・サイズと比較される。先行フレーム数 $F$ からのすべてのビットが受信側デコーダに正常に伝送されることを保証する、チャンネル $i$ の最小限減少ビットレート割当て $R_{reduced}^i$ は式(9)で求められる。

$$R_{reduced}^i = \frac{e_{max, F}}{\Delta} \quad (9)$$

上記のような限界値がマルチプレクサ・システムで課されている場合は、ビットレート割当てが式(3)、(4)、(5)または(6)に従って計算されたあと、これらのビットレート割当てがチェックされ、そのチャンネルの現在の上限と下限の範囲内にあるかどうか判断される。最初に、各チャンネル $i$ の上限と下限が判断される。任意の割当量周期 $k$ の間の上限ビットレート割当て( $R_{upper}^i[k]$ と呼ぶ)は先行割当量周期 $k-1$ にわたる最大許容増加割当てと、最大ビットレート割当て限界値とのうちの最小のものである(式(10)を参照)。

$$R_{upper}^i[k] = \min\{R_{max}^i, (1 + \alpha) R^i[k-1]\} \quad (10)$$

任意の割当量周期 $k$ の間の下限ビット割当て $R_{lower}^i[k]$ は、最小限ビットレート割当てと、先行割当量周期 $k-1$ にわたる最小許容減少割当てと、式(9)から求めた最小限バッファ管理減少ビットレート割当てとのうちの最大のものである(式(11)を参照)。次に、チャンネルのビットレート割当ての調整が行われる。

$$R_{lower}^i[k] = \max\{R_{min}^i, (1 - \beta) R^i[k-1], e_{max, F} / \Delta\} \quad (11)$$

いずれかのチャンネル用に割り当てられたビットレートがどちらかの限界値を越えたときは、そのチャンネルのビットレート割当てはその限界値にセットされ、利用可能な残余ビットレートは他のチャンネル間で再割当てされる。例えば、チャンネル $i$ に割り当てられたビットレート(式(3)、(4)、(5)または(6)で計算したもの)がそのチャンネルの上限値(式(10)で計算したもの)より大であれば、チャンネル $i$ のビットレートはその上限値 $R_{upper}^i$ にセットされる。逆に、ビットレートが式(11)で計算した下限値より小であれ

10

20

30

40

50

ば、ビットレートはその下限値 $R_{lower}^i$ にセットされる(式(12)を参照)。

$$R^i[k] = \begin{cases} R_{lower}^i[k] & \text{if } R^i[k] < R_{lower}^i[k] \\ R_{upper}^i[k] & \text{if } R^i[k] > R_{upper}^i[k] \\ R^i[k] & \text{otherwise} \end{cases} \quad (12)$$

ビットレート割当てのいずれかが式(10)、(11)および(12)の制限的演算によって変更されたときは、利用可能な残余ビットレートは制限を受けないチャンネル間で、式(3)、(4)、(5)または(6)に従って再割当てされる。そのあと、これらのチャンネルは、再度、式(10)、(11)および(12)の限界値と突合わせ検査される。このサイクルは、すべてのビットレート割当てが完了するまで繰り返される。上記実施例では、符号化複雑度周期はスライディング・ウィンドウをベースとするピクチャ単位で判断されたGOP周期であり、これは十分な持続時間になっているため、ある割当量周期から次の割当量周期までの、チャンネルにおけるビットレート割当ての変更が一般的に相対的に小さくなるようにしている。その結果、式(10)、(11)および(12)はまれにしか呼び出されないようになっている。

符号化複雑度のサンプリングと符号化複雑度に基づく更新ビットレートの生成のタイミングは、チャンネルが異なるGOP時間周期で動作する場合には複雑になっている。しかるに、そのような場合に正確な符号化複雑度サンプリングとビットレート割当量の割当てが得られるようにするアプローチが2つある。第1のアプローチでは、各チャンネルが各GOPの中で同数の割当量更新周期をもつような形で一定の割当量更新周期が計算される。このアプローチでは、GOP当たりのサンプルと割当量更新周期の数はチャンネルとチャンネルとの間で変化することがあるが、どのチャンネルの場合も、GOP内の上記サンプルと割当量更新周期は一定になっている。第2のアプローチでは、サンプルがとられ、いずれかのチャンネルが新しいGOPを開始すると、新しい割当てが生成され、新しい割当量で割り当てられるビット数は先行サンプルから現サンプルまでの時間の長さを考慮に入れて計算される。

図7は、第1アプローチを使用するシステムでのサンプリングと割当量更新を示すタイミング図である。図面を簡単にするために、図には2チャンネルしか示されていない。図7において、チャンネル1はフレームレートが毎秒約30フレーム(米国の場合)である標準ビデオを伝送するチャンネルの例である。チャンネル2はフレームレートが毎秒24フレームであるフィルムを伝送するチャンネルの例である。これらのチャンネルの各々は、GOP当たりのフレーム数が12であるものと想定している。従って、チャンネル1は0.4秒ごとに新しいGOPを開始する。つまり毎秒2.5個のGOPを開始するのに対し、チャンネル2は0.5秒ごとに新しいGOPを開始する。つまり、毎秒2個のGOPを開始する。選択されたサンプリングレートは0.1秒ごとに1サンプルになっている。従って、チャンネル1では、4つのサンプルと割当量更新が各GOPにあり、チャンネル2では、5つのサンプルと割当量更新が各GOPにある。サンプリング時刻 $t_s$ は縦の破線で示されている。サンプル間の時間周期 $t$ は一定(0.1秒)であるので、上記の式(3)~(12)は次のサンプル周期のビットレート割当てを計算するとき、変更を加えないで使用することが可能である。これらのビットレート割当ては、「トークン・リーキバケット(token and leaky bucket)」方式と呼ばれる公知の方式に従って、チャンネル・プロセッサ(図1)で累積して使用することが可能である。

図8は、第2アプローチを上述したように使用するシステムでの符号化複雑度値のサンプリングと割当量更新を示すタイミング図である。図8に示すそれぞれのチャンネルは図7に示すものと同じ信号を伝送する。図8に示すように、すべてのチャンネルからの現在の符号化複雑度値のサンプルは、いずれかのチャンネルが新しいGOPを開始するととられる。新しい割当てはこれらのサンプルの値と、最後のサンプル以後の時間周期 $t$ に基づいて行われる。これらのサンプル時刻は図8に縦の破線 $t_1$  -  $t_8$ で示されている。ただし、 $t_2$ 、 $t_3$ 、 $t_4$ 、 $t_6$ および $t_8$ はチャンネル1におけるGOPの開始に対応し、 $t_1$ 、 $t_3$ 、 $t_5$ および $t_7$ はチャンネル2におけるGOPの開始に対応している。 $t_3$ は両方のチャンネル1と2におけるGOPの開始に対応するサンプル時刻を示しているが、そのような時刻が現れることは必要条件ではない。

10

20

30

40

50

各サンプル時刻に、すべてのチャンネルにおける現在の符号化複雑度値（先行GOPからのもので、スライディング・ウィンドウをベースとするピクチャ単位で利用可能である）がサンプリングされる。式（1）～（12）は、次のビットレート割当量の大きさを計算するために使用できるが、割当てのために利用できる実際のビット数を求めるには、最後のサンプル以後の時間量  $t$  を考慮に入れなければならない。異なるサンプル周期を正しく補償するためには、式（3）～（12）における総利用可能ビットレート  $R$  に、割当てのために利用できるビット数  $C$  が代入されるが、これは総利用可能ビットレート  $R$  とサンプル周期  $t$  の積である。つまり  $C = R \cdot t$  である。式（3）～（12）で求めたビット数はそれぞれのチャンネル・プロセッサ10（図1）に割り当てられたあと、チャンネル・プロセッサは、上述したように「トークン・リーキバケット」方式を使用して割当てビットを累積して使用する。

10

上述した2アプローチによれば、どちらも、異なるチャンネル5からのビデオ信号が異なるGOP時間周期をもっているときビットレートがそれぞれのチャンネル・プロセッサ10に正確に割り当てられる。

符号化複雑度値のサンプリングと異なるチャンネル用の更新ビットレート割当量の生成のタイミングは、チャンネルのすべてが同一フレームレートで動作し、GOP内のフレーム数が同一である場合には、つまり、すべてのチャンネルが同一GOP時間周期  $GOP_{time}$  をもつ場合には、単純化することが可能である。図6は、そのようなシステムでの符号化複雑度サンプルと割当量更新のタイミングを示すタイミング図である。図6において、各水平線はそれぞれのチャンネル  $1 - k$  に対応している。水平線から上方に延びた短い縦線は、Iフレームの符号化がそのチャンネルで開始される時刻を表し、これはそのチャンネルのGOPの開始とみなされる。GOPの時間周期  $GOP_{time}$  はどのチャンネルの場合も等しくなっているが、それぞれのチャンネルのGOPの開始時刻は異なっている。事実、それぞれのチャンネルのGOPの開始時刻が異なっていると、Iフレームの符号化がオーバーラップしないという点で望ましいことが判明している。このようにすると、異なるチャンネル間の複雑度に変化をもたせることができる。

20

符号化複雑度値を計算するとき考慮に入れるIフレーム、PフレームおよびBフレームが同数であるかぎり、これらのフレームが異なるGOPからのものであることは重要でない。従って、すべてのチャンネルの時間軸を横切る実線で示すように、符号化複雑度値のサンプルは、GOP内でいつでもすべてのチャンネルから同時にとることができる。その後、チャンネルのすべてのビットレート割当量の更新をそのサンプルから生成して、チャンネル・プロセッサ10（図1）へ送り返すことができる。

30

上述のマルチプレクサ・システムは一箇所に配置されたシステムとして説明してきた。しかし、複数のチャンネル・プロセッサ10はビットレート・アロケータ30およびマルチプレクサ20から離れたロケーションに置いておくことも可能である。そのようなシステムでは、通信リンクはエンコーダとビットレート・アロケータの間で確立されることになる。その場合には、プロセッサ10とマルチプレクサとの間で伝送されるビットの一部を、プロセッサからの複雑度情報の伝送に専用することが可能である。



---

フロントページの続き

(51)Int.Cl.<sup>7</sup> F I  
H 0 4 N 7/24

審査官 阿部 弘

(56)参考文献 特開平06-062393(JP,A)  
特開平07-111649(JP,A)

(58)調査した分野(Int.Cl.<sup>7</sup>, DB名)  
H04J 3/00 - 3/26  
H04L 5/22 - 5/26  
H04 7/12 - 7/32