

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4729570号

(P4729570)

(45) 発行日 平成23年7月20日 (2011. 7. 20)

(24) 登録日 平成23年4月22日 (2011. 4. 22)

(51) Int. Cl.

F I

H04N 7/173 (2011.01)

H04N 7/173 610B

請求項の数 1 (全 33 頁)

(21) 出願番号	特願2007-522794 (P2007-522794)	(73) 特許権者	505369033
(86) (22) 出願日	平成17年7月22日 (2005. 7. 22)		ビーチ・アンリミテッド・エルエルシー
(65) 公表番号	特表2008-507922 (P2008-507922A)		アメリカ合衆国ネバダ州89119, ラス
(43) 公表日	平成20年3月13日 (2008. 3. 13)		ベガス, ルネッサンス・ドライブ 221
(86) 国際出願番号	PCT/US2005/026011		5ービー, スイート 5
(87) 国際公開番号	W02006/012496	(74) 代理人	100070150
(87) 国際公開日	平成18年2月2日 (2006. 2. 2)		弁理士 伊東 忠彦
審査請求日	平成19年4月6日 (2007. 4. 6)	(74) 代理人	100091214
(31) 優先権主張番号	60/590, 504		弁理士 大貫 進介
(32) 優先日	平成16年7月23日 (2004. 7. 23)	(74) 代理人	100107766
(33) 優先権主張国	米国 (US)		弁理士 伊東 忠重
前置審査			
			最終頁に続く

(54) 【発明の名称】 トリックモードおよび速度移行

(57) 【特許請求の範囲】

【請求項 1】

帯域調整モジュールを有するシステムのためにデータストリームを通信する方法であって、

(A) 前記帯域調整モジュールにより、第1のデータストリームのための第1のタイムスロットを決定する工程と、

(B) 前記帯域調整モジュールにより、第2のデータストリームのための第2のタイムスロットを決定する工程と、

(C) 前記帯域調整モジュールにより、前記第2のデータストリームが前記第2のタイムスロットより大きい場合、当該第2のデータストリームの一部を前記第1のタイムスロットに移動する工程と、

前記工程 (C) は、

前記帯域調整モジュールにより、前記第2のデータストリームのタイムスロット数、フレーム数、ビットレート及びフレームレートに基づき前記移動部分の量を決定する工程、

を更に有し、

前記第1のタイムスロットは前記第2のタイムスロットの前に生じる、

ことを特徴とする方法。

【発明の詳細な説明】

【技術分野】

【0001】

10

20

本出願は、2004年7月23日に出願された「Buffer Optimized Trickmodes and Speed Transitions」と題する米国特許仮出願第60/590,504号に対する優先権を主張するものであり、前記仮出願はこの参照によりその全体が本明細書に組み込まれる。

【0002】

本発明の開示はビデオデータを伝送するための方法に関する。

【背景技術】

【0003】

ビデオフレームは、早送り、巻き戻し、一時停止、停止、および再生が可能なデータを表すのに使用される場合がある。そのようなビデオフレームに関連付けられたデータの任意の時間またはタイムスロットに伝送または受信されるデータ量を使い、利用可能な帯域幅を決定することができる。ビデオフレームの1つのタイプは、「トリックモード (trick mode)」フレームと呼ばれるものであり、数多くの異なるビデオ伝送方法に使用されている。

10

【0004】

ビデオ伝送におけるビデオフレームの伝送の管理は、望ましい質のビデオ制作を達成するための一つの考慮点である。トリックモードフレームの場合、任意のフレームに関連付けられたデータ量が予測不可能で、かつ変動する場合があります、これが管理上の問題点となることがある。例えば、利用可能なタイムスロット内にトリックモードフレームが確実に収まるようにするためにタイムスロット法が使用されることがあるが、このタイムスロット法では各スロットは最大のトリックモードフレームを伝送するのに十分な帯域幅の固定量を有する。しかし、その他のより短いトリックモードフレームで、タイムスロットが必要以上に大きい場合、帯域幅が未使用となる場合がある。その結果、未使用の帯域幅の量が増える。

20

【0005】

また、多くの場合、トリックモードフレームの伝送前にトリックモードをバッファするために必要なメモリ量を予測することが難しい。その結果、時として、バッファを必要とするデータ量が、利用可能なメモリを超え、「バッファオーバーフロー」を引き起こす場合がある。このバッファオーバーフローの結果、ジャンプやジッタのような望ましくない様々な視覚的状況が発生する場合がある。

30

【発明を実施するための最良の形態】

【0006】

本発明の実施形態は、トリックモード・ビデオ・ストリームのようなビデオデータ通信において帯域幅を効率的に使用すると同時に、望ましいビデオ再生条件を提供するバッファレベルを維持する方法を提供する。Motion Picture Experts Group (MPEG: エムペグ) のコンテキストで実施形態について説明するが、上述の方法をその他のタイプのデータ圧縮/解凍方法に用いることも可能であると理解されるべきである。

【0007】

一部のビデオ圧縮/解凍方法では、ビデオはフレームに分割される場合がある。例えば、MPEGは、Iフレーム、Pフレーム、Bフレームという少なくとも3つの異なるタイプのビデオフレームを使用する。Iフレームすなわち「intra coded frames」は、シーケンスに他の前フレームまたは将来フレームがなくてもIフレームのデコードを可能にするイントラフレームマクロブロックを含む。MPEGビデオのランダム再生では、デコーダはIフレームからデコードを開始することができる。12~15フレームごとにIフレームを挿入可能であり、Iフレームを用いてシーケンスを開始することができるので、ランダムな位置からビデオを再生すること、および例えば早送りや巻き戻しなどトリックモード機能でビデオを再生することが可能である。

40

【0008】

Pフレームは前フレームとの差分としてコード化される。新しいPフレームは、前フレ

50

ームおよび現在フレームの新しいピクセルの予測値を取り入れて予測することができる。
Pフレームは、存在する動作量に応じてより高い圧縮比を提供する。

【0009】

Bフレームすなわち「双方向フレーム」は、前Bフレームまたは後続Bフレームとの差分としてコード化されるものであり、前フレームおよび後続フレームを使って正確にデコードすることができる。このため、読み込まれるフレームの順序は、必ずしも表示順序と同じでない場合がある。すなわち、後続フレームが現在Bフレームの前に伝送およびデコードされても、現在フレームの後に表示される場合があることを意味する。例えば、 I_1 B_2 B_3 P_4 B_5 B_6 P_7 という表示シーケンスのフレームは、 I_1 P_4 B_2 B_3 P_7 B_5 B_6 と順序換えされて伝送される可能性もある。

10

【0010】

MPEGビデオのシーケンスは画像群 (Group of Pictures: GOP) を含む。各GOPはビデオフレームを含む。GOP構造には、GOP構造が含むフレーム数 (N) および2つの参照フレーム間の距離 (M) が関連付けられている。例えば、典型的なGOP構造はIBBPBBPBBPBBPBBP (この構造において $N=15$ 、 $M=3$)、および／またはIBBPBBPBBPBB (この構造において $N=12$ 、 $M=3$) などである。当然、これら構造は変化するものであり、例えばPPPPPPPPなどPフレームだけのストリームを含むこともある。

【0011】

トリックモードGOPは、Iフレームと、可変数のダミーBフレームおよびPフレームとを含むビデオシーケンスである。トリックモードGOPのサイズには、トリックモードGOPに含まれるフレーム数が関連付けられている場合がある。例えば、IBBPPPというGOP構造のGOPサイズは6である。GOPまたはトリックモードパケットのタイムスロットは、IフレームDTS (デコード・タイム・スタンプ) から次のIフレームDTSまでの期間が可能である。

20

【0012】

トリックモードパケットはトリックモードGOPと関連付けられており、有効なトランスポートストリームを調整するためのデータを含む。前記トリックモードパケットは、プログラム割り当てテーブル (Program Allocation Table: PAT) と、プログラム・マップ・テーブル (Program Map Table: PMT) と、「同期」パケットとして参照される同期のためのみの (例：データなし) プログラム・クロック・レファレンス (Program Clock Reference: PCR) を有するトランスポート・ストリーム・パケットと、トリックモードGOPと、様々なサイズのフィラー・ヌル・パケット (filler null packets) とを含む。

30

【0013】

トリックモードパケットのサイズは、トリックモードパケット自体に基づくサイズである可能性がある。前記トリックモードパケットのサイズは、ストレージオーバーヘッド、ネットワークオーバーヘッド、およびビットレートコントロールに対応することもできる。例えば、Iフレームを含むファイルセグメントは、トランスポート・ストリーム・レベルで多重化されたその他のパケット識別子または「PID」 (例：PAT、PMT、オーディオ) を含むこともできる。ブロックをメモリに読み込み、非ビデオパケットをヌル (例：「ミューティング」) に替えることも可能である。

40

【0014】

トリックモードパケットは1316バイト (例：ユーザ・データグラム・プロトコル (UDP) を介したMPEG2トランスポートストリーム) も可能である。

【0015】

図1は、固定タイムスロット割り当てを使ったIフレームベースのトリックモードのビデオ・バッファ・レベルの一例である。図1が示すように、各GOPは7つのフレームを含む。その他のサイズのGOPを使用することも可能である。図1で図示する横軸目盛 (t

50

）は、フレーム期間（例：1／30秒）を表す。例えば、当該「トリックモードGOP」構造はIBBP P P Pであってもよいし、または2つのダミーBフレームおよび4つのPフレームが後続するIフレームであってもよい。この構造では30秒毎に7つのIフレーム（すなわち4．28秒毎に1つのIフレーム）が作成される。図1の $t = 2$ の位置の縦線が示すように、第1のGOPは $t = 2$ で受信される。前記GOP構造は7つのフレームで設定されているため、第1のGOP構造は $t = 7$ まではデコードおよび提示されない。第2のGOP構造は $t = 13$ で受信される。この例では、前記第2のGOPが伝送およびバッファされる間、デコーダは $t = 7$ から $t = 13$ まで第1のGOPを提示する。 $t = 14$ で、前記第2のGOPはすでにバッファされており、デコードできる状態にある。

10

【0016】

GOPのデコードおよび提示ができる状態にあるインターバルの前にGOPを受信した場合、前記デコーディングプロセスに中断が起こらない。しかしながら、GOPのデコードおよび提示準備が整う前記インターバルの前にGOPを受信可能であるため、未使用の帯域幅が存在する可能性がある。図1において、この未使用の帯域幅は長方形部分として描かれている。

【0017】

無駄な帯域幅の問題を解決するための1つの試みは、例えばそのタイムスロットおよび／またはGOPサイズを減らすことによって、毎秒送信されるIフレーム数を増やすことである。例えば、図1を参照し、GOPを6フレームに減らすことによって、前記第1のGOPによる未使用の帯域幅の量は1フレーム少なくなる。しかし、前記第2のGOPは、そのデコードおよび提示後の $t = 13$ ．5まで伝送状態ではない。前記第2のGOPが伝送されるには約6．5フレーム（すなわち13．5－7）のインターバルが必要なため、 $t = 12$ で前記第1のGOPによって不完全な第2のGOPがデコーダに提示されることになる。不完全な第2のGOPをデコーダに提示すると、「バッファアンダーフロー（buffer underflow）」が生じる可能性がある。1実施形態において、GOPの長さおよびタイムスロットを、その後続GOPのサイズの関数とすることができる。例えば、図1

20

のサンプル分布が示すように、一部のIフレームは2フレームという短いタイムスロットのみを必要とし、他のIフレームは6フレームという長さである場合もある。

30

【0018】

図2は、対象GOPに後続するGOPに基づき前記対象GOPの長さを増す方法の一例である。図2が示すように、同一のGOPシーケンスを用い、各GOPについて合計インターバルをランダムにすることによって、未使用の帯域幅の量（長方形部分によって示される）をわずか1インターバル長に減少することができる。具体的には、前記第1のGOPは4フレームのインターバルを有し、前記第2のGOPは5インターバルを有する。それにより生成されるトリックモードシーケンスは、ランダムなインターバルで提示される新しいIフレームを導入することができる。

【0019】

図1および図2が示す方法は、すべてのGOPがデコードされた後、バッファは残りのダミーPフレームとBフレームを含み、これらはごく小さいサイズであるため、ビデオバッファが空となるという前提に基づいて行われている。

40

【0020】

データの取り込みプロセスは、受信される「N」個めのフレームごとに1フレームをデコードし、「N速度」のトリックモードストリームを生成する。例えば、8xのストリームを生成するために、前記取り込みプロセスはフレーム1，9，17，25．．．（ $8n + 1$ ）をデコードすることができる。次に、これらのフレームを使って、結果的に得られるストリームが例えば毎秒30の一意のフレームを含むように、MP EG2トランスポートストリームを生成することができる。次に、フレームのシーケンスを、元のトラン

50

スポーツストリームの一部の特性（フレームレート、ビットレート、P I D割り当て、ビデオフォーマット、ビデオバッファ特性（例えば滑らかな速度移行のためのバッファサイズおよびバッファレベル）など）を維持する新しいM P E G 2トランスポートストリームにエンコードすることができる。この方法によって、より優れたトリックモードの品質を提供することが可能である。また、この方法は、より高いプロセッサ能力と追加的なストレージオーバーヘッド（例えば通常30%）を使用する。

【0021】

一部の実施形態では、IフレームベースのトリックモードはダミーBフレームおよびPフレームを挿入することもできる。また、前記実施形態の一部では、フレームジッタになる可能性のあるPフレームおよびBフレームを、フレーム予測を用いてエンコードすることができる。例えば、放送テレビジョン（すなわち全国テレビジョン方式委員会（National Television System Committee：NTSC））を用いる実施形態では、各フレームは2つのインターレースフィールドから構成されており、合計每秒60フィールドとなる。24fpsという映画のフレームレートと30fpsというテレビのフレームレートの差分の結果、映画をテレビ用コンテンツに変換するために「3：2引き下げ」方法を採用することがある。

10

【0022】

前記「3：2引き下げ」方法は、フレームを3フィールドと2フィールドに交互に変換することができる。例えば、24fpsで4フレーム（すなわち6秒毎に1フレーム）では、30fpsで10フィールドまたは完全な5フレームが作成される。（プログレッシブモードではなく）インターレースモードを使ったエンコードでは、1フレームは2フィールド（上のAと下のB）を含む。ダミーBフレームとPフレームを生成するためにフレーム予測を用いると、デコーダはその参照画像またはIフレームから両方のフィールドをコピーすることができる。よって、例えば、構造IBBPを持つトリックモードGOPでは、前記デコーダはABABABABABという構造を有する5フィールドのシーケンス（各フレームに2つのAB）が作成される。

20

【0023】

Iフレームは2つの異なる画像から生成されたフィールドを含む場合があり、ABABABABABというフィールドのシーケンスは「ジッタ」の印象を与えることがある。

【0024】

一部の実施形態では、トリックモード機能を開始する際、BフレームおよびPフレームによって使用される参照フレームは、これらのフレームを出力ストリームにコピーするときに破壊される可能性がある。この原因の一部は、Nフレームから1フレームを選択することによってトリックモードファイルが生成されることにある。結果的に、前記BおよびPフレームは前記出力ストリームにもはや存在しない可能性があるため、これらのフレームを欠落している参照フレームと共に完全にデコードし、次に、異なる参照フレームを用いて前記トリックモードストリームのコンテキストで再エンコードする必要がある。

30

【0025】

例えば、IBBPBBPBBPBBPBBIBBPBBPBBPBBPBBIBBPというビデオ・フレーム・シーケンスは、前記トリックモードシーケンスに対しIBBPBBPによって表すことができる。元のビデオ・フレーム・シーケンスのBフレームは、前記トリックモード・シーケンス・ファイルの一部ではない前および後続のIおよびPフレームに依存する場合がある。また、前記トリックモードファイルでは、シーケンスへの挿入場所により、一部のフレームは異なる方法でエンコードされる。例えば、前記元のシーケンスのIフレームを、前記トリックモードファイルでPフレームとしてエンコードし、BフレームをPフレームにすることも、全く異なる参照フレームを持つ別のBフレームにすることも可能である。

40

【0026】

ケーブル業界で使われる典型的な帯域速度は、Cable Labs（商標）コンテンツ仕様1.0[4]の説明によれば、3.75メガビット/秒であり、每秒30フレームの

50

ストリームが3.75メガビット/秒で取り込まれる。4つのトリックモードファイルが必要とするビデオ・オン・デマンド（VOD）サーバの場合（例えば15x、-15x、60xおよび-60xの速度）、4つのエンコーダを平行して使用して元のビデオストリームから抽出およびデコードされたフレームを処理することによって、最高4つの異なるトリックモードファイルを生成することができる。

【0027】

各トリックモードエンコーダは、毎秒2フレーム（2fps）（30/15）、2fps（30/15）、0.5fps（30/60）、および0.5fps（30/60）で受信することができ、4つのすべてのエンコーダで毎秒合計5フレームとなる。通常、例えば2.4GHzのPentium（商標）4プロセッサから得られる処理能力のすべてを実質的に使用することにより、約45メガビット/秒の取り込み帯域幅で12ストリームをエンコードすることが可能である。その結果得られるトリックモードファイルは、それぞれ前記元のファイルサイズの6.7%、6.7%、2.2%、および2.2%であり、合計約16.6%のストレージとなる。よって、一部の実施形態では、標準トリックモードファイルを生成するために、かなり高度なコンピュータ処理能力およびコンピュータ論理が必要とされる場合がある。さらに、前記プロセスは一部のフレームを完全にデコードし、Bフレームのようなその他のフレームをデコードしない場合がある。

【0028】

Iフレームの表示は前フレームまたは後続フレームに依存しないので、Iフレームをランダムアクセス機構に用いることが可能である。よって、一部の実施形態では、新しく作成されたストリームにIフレームをマージし、ビットレートを制御するためにヌルパケットを挿入することによってトリックモードを使うことができる。

【0029】

Iフレームの通信所要時間はフレーム間隔より長い場合がある。例えば、平均Iフレームサイズ40kbでは、30fpsのIフレーム専用トリックモードストリームは最低限9.8メガビット/秒（40kb x 8 x 30）すなわちケーブル産業で一般に使用される3.75メガビット/秒という速度の約2.6倍の速度を必要とする。

【0030】

一部の実施形態では、30fpsのように高いフレームレートを維持するために、前記Iフレームレートを毎秒約10Iフレームに減少させる場合がある。例えば、残りのIフレームの代わりに他のフレームを挿入することによって速度を維持することができる。例えば、2若しくはそれ以上のフレーム期間またはフレーム間隔に1つのIフレームを表示し、後続のIフレームを伝送およびバッファすることができる。最後に表示されたフレームのコピーである「ダミー」BフレームまたはPフレームをビデオストリームに挿入することができる。1実施形態において、「ダミー」または複製フレームは、平均Iフレームサイズよりも小さいサイズの「静止」フレームである場合がある。ダミーフレームは実質的に同時にエンコードできるので、プロセッサの効率を高めることができ、GOPのサイズを拡張するために出力ストリームに挿入可能である。

【0031】

毎秒10のIフレームという速度のトリックモードは、「トリックモードGOP」すなわちビデオフレームのトリックモード・シーケンスを作成することによって達成できる。例えば、「トリックモードGOP」のシーケンスを、後続する2つのダミーBフレームを持つ1つのIフレームで作成することによって、例えばIBBIBBIBBIBBIBBのようなシーケンスを作成することができる。前記ダミーBフレームのサイズが例えば約1.2kbであれば、その出力ストリームの平均ビットレートは3.5メガビット/秒（40kb * 8 * 10 Iフレーム/秒） + （1.2kb * 8 * 20 Bフレーム/秒）となり、最大帯域速度3.75メガビット/秒以内となる。

【0032】

一部の実施形態では、タイムスロット割り当ておよびトリックモードGOPサイズをIフレームベースのトリックモードストリームで動的に調整し、帯域幅の使用を効率化する

10

20

30

40

50

。これらの方法は、さらに、バッファオーバーフローを検出および予防することによってビデオバッファ稼働率を監視する。1 態様においては、「ダミー」P フレームを挿入する場合がある。

【0033】

また、これらの方法を用いて帯域幅利用率を最大限にすると同時にバッファレベルを最少レベルに維持し、処理速度の変化につれてシステムの応答性を増すことができる。例えば、8 x のストリームを生成するために、これらの方法は平均して每秒約 10 の一意の I フレームを提供することができる。残りのフレーム（例えば每秒 30 フレームのストリームで每秒 20 フレーム）は、ダミーまたは静止 B フレームおよび P フレームが可能である。

10

【0034】

1 実施形態では、これらの方法をビデオ・ストリーム・ソフトウェア製品に組み入れる。また、この方法を、ハードウェア、ファームウェア、またはそれらの任意の組み合わせを用いて達成することも可能である。

【0035】

データ構造を用いてビデオの取り込みストリームを解析することができる。例えば、「ヒンター (H i n t e r)」構造およびその解析データ（すなわち I フレームおよびストリーム情報）を「ヒント (H I N T)」ファイルというファイルに保管することができる。ビデオの取り込み、例えば、通常の P e n t i u m 4 (商標) 2.4 GHz プロセッサで約 300 メガビット/秒の取り込みが可能であると理解される。前記ヒントファイルは、例えば約 64 k のヘッダを含むことができる。また、前記ヒントファイルは、1 つの I フレームにつき約 128 バイトの I フレームテーブルを含むことができ、関連付けられた I フレームの位置を示すポインタを有することができる。3.75 メガビット/秒で每秒 20 の I フレームを有する 2 時間の映画（すなわち合計 14400 の I フレーム）では、元のファイルサイズの約 0.06 % 未満である約 1.9 メガバイトのサイズのヒントファイルが作成される。しかし、取り込み時に前記のトリックモードが生成されない場合があるので、例えばソフトウェアおよび/またはハードウェアのストリーミングに使われる場合、これらの方法は動的にトリックモードストリームを生成する場合がある。

20

【0036】

図 3 は、I フレームのシーケンスがバッファレベルに与える影響を図示する。縦軸は I フレームのデコード時間を表す。図 3 が示すように、トリックモードパケットはバッファリングを利用することによって「バック」される。I フレームの大型シーケンスによって、フレームレートにほとんど若しくは全く影響を与えずに、バッファレベルが短期間上がる場合がある。ダミー P および B フレームも伝送およびデコードされるが、それらは I フレームの約 30 倍小さく、それぞれ約 1.3 k である。

30

【0037】

一部の実施形態では、バッファレベルの調整に用いられるこれらの方法は固定タイムスロットのトリックモードシーケンスから導き出されている場合がある。そのようなシーケンスは図 1 を参照して説明したシーケンスと類似している場合がある。また、比較的優れた表示品質を達成するために、これらの方法によって、実質的に一定の速度の I フレームを生成する試みが行われる。前記のビデオストリームは任意の分布を有する I フレームを含むことができるが、本発明の理解および明確化のために、以下の説明では、図 4 が示すように、入力ビデオストリームが一定のサイズの分布曲線を含むものと仮定する。

40

【0038】

上記の方法は、与えられたサイズを満たさない I フレームを含むタイムスロットの未使用の帯域幅を再分配し、超過サイズの I フレームを取り入れることができる。これは、例えば、(GOP サイズに基づき) 十分なトリックモードのタイムスロットサイズを選択し、バッファオーバーフローを発生せずに一式のトリックモードパケットが確実に調整または再配列されるよう十分に大きいタイムスロット調整または枠を選択することによって達成できる。

50

【0039】

伝送前にトリックモードパケットのシーケンスを再配列することにより、帯域幅利用率を改善することができる。また、本発明の方法は、統計平均を採用し、任意のシーケンスのトリックモードパケットのGOPサイズを取り入れることができる。1実施形態では、前記GOPサイズを確実にそのパケット調整のサイズ未満若しくは最低限そのサイズと同等にすることによって平均化が達成される。

【0040】

実質的に固定されたIフレームのタイムスロットからのIフレームをこのように再分配および再配列する結果、前記の方法はデコードおよびそれに伴うバッファリングを含む場合がある。さらに、バッファされたデータの量を管理することにより、バッファオーバーフローまたはアンダーフローという状況を予防することができる。

10

【0041】

図5は、トリックモードパケット調整を図示する。図5が示すように、上の枠501は、一部の超過サイズのトリックモードパケットがその固定タイムスロットおよび利用可能なタイムスロットに適合しない場合があることを図示している。例えば、トリックモードパケット506は図のタイムスロット503内に適合しているが、トリックモードパケット504は前記503の後続タイムスロット505内に適合していない。結果として、トリックモードパケット504の一部はタイムスロット507に入り込んでいる。

【0042】

下の枠502は、前記タイムスロットを再配列または順序替えする方法を示している。そのような再配列によって、前または後続のトリックモードパケットからの利用可能な帯域幅または「ヌル」を、より大きいトリックモードパケットに使用することが可能になる。

20

【0043】

以下の説明において、上述の概念を数学的に定量化する。しかし、本発明の開示がこれらの等式の計算または使用に限定されるものではないことが理解されるものである。むしろ、以下の説明は、この新規概念の理解を深めることを目的として提供されるものであり、例として与えられる等式は本発明の実施形態で考慮される可能な1つのアプローチに過ぎない。

【0044】

ビットレート b_r 、フレームレート f_r 、および q フレームのGOPサイズを有するデータストリームでは、与えられたタイムスロットにおいて伝送可能なデータ量 Q は次のように表される。

30

【0045】

【数1】

$$Q = \frac{qb_r}{8f_r}$$

【0046】

そのIフレームレートは次のように計算することができる。

40

【0047】

【数2】

$$r = \frac{f_r}{q}$$

【0048】

場合によっては、よりよい表示品質を得るために、1GOP当たりのフレームを定数にすることにより q を整数として維持することが好ましい。あるいは、GOPごとに q を変動させる（すなわちGOPサイズを変動する）こともできる。その他の可能な実施形態は、 q の平均値を用いる場合がある。例えば、2.4フレームの q の値を、2, 3, 2, 2

50

、3というサイズを有し毎秒12.5のIフレームを提供するGOPのシーケンスに対して設定することができる。

【0049】

上記の等式を $b_r = 3.75$ メガビット/秒、 $f_r = 30$ fps、および $q = 3$ であるビデオストリームに適用することにより、 $r =$ 毎秒10のIフレームであるIフレームレートで、各タイムスロットにて46,875バイトを伝送することができる。

【0050】

nトリックモードパケットのある調整枠に必要な帯域幅を予測するために、多くの観点の予測を考慮する必要がある場合がある。例えば、Iフレームのサイズ、ダミーBフレームとPフレームの数およびそれらのサイズ、PAT、PMT、PCRパケットのようなオーバーヘッド・データのサイズ、およびディスクとネットワークのオーバーヘッドなどを考慮しなくてはならない場合がある。これらの値を予測することができる。例えば、元のストリームでのIフレームのサイズが、特定の分布曲線に必ずしも従わない一定のサイズの分布(I, σ_1)を有することができるのと予測することができる。ディスクまたはストレージのオーバーヘッドは、トリックモードパケットのサイズを決定することによって予測することができる。

【0051】

上記の予測は、ビデオ・バッファ・レベルを約10%高く予測する場合がある。結果として、一部の実施形態では、ストレージから直接入手される40KBブロックにおいて、ビデオバッファに保管されることになる実際のビデオデータ36KB以下を伝送しなくてはならない場合がある。残りの4KBは、前記ブロックに埋め込まれていて「ヌル化」または「ミュート化」されたその他のPID（例えばオーディオ、PMT、PAT）を含むことができる。あるいは、前記ビデオデータを再配列し、約36KBのビデオデータを送信することができる。

【0052】

バッファオーバーフローを確実に避けるために、本発明の方法はバッファ能力の90%という制限を設定することができる。この90%制限は、上述の方法におけるエラーの増大を予防することもできる。前記バッファレベルはそのトリックモードシーケンスにおいて超過サイズのIフレームの比較的大きいシーケンスを必要とするため、前記バッファレベルがその制限の90%に到達する確率は比較的低い。さらに、前記バッファレベルを高めに予測することにより、前記トリックモードの性能をあまり劣化させずにバッファオーバーフローを検出する確率が高まる。

【0053】

n個のランダムなIフレームのシーケンスにおける各Iフレームのサイズを決定する場合、合計サイズ

【数3】

$$S_n = \sum_{i=0}^{n-1} I_i$$

【0054】

は

【数4】

$$(nI, \sqrt{n\sigma_1})$$

【0055】

という分布を持つランダムな変数となる。nの値が大きいほど、前記ランダムな分布(S_n)は正規分布に近づく。この予測の結果生じるすべてのエラーをPフレーム挿入によって修正することができる。

【0056】

さらに、前記のダミーPフレームおよびBフレーム(P)およびオーバーヘッドデータ

(OH) のサイズを予測することによって、 n タイムスロットを持つ調整枠に含まれるストリーミングデータ (T_n) の合計量を以下の等式によって表すことができる。

【0057】

【数5】

$$T_n = S_n + nOH + n(q-1)P = n(I + OH + (q-1)P)$$

$$\sigma_T = \sigma_S = \sqrt{n}\sigma_1$$

【0058】

前記調整枠にて利用可能な帯域幅を以下の等式によって表すことができる。

【0059】

【数6】

$$Q_n = nQ = nqb_r/8f_r$$

【0060】

トリックモードパケットのシーケンスが指定調整枠に適合する確率を最大限にするために、本発明の方法は、一部の実施形態で、修正 ε を実行する必要がある確率を低く維持しようと試みることができ、ここで、 10^{-3} の階数の値に対し $\varepsilon > P(T_n > Q_n)$ である。

【0061】

S_n が正規分布と考慮され得るほど n が十分大きいと考えれば、 n は以下の等式を満たすのに十分な大きさである場合がある。

【0062】

【数7】

$$\text{erf}\left(\frac{Q_n - T_n}{\sigma_T}\right) < \varepsilon$$

【0063】

前記等式に、 $B_r = 3.75$ メガバイト/秒、 $f_r = 29.97$ fps、 $q = 3$ フレーム、 $I = 40491$ バイト、 $\sigma_T = 10835$ バイト、 $P = 1.0$ kb、 $OH = 2.0$ kb という実数値を投入すると、次のようになる。

【0064】

【数8】

$$Q_n = n \cdot \frac{3 \cdot 3.75 \cdot 10^6}{8 \cdot 29.97} = 46921.92 \cdot n$$

$$T_n = (40491 + 2048 + 2 \cdot 1024) \cdot n, \sigma_T = 10835 \cdot \sqrt{n}$$

$$T_n = 44587 \cdot n, \sigma_T = 10835 \cdot \sqrt{n}$$

【0065】

$\varepsilon = 10^{-3}$ であれば、その調整枠は次のようになる。

【0066】

【数9】

$$\text{erf}\left(\frac{(45922 - 44587) \cdot n}{10835 \cdot \sqrt{n}}\right) < 10^{-3}$$

$$\text{erf}(0.1232 \cdot \sqrt{n}) < 10^{-3}$$

$$0.1232 \cdot \sqrt{n} \geq 3.08$$

$$n \geq 625$$

10

20

30

40

50

【0067】

ここで、Iフレームの平均サイズは選択されたタイムスロットのサイズよりわずかに小さく、本発明の方法は每秒約10のIフレームのトリックモードストリームを可能にする。また、この例において、その帯域幅利用率は約97%である（すなわち44587バイト割る45922バイト）。百分率で表す実際の帯域幅利用率はバッファオーバーフローを修正する必要性（例えばPフレームの挿入）によって低下する場合がある。

【0068】

以下の例は、1つの調整枠サイズを選択し、最大トリックモード速度または最小GOPサイズ（ q ）を決定することによって、異なるアプローチを採用する。

【0069】

【数10】

$$Q_{\text{eff}} = 64 \frac{q \cdot 3.75 \cdot 10^6}{8 \cdot 29.97} = 1.001 \cdot 10^6 \cdot q$$

$$T_{\text{eff}} = (40491 + 2048 + (q-1) \cdot 1024) \cdot 64, \quad \sigma_T = 10835 \cdot \sqrt{64}$$

$$T_{\text{eff}} = (2.657 + 0.066 \cdot q) \cdot 10^6, \quad \sigma_T = 86680$$

$$\text{erf}\left(\frac{1.001 \cdot q - (2.657 + 0.066 \cdot q)}{0.086680}\right) < 10^{-3}$$

$$\frac{0.935 \cdot q - 2.657}{0.086680} > 3.08$$

$$q > 3.127 \text{ frames}$$

【0070】

ここで、調整枠サイズ（ N ）を64サンプルに設定し、 ε を 10^{-3} として、許容Iフレームレートを算出する。そのストリームから統計を収集し、最大トリックモード速度を算出することによって、これを達成することが可能である。そのIフレーム統計は、すでに説明したように、前記ストリームに関連付けられた「ヒント」ファイルに保管することができる。 $q = 3.127$ という結果は每秒約9.6のIフレームであり、例えば3, 3, 3, 3, 3, 4, 3, 3といった不正規のGOPサイズを許容するものである。別の実施形態では、この結果を次の整数 $q = 4$ に切り上げ、每秒7.5のIフレームとする。

【0071】

このバッファ調整方法は、各トリックモードパケットからのパラメータ一式の計算を必要とする場合がある。これらのパラメータに指令を与え、Iフレームの選択、Iフレームのデータ収集、および制御変数の初期化をすることができる。

【0072】

Iフレームの選択に関し、一定の速度（例えば15x、30x、-10x...）でトリックモードストリームを生成するためにIフレームのシーケンスを決定することができる。前記Iフレームのシーケンスを、（1若しくはそれ以上の）速度、選択されたGOPサイズ（ q ）、および元のストリームから抽出された情報（フレームレート（ f_r ）およびそのストリームの毎秒の平均Iフレーム数など）に基づいて決定することができる。MP EG2ファイルが最初にヒントファイルに取り込みおよび保管されるときに、ヒント過程の一部として I_r を算出することができる。

【0073】

さらに理解を深めるために、以下に例示的な実施形態を提供する。毎秒平均2つのIフレームでありトリックモードストリームが毎秒10のIフレームで生成されると仮定し、すべてのIフレームが選択される場合、前記トリックモードストリームは5xという速度

10

20

30

40

50

で生成されることになる。あるいは、1つおきに（すなわち2のインクリメントで）Iフレームが選択される場合、10xのトリックモードストリームの生成が可能である。最後のIフレームから最初のIフレームにかけて1つおきに（逆順に-2のインクリメントで）Iフレームを選択すると、-10xのトリックモード速度が得られる。

【0074】

トリックモード速度がありIフレーム/秒(I_r)の平均を有するビデオストリームの場合、 $i = sq I_r / f_r$ としてその指数インクリメント(i)を算出することができる。前記指数インクリメントを用いて、同じく変数であるIフレーム指数(x)のシーケンスを算出することができる。実際のIフレームは、以下の例で提供される指数のシーケンスを切り捨てることによって得られる。

10

【0075】

$I_r = 2$ のIフレーム/秒、 $b = 3$ または10のIフレーム/秒(f_{ps})および $f_r = 30$ フレーム/秒(f_{ps})。トリックモードをIフレーム番号600から（すなわち映画の開始後約5分後）-16x（すなわち高速巻き戻し）の速度で実行する場合、Iフレーム指数のシーケンスは600であり $I = 3$ 。 $2 = -16x3x2/30$ となる。よって、作成される指数のシーケンスは600.0, 596.8, 593.6, 590.4, 587.2, 584.0等となる。また、このバッファ調整アルゴリズムのために選択されるIフレームのシーケンスは600, 597, 594, 590, 587, 584等である。

【0076】

20

トリックモード再生速度がより小さい場合、例えば4倍小さい場合は、結果的に得られる指数インクリメントが1.0未満となり、フレームの繰り返しが生じる場合がある。これらのケースでは、算出された指数インクリメントに基づくIフレーム選択中にGOPサイズ(q)を修正することができる。例えば、前記数値を用い、GOPサイズ(q)を3.75に修正し、4.0に切り上げることができる。これにより、毎秒の平均Iフレーム数は、毎秒10のIフレームから毎秒7.5のIフレームに減少する。これにより、その指数インクリメントは約 $i = 1.067$ になる。

【0077】

また、一部の実施形態で、GOPサイズ(q)を変数または浮動小数点として扱うことができ、従ってその指数インクリメントは1.0に切り上げられ、 q は例えば3.75など非整数の値を採ることができるものと理解される。これにより、4, 4, 4, 3等といったサイズのGOPのシーケンスが作成される。

30

【0078】

Iフレームのデータ収集に関しては、Iフレームのシーケンスがいったん決定されると、Iフレームに関する情報の収集および一部のデータ構造の初期化（例えばトリックモードパケット毎に1若しくはそれ以上）が可能である。適切なIフレームのエントリを指すだけで、そのヒントファイルから前記データを入手することができる。

【0079】

以下の説明は、使用可能なデータ情報のタイプのいくつかの例を提供する。「開始」データを収集することができる。開始データは、Iフレームに関連付けられたPESヘッダを含むトランスポート・ストリーム・パケットのオフセットである。これは、Iフレームの開始点のオフセットである場合がある。「終了」データも収集することができる。終了データは、ファイル内のIフレームの最後のオフセットである場合がある。これは、最後のIフレームのビデオデータの通過地点でのオフセットである。前記開始オフセットと終了オフセットの間に、他の非ビデオのトランスポート・ストリーム・パケットがファイル内に存在可能であると理解される。これらのパケットをストリーミング前にヌルに変換することができる。

40

【0080】

「サイズ」データも収集することができる。サイズデータは、（終了から開始を引いた）差分すなわち送信可能であるIフレーム全体を含むデータ量として算出することができ

50

る。「タイムコード」データも収集することができる。タイムコードデータは、ストリーム中の現在のタイムコードを後にクエリすることになる他のコンポーネントとのインタフェースを提供することができる。前記タイムコードはGOPヘッダの中にある場合があり、ヒント過程中に抽出することができる。

【0081】

「ファイルPCR」データも収集することができる。ファイルPCRデータを前記開始オフセットに関連付け、ストリーミングソフトウェアがPCR再スタンプを実行できるようにすることができる。「ファイルDTS」データを収集することができ、これを元アセットにあるIフレームに関連付けることによりDTS再スタンプを実行し、通常再生からトリックモードへ移行して再び通常再生に戻る間の滑らかなバッファ移行を実行することができる。「ファイルPTS」データを収集することができ、これを元アセットにあるIフレームに関連付けることによりプレゼンテーション・タイム・スタンプ(Presentation Time Stamp: PTS)を実行し、全移行のフレーム間隔を維持することができる。

【0082】

「CC開始」および「CC終了」データを収集することができる。CC開始データは、その開始パケットのトランスポートストリーム継続カウンタである。CC終了は、その終了パケットの継続カウンタである。CC開始およびCC終了データは、CC再スタンプを実行するために一部の実施形態で必要な場合がある。

【0083】

「次フィールド」データを収集することができる。次のフィールドデータを用い、「repeat first field」フラグを使ってIフレームをエンコードし、2つでなく3つのフィールドをIフレームが含むようにすることができる。移行中にフィールドのシーケンスを維持するために、移行の後に第1のダミーBフレームにフィールド調整機構を用いることができる。

【0084】

制御変数の初期化に関しては、一部のストリーミング変数を追跡し続けることが好ましい場合がある。例えば、ストリームオフセット、ストリームPCR、ストリームDTS、およびストリームPTSである。前記ストリームオフセットは、そのストリーミングソフトウェアが作成する合計データ量である場合があり、そのファイルオフセットとは異なるものである。前記ストリームPCRは、PCR再スタンプ機構の後に出力ストリームで観察される実際のPCRである場合がある。前記ストリーミングソフトウェアは一定のビットレートで動作するので、ストリームオフセットのインクリメントをストリームPCRインクリメントと関連付けることができる。

【0085】

一部のフィールドを初期化することができる。例えば、「フレーム」をフレーム数とすること、あるいは現在トリックモードパケットのGOPサイズとすることが可能である。最初は前に算出された数qに設定しておき、この数を必要(例えばPフレームの挿入の機構)に応じて増していくことができる。qを浮動小数点または変数として実行する場合、下記に示すエラー伝播機構に基づいて前記フレーム数を算出することができる。

【0086】

【数11】

$Frames[i] - truncate(q + q_error);$

$Q_error = q + q_error - frames[i];$

If $q = 2.6666\dots$, the sequence produced would be: frames = { 2,3,3,2,3,3,etc. }

【0087】

パケット・サイズ・フィールドは、合計パケットサイズを表す。前記パケットサイズは前パケットのタイムスロットに基づき、非整数のTSパケットを作成する。このパケット

10

20

30

40

50

サイズは前トリックモードパケットからの超過分を考慮するエラー伝播機構によって修正することが可能である。この時点で、例えば188バイト（トランスポートストリームのパケットサイズ）または1316バイト（UDPパケットサイズを超過するMPEG2）など、ある程度の粒度をトリックモードパケット全体に実行する場合がある。

【0088】

そのストリーミングエンジンの状態によって、最初のトリックモードパケットの扱いが異なる場合がある。例えば、前記ストリーミングエンジンが休止中（すなわち一時停止および停止）であれば、そのデコーダは休止中なのでバッファアンダーフローが生じる危険がない場合がある。そのパケットサイズをゼロに設定し、当該調整方法によって修正することができる。あるいは、そのストリーミングエンジンが再生中であれば、通常で表示された最後のフレームのDTSと現在PCRの差分に基づいて最初のタイムスロットを算出することができる。言い換えれば、そのバッファから「通常再生」データが実質的に取り除かれた後に、最初のトリックモードパケットをデコードすることができる。これは、「再生」から「トリックモード」への移行に使用される「デバッファリング（debuffering）」法である場合がある。

10

【0089】

この例では、前記トリックモードパケットサイズを算出するシーケンスは、前パケット（frames[j-1]*f_r）のタイムスロットに基づく。前記パケットサイズおよびパケット超過分は浮動小数点または変数であり、次のように算出することができる。

【0090】

20

【数12】

If(j=0 and StreamState=STOPPED) /* First trickmode packet after a full stop */

 packet_size[j] = 0;

if(j=0 and StreamState=PLAYING) /* First trickmode packet after playing */

 packet_size[j] = ((StreamDTS - StreamPCR)/27000000.0)*(b_r/8.0);

else

 packet_size[j] = (frames[j-1]/f_r)*(b_r/8) + packet_excess[j-1];

 packet_excess[j] = packet_size[j]-truncate(packet_size[j]/granularity)*granularity;

 packet_size[j] = packet_size[j]-packet_excess[j];

30

【0091】

パケット超過分は、トリックモードパケットに対するある特定の粒度の実行に使われる制御変数である場合があり、Pフレーム挿入法によって用いることにより、パケットサイズを拡張する場合にその粒度を維持することができる。データサイズは合計データサイズを表し、Iフレーム、ダミーBフレームおよびPフレーム、PAT、PMT、および前記トリックモードパケットの構築に関連付けられるオーバーヘッドを含む。データサイズは、以下のように算出することができる。

【0092】

40

【数13】

data_sizeD] = size[j] + (frames[j]-1)*P + OH.

【0093】

「Bw_balance」は、バッファ調整に利用可能な帯域幅を表す。これは、前記packet_sizeと前記data_sizeの差分である場合がある。未使用の帯域幅をスルで満たし、そのストリームのビットレートを維持することができる。

【0094】

一部の実施形態では、トリックモードパケットの最小限のサイズが課される。これは、調整のために利用可能な帯域幅を、算出された実際に利用可能な帯域幅より少なくするこ

50

とによって達成できる。一部の実施形態では、前記トリックモードパケットのサイズを決定するときに、ハードウェアの制限によって課されるトリックモードパケット間の最小「シークタイム」または最小遅延のようなハードウェア制約が考慮される場合がある。また、 q はトリックモード帯域幅の可能な使用方法についての一部の仮定を変えるので、 q の計算にこれらの考慮事項が含まれる場合がある。

【0095】

加えて、調整に利用可能となるスルパケットに特定の粒度が課される場合があり、例えばUDPパケットのトランスポートストリームに対して1316の粒度が課される。これは、ストリーミングソフトウェアまたはハードウェアの具体的な実装に依存する場合がある。利用可能な帯域幅は以下のように算出することができる。

10

【0096】

【数14】

If(data_size[j]<min_size)

then bw_balance[j] = packet_size[j] - min_size;

else

bw_balance[j] = packet_size[j] - data_size[j];

if(bw_balance[j]<0)

bw_balance[j] = granularity*(truncate(bw_balance[j]/granularity — 1);

20

else

bw_balance[j] = granularity*truncate(bw_balance[j]/granularity).

【0097】

Bw_balance は、負の値は超過サイズのパケットを表すものと仮定することが多い。前記負の値は、そのトリックモードパケットに不足しているため他のトリックモードパケットから取り出す必要のある帯域幅の量である場合がある。これは、小パケットからの利用可能な帯域幅を使い、大パケットが必要とする帯域幅を均衡することによって達成することができる。その統計解析を用い、その調整枠にある利用可能な帯域幅の全体的なバランスが、前記のパラメータ ε に依存して正 ($\sum bw_balance[i] > 0$) となるようにすることができる。

30

【0098】

「ストリームオフセット (Stream offset)」は、パケットに期待される現在ストリームのオフセットである。前記現在パケットが最初に送信されるものである場合、そのパケットは上述のようにストリーミングエンジンから取り出されるストリームオフセットである場合がある。

【0099】

【数15】

if(j=0)

stream_offset[j] = StreamOffset;

40

else

stream_offset[j] = stream_offset[j-1] + packet_size[J-1].

【0100】

「ストリームPCR (Stream PCR)」は、ディスクから検索したビデオデータの正確なPCR再スタンプをするために必要な場合がある。

【0101】

【数 1 6】

```
if(j=0)
    stream_PCR[j] = StreamPCR;
else
    stream_PCR[j] = StreamPCR[j-1] + round(27,000,000*packet_size[j-1]*8/br);
```

【0 1 0 2】

「ストリーム D T S (S t r e a m D T S)」は、そのトリックモード G O P の一部として送信される I フレームのデコード時間を表す。D T S および P T S に 3 0 0 を掛けることにより、それらを P C R と同じ時間ベースにすることができる。再生からトリック 10
モードへの移行がある場合は、前記 D T S を 2 分の 1 フレーム修正することによって前述のようなフィールド調整を行う必要がある場合がある。そのような移行がない場合は、前記トリックモードパケット D T S は次のように算出される。

【0 1 0 3】

【数 1 7】

```
if(j=0)
    stream_DTS[j] = StreamDTS;
else
    stream_DTS[j] = StreamDTS[j-1] + round(27,000,000*frames[j-1]/fr);
```

20

【0 1 0 4】

「ストリーム P T S (S t r e a m P T S)」は、その I フレームの正確なプレゼンテーションタイムを次のように表す。

【0 1 0 5】

【数 1 8】

```
if(j=0)
    stream_PTS[j] = StreamPTS;
else
    stream_PTS[j] = StreamPTS[j-1] + round(27,000,000*frames[H]/fr);
```

30

【0 1 0 6】

「バッファレベル」は、そのデコーダでの最大バッファレベルである場合があり、前記デコーダがビデオデータの最終ブロックを受信したときに達成され得るものであり、そのオフセットは次のように算出される： $peak_offset[j] = stream_offsetQ + data_size[j]$ 。

前記バッファは、前 G O P からのいくつかのダミー B フレームおよび P フレームを含むことができ、最大バッファレベルを次のように算出することができる。 $bufferJewel[j] = size[j] + (frames[i] - 1) * P + (frames[i - 1] - 1) * P$ 。現在トリックモードパケットの伝送中に前 G O P からのダミー B フレームおよび P フレームが消費されるので、実際のバッファレベルは上記数値より低くなる場合がある。前記バッファレベルは、オーバーフロー対策として多めに予測される場合がある。現在トリックモードパケットの D T S (すなわち I フレーム D T S) で、前 G O P からのデータは消費済みである場合があり、その場合、そのバッファレベルは次のように算出することができる： $bufferJewel[j] = size[j] + (frames[j] - 1) * P$ 。

40

【0 1 0 7】

図 2 は、デコードバッファの経時変化を図示する。本発明の実施形態は、I フレームがデコードされる前に、D T S [j] によって与えられた時点で前記バッファレベルを制御 50

することができる。これは、バッファ調整によって前記バッファレベルのピークがこの位置に移動するからである。前トリックモードパケットからのダミーBフレームおよびPフレームのサイズは比較的小さく、しかも特に前記Iフレームのサイズは前述したように多めに予測されているため、そのデコード時間を表す「Stream DTS」を使う式を、リスクを伴わずに用いることができる。あるいは、そのバッファがオーバーフローに到達するのを確実に防ぐために、「Stream offset」が現在ストリームである式を使うことができる。

【0108】

説明したように、図5は、利用可能なタイムスロットインターバルを再配列するためにトリックモードパケットをどのように調整するかを図示している。説明したように、当該制御変数bw_balance[j]が負であるということは、最初に保留したタイムスロットでそのパケットを伝送できないことを示す。本発明の進歩的な方法は、図5が示すように、前記パケットサイズを変更および拡張し、前パケットからの利用可能な帯域幅を消費すると同時に、前記パケットを短縮する。

10

【0109】

前記パケットサイズを変更および拡張し、前パケットからの利用可能な帯域幅を消費すると同時に、前記パケットを短縮するCコードの例を以下に示す。

【0110】

【数19】

```
int bw_adjust;
for(int j=n-1; j>0; j--)
{
    bw_adjust = bw_balance[j];
    if(bw_adjust<0)
    {
        packet_size[j] = bw_adjust; // Extends the packet to perfectly fit the trickmode data
        bw_balance[j] = 0; // No BW available, no BW needed
        stream_offset[j] += bw_adjust; // Shifts the packet to allow buffering before DTS is due
        stream_PCR[j] += 27000000*(bw_adjust*8/bitrate); // Adjust the packet PCR
        packet_size[j-1] += bw_adjust; // Shorten the previous packet by the same amount
        bw_balance[j-1] += bw_adjust; // Consume the BW from the previous packet
    }
}
```

20

30

【0111】

このコードセグメントは、利用可能な帯域幅をその調整枠内で再配列させることができ、そのトリックモードパケットのデコード時間(DTS)が期限切れになる前に、それらを完全に伝送させることができる。

【0112】

さらに、帯域幅の再配列のほかに、バッファレベルも考慮することが望ましい場合がある。また、一部の実施形態では、最初のトリックモードパケットのbw_balance[j]は負である場合があり、それがそのシーケンスにおける最初のパケットであるため、割り当てる帯域幅の元となる前パケットがない場合がある。以下のように、Pフレーム挿入および移行法を使用することができる。さらに、当該制御法は各トリックモードパケットのパラメータを計算するが、必ずしもそのストリームを生成するとは限らない。これは、前記トリックモードストリームを生成するための調整法を用いるストリーミングエンジンによって達成される。

40

【0113】

図6は、トリックモードパケットの変化がバッファレベルに与える影響を表す図である。図6はトリックモードパケットの変化に対してバッファレベルが受ける影響を説明するものであるが、一部の実施形態では、他の帯域幅の制御法および他のデータ操作法はバッファ制御を必要とする場合があると理解される。

【0114】

図6が示すように、上の枠600は、そのトリックモードパケットを変更する前のバッ

50

ファレレベルを反映しており、下の枠 6 0 1 は、帯域幅の制御に従って前記トリックモードパケットを変更した後のバッファレレベルを反映している。前記上の枠 6 0 0 が示すように、図のパケット 6 0 2 はポイント 6 0 3（すなわちそのデコード時間切れ）での DTS の後まで完全に受信されない。前記トリックモードパケットの変化によって、DTS で最大バッファ・ストレージ・レベル 6 0 4 が作成される。さらに、最大バッファレレベルを、変更されたトリックモードパケット 6 0 5 のデータ量と同等にすることができる。その差分（ $data_size[j] - packet_size[j]$ ）は、 $DTS[j-1]$ の時点でそのデコーダバッファにて準備され、前記トリックモードパケットがデコードされる前にその完全なバッファが可能となる。

【0115】

次のコードは、最大バッファレレベルの予測の一例に過ぎない。

【0116】

【数 2 0】

```
int bw_adjust;
for(int j=n-1; j>0; j--)
{
    bw_adjust = bw_balance[j];
    if(bw_adjust<0)
    {
        packet_size[j] -= bw_adjust; // Extends the packet to perfectly fit the trickmode data
        bw_balance[j] = 0; // No BW available, no BW needed
        stream_offset[j] += bw_adjust; // Shifts the packet to allow buffering before DTS is due
        stream_PCR[j] += 27000000*(bw_adjust*8/bitrate); // Adjust the packet PCR
        packet_size[j-1] += bw_adjust; // Shorten the previous packet by the same amount
        bw_balance[j-1] += bw_adjust; // Consume the BW from the previous packet
        buffer_level[j-1] -= bw_adjust; // Estimate Buffer level at the previous packet
    }
}
```

【0117】

バッファレレベルは、超過サイズのフレーム（すなわち保留タイムスロットを超過するサイズのフレーム）が送信される度に増す場合がある。超過サイズのフレームの長いシーケンスを調整するとき、バッファオーバーフローが生じる場合がある。利用可能なタイムスロットより大きいデータ量がいったん決定されたら、そのバッファオーバーフローを取り込むためのアクションを採ることが望ましい。これは、様々な方法を任意に用いて達成することができる。以下の例は、本実施形態で考えられるすべての方法を排除することを意味するものではない。

【0118】

バッファオーバーフローを処理する 1 つの方法は、P フレーム挿入を含む場合がある。P フレーム挿入は P フレームを追加するものであり、よって、付加的な帯域幅を生成するために前 GOP のサイズを拡張する。付加的 P フレームの挿入を行う方法は多数ある。付加的な帯域幅を生成するために P フレームを挿入する 1 つの方法について説明する。しかし、ここで開示する実施形態は、この方法に限定されない。以下は一例である。

【0119】

大型のトリックモードパケットを持つ 3.75 メガビット／秒のビデオストリームを有するトリックモードストリームを送信するには、6 つのフレーム期間が必要である場合がある。しかし、そのタイムスロットは前パケットの GOP サイズによって決定されるため、4 フレームしかないという場合がある。上述したように、前記パケットを変更し、2 フレーム期間拡張して、伝送に必要な付加的期間を取り込むことができる。説明したように、前記トリックモードパケットを 2 フレーム分変更することにより、その前パケットのピーク・バッファ・レベルが約 30 kb 増す場合がある。100 kb というサイズのビデオバッファおよび 90 kb というサイズの前パケットの場合、さらに 30 kb のバッファを試みることによってバッファオーバーフローが生じることになる。

【0120】

1 実施形態では、ダミー P フレームを追加することができる。例えば、各々が P フレームにつき約 1 kb である 2 つの余分な P フレームを追加すると、その前 GOP サイズを 6 に上げることができる。そのバッファレベルは 92 kb まで 2 kb ずつだけ増し、100 kb という制限内に留まる。前記前 GOP に 2 つの付加的ダミー P フレームを加えることにより、現在パケットサイズは拡張され、「超過サイズ」パケット全体の伝送が可能となる。言い換えれば、この方法は、前のフレームを画面に 2 フレーム期間多く保留することにより、前記超過サイズフレームの完全な伝送を可能にするのである。さらに、挿入された各付加的 P フレームは後続のトリックモードパケットを約 1 フレーム期間拡張することができるので、付加的な帯域幅を得ることが可能となる。この例では、各々約 1 kb のダミー P フレームが、1 フレーム期間に伝送可能なデータ量である約 15 kb の帯域幅を生成する。

10

【0121】

また、ダミー P フレームを挿入することにより、その I フレームレート全体を減少することができ、その調整枠を 1 フレーム拡張することができる。バッファオーバーフローが検出される度に余分な 1 フレームを挿入することができるコードセグメントの例を以下に示す。

【0122】

【数 2 1】

```

int bw_adjust;
for(int j=window-1; j>0; j--)
{
    bw_adjust = bw_balance[j];
    if(bw_adjust<0)
    {
        packet_size[j] -= bw_adjust; // Extends the packet to perfectly fit the trickmode data
        bw_balance[j] = 0; // No BW available, no BW needed
        stream_offset[j] += bw_adjust; // Shifts the packet to allow buffering before DTS is due
        stream_PCR[j] += 27000000*(bw_adjust*8/bitrate); // Adjust the packet PCR
        packet_size[j-1] += bw_adjust; // Shorten the previous packet by the same amount
        bw_balance[j-1] += bw_adjust; // Consume the BW from the previous packet
        buffer_level[j-1] = bw_adjust; // Estimate buffer level at the previous packet

        /* perform p-frame insertion until the buffer overflow is fixed */
        while(buffer_level[j-1]>video_buffer_level)
        {
            double increment;

            // Insert a P-frame in the packet where the overflow was detected.
            frames[j-1]++; // Insert a p-frame in the previous packet
            data_size[j-1] += P; // Account for an extra p-frame...
            bw_balance[j-1] -= P; // Take it from the bandwidth balance
            buffer_level[j-1] += P; // Update the buffer level estimation

            /* Use all the bandwidth created to revert the buffer overflow */
            increment = (1/frame_rate)*(bitrate/8); // Calculate the amount of bandwidth created
            packet_size[j-1] += increment; // Restore the packet size (bw insertion is here!!)
            bw_balance[j-1] += increment; // Restore the bandwidth (and here!!)
            buffer_level[j-1] -= increment; // Restore the buffer level

            /* Shift the current and subsequent packets by one frame */
            for(int k=j; k<window-1; k++)
            {
                stream_offset[k] += increment; // Move packet back
                stream_PCR[k] += 27000000*(increment*8/bitrate); // Move packet back
                stream_DTS[k] += 27000000/frame_rate; // Account for an extra frame
                stream_PTS[k] += 27000000/frame_rate; // Account for an extra frame
            }
        }
        /* End of the p-frame insertion algorithm */
    }
}

```

【0 1 2 3】

また、一部の実施形態は、例えば少なくとも1つのトランスポート・ストリーム・パケット（例えば典型的に188バイト）など、トリックモードパケットの粒度の解析を考慮する場合もある。Pフレームの挿入によって生成される帯域幅は、次の等式によって決定される：

$$\text{インクリメント} = (b / 8) * (1 / f r) ;$$

【0 1 2 4】

上記の例において、インクリメントは15,625バイトである。各トリックモードパケットのサイズをエラー伝播法によって元々決定したので、同様の方法を、より正確なメモリ量を計算するために考慮することができる。例えば、トリックモードのGOPサイズをq=4、ビデオバッファのサイズを110kb、トリックモードのデータサイズを25kb、90kb、40kb、70kb、80kb、80kb、および80kb、パケット粒度を1316バイト（例えば1つのUDPパケットに7つのトランスポート・ストリーム・パケット）として、これを決定することができる。

【0 1 2 5】

算出されるタイムスロットは62,500バイト=4*15,625バイトとなる。よ

10

20

30

40

50

って、このエラー伝播法は、6 1 8 5 2、6 1 8 5 2、6 3 1 6 8、6 1 8 5 2、6 1 8 5 2、6 3 1 6 8、6 1 8 5 2というパケットサイズのシーケンスを生成することができる。また、この方法は、3 5 5 3 2、-3 1 5 8 4、2 1 0 5 6、-1 0 5 2 8、-2 1 0 5 6、-1 9 7 4 0、-2 1 0 5 6というbw_balance値を持つ0, 6 1 8 5 2、1 2 3 1 6 4、1 8 6 3 3 2、2 4 8 1 8 4、3 1 0 0 3 6、および3 7 3 2 0 4というパケットオフセットを生成することができる。トリックモードのデータサイズと同一であると最初に予測されたバッファレベルで、最終パケット（指数6、0ベースの指数）から始まるバッファ調整を適用することにより、以下に示すようにそのヌルサイズを差分（packet_sizeからdata_sizeを減算して得られる差分）として算出することができる。（注記：ここではタイムスタンプは除外される（PCR、DTS、およびPTS））。

10

【0 1 2 6】

【数 2 2】

```
packet_size[6] -= bw_balance[6] => packet_size[6] = 61852 + 21056 = 82908
bw_balance[6] = 0;
stream_offset[6] += -21056 => stream_offset[6] = 373204 - 21056 = 352148
packet_size[5] += -21056 => packet_size[5] = 63168 - 21056 = 42112
bw_balance[5] += -21056 => bw_balance[5] = -19740 - 21056 = -40796
buffer_level[5] -= -21056 => buffer_level[5] = 81920 + 21056 = 102976
```

【0 1 2 7】

第2の段階では、以下が行われる。

20

【0 1 2 8】

【数 2 3】

```
packet_size[5] -= bw_balance[5] => packet_size[5] = 42112 + 40796 = 82908
bw_balance[5] = 0;
stream_offset[5] += -40796 => stream_offset[5] = 310036 - 40796 = 269240
packet_size[4] += -40796 => packet_size[4] = 61852 - 40796 = 20876
bw_balance[4] += -40796 => bw_balance[4] = -21056 - 40796 = -61852
buffer_level[4] -= -40796 => buffer_level[4] = 81920 + 40796 = 122716 (buffer overflow!)
```

【0 1 2 9】

この時点で、余分なダミーPフレームをトリックモードパケット4に挿入し、そのGOPサイズを5に変更することができる。Pフレームサイズを1 3 1 6バイトと仮定すると、以下のシーケンスが提供される。

30

【0 1 3 0】

【数 2 4】

(P-frame insertion)

```
frames[4] = 5;
data_size[4] += 1316 => data_size[4] = 81920 + 1316 = 83236
bw_balance[4] -= 1316 => bw_balance[4] = -61852 - 1316 = -63168
buffer_level[4] += 1316 => buffer_level[4] = 122716 + 1316 = 124032 (looks a little worst, but wait!)
```

【0 1 3 1】

挿入された前記Pフレームによって、以下の付加的な帯域幅を提供することができる。

40

【0 1 3 2】

【数 2 5】

(P-frame insertion)

```
frames[4] = 5;
data_size[4] += 1316 => data_size[4] = 81920 + 1316 = 83236
bw_balance[4] -= 1316 => bw_balance[4] = -61832 - 1316 = -63168
buffer_level[4] += 1316 => buffer_level[4] = 122716 + 1316 = 124032 (looks a little worst, but wait!)
```

(Add bandwidth thanks to the P-frame inserted)

```
increment = (1/30)*(3750000/8) = 15625
packet_size[4] += 15625 => packet_size[4] = 20876 + 15625 = 36501
bw_balance[4] += 15625 => bw_balance[4] = -63168 + 15625 = -47543
buffer_level[4] -= 15625 => buffer_level[4] = 124032 - 15625 = 108407 (overflow is fixed!)
```

10

(Propagates the frame insertion to subsequent packets)

```
stream_offset[5] += 15625 => stream_offset[5] = 269240 + 15625 = 284865
stream_offset[6] += 15625 => stream_offset[6] = 352148 + 15625 = 367773
```

【0 1 3 3】

粒度が問題となる実施形態では、パケットサイズに対して上述のものと同一エラー伝播法を用い、新しいGOPサイズに基づいてパケットサイズを再計算することができる。

【0 1 3 4】

図7は、トリックモードストリームを作成するために使用可能なバッファ最適化の方法を図示する。図7が示すように、調整が必要なパケットは斜線模様で示されており、調整に成功したパケットは点描模様で示されている。上の枠700は、このバッファ最適化法が採用される前の、最初の計算でのトリックモードパケットを示す。第2の枠701は、最後の超過サイズフレームがどのように拡張および変更され、その前フレームがその利用可能な帯域幅を消費したかを示す。第2の枠701も、同一のパケットの変更がどのようにそのバッファレベルに影響し得るかを示す。第3の枠702は、調整に成功したパケットを示す。図8は、これらの方法を用いて生成されたトリックモードストリームの出力の例を提供する。

20

【0 1 3 5】

Iフレームベースの方法は「低遅延」モードで機能可能であり、従ってそのデコーダでバッファレベルが低く維持されると理解される。通常再生で映画を再開するために、前記デコーダはデータの0.5sから1.0sをバッファする場合がある。バッファレベルの相違がバッファアンダーフローを引き起こすことにより、画面の横揺れまたはちらつきが生じることがあり、一時的に完全に暗黒となることさえあり得る。

30

【0 1 3 6】

ファイルベースのトリックモードは、トリックモードファイルと正規ファイルの間で切り替わる場合がある。この方法は、その移行点でのバッファレベルの一致を要求することができるので、トリックモードファイルの生成時にバッファレベルを制御することができる。前記移行点で前記バッファレベルを調整するために、付加的ロジックを追加することができる。

【0 1 3 7】

1実施形態では、通常再生の再開時に、当該バッファ管理方法はより精度の高い移行のためにバッファレベルを増すことによって、最後のトリックモードフレーム（典型的に2～4フレーム）を修正する動作をすることができる。このいわゆる「スプライシング (s p l i c i n g)」法は前記最後のフレームの速度を下げ、再バッファリングに使用可能な超過帯域幅を生成し、ビデオバッファを通常レベルに戻すことができる。動きの感覚を中断しない方法でこの方法を実施することができるので、その移行は比較的シームレスである。

40

【0 1 3 8】

再生シーケンス後にトリックモードの再生を開始するために、ビデオバッファに保管可能な再生データはそのデコーダによって消費される。これは、そのトリックモードストリームのデコードの開始時に起き、前記ビデオバッファに存在するデータは、そのトリック

50

モード・ストリーミング・エンジンによって生成されるデータである。最初のトリックモード packets を、その DTS を再生中の最後のフレームの DTS とフレーム間隔に設定して送信することができる。また、その PTS を、表示されている最後のフレームに 1 フレームを足した PTS に設定することができる。前記最後のフレームの「repeat first field」フラグを 1 に設定すれば、PTS および DTS は 1 フレーム期間の半分でインクリメントする。

【0139】

前記デバッファリング法の実施は、その最初のトリックモード packets のサイズを現在位置（PCR）からその再生バッファが空になるまで伝送可能なデータ量として設定することによって可能であり、その最初のトリックモードフレームは、次の等式を用いて予測される（例えば上述の DTS）：

$$\text{frame_size}[0] = ((\text{S_streamDTS} - \text{StreamPCR}) / 270000.0) * (\text{BI} / 8)。$$

一部のビデオストリーム（例えば MPEG2）は、3：2 のプルダウンを実行する方法として、「repeat first field」および「top field first」といったいくつかの特別なフラグを使う場合がある。それらのストリームでフィールド継続性を維持するために、ある特定の方法をその移行シーケンスに適用することができる。

【0140】

例えば、1 つの方法は、表示された最後のフレームで「top field first」が 0（最初にボトム）に設定され、かつ「repeat first field」が 0 に設定されている場合に使うことができる。これは、その最後のフレームが前記トップフィールドの表示を終了したことを示す。また、前記最後のフレームの「top field first」が 1（最初にトップ）、その「repeat first field」が 1 に設定されていれば、表示された最後のフィールドがそのトップフィールドであったことを示す場合がある。いずれの場合も、次に期待されるフィールドは、そのボトムフィールドである場合がある。これらの方法は「top field first」であると仮定するので、フィールド調整フレームを挿入することができる。

【0141】

これらの方法は、ダミー B フレームである場合があるその最初のトリックモードフレームで「top field first」というフラグを 0 に設定し、「repeat first field」というフラグを 1 に設定することによって実行可能である。このシーケンス（ボトム、トップ、ボトム）は、その再生シーケンスからのフィールドシーケンシングを維持するだけでなく、後続フレームを前記トップフィールドで開始することを可能にする。

【0142】

このフィールド調整法はその GOP サイズを 2 分の 1 フレーム（すなわち 1 フィールド分）拡張する。ディスクから読み込まれたすべての I フレームが確実に正しいフィールドシーケンシングを有するようにするために、前記 I フレームがディスクからメモリに読み込まれた後に実行される再スタンプ法によって、前記 I フレームにおいて前記「top field first」フラグを 1 に設定し、前記「repeat first field」フラグを 0 に設定することができる。

【0143】

一部の実施形態では、その再生シーケンスが比較的低いバッファレベルで動作する場合、その最初のトリックモード packets を伝送する時間が十分でない場合がある。前記再生シーケンスからの最後のフレームが消費される時点で、前記最初のトリックモード packets がまだ伝送中である場合がある。一つの解決策は、前記トリックモード packets の開始部分に P フレームのシーケンスを添付することであり、これは前述の P フレーム挿入法に類似している。これら P フレームはそのトリックモード GOP の一部である必要はなく、その前 GOP（すなわち再生シーケンス）を拡張し、いくつかのフレーム間隔のためにそ

10

20

30

40

50

のデコーダに最後の画像を繰り返させるので、そのトリックモード packets を完全に伝送することができる。

【0144】

さらに、トリックモードから通常再生に戻る移行に用いられる方法は、そのトリックモードシーケンスの最後のGOPを拡張して利用可能な余分な帯域幅を生成することによって達成可能である。前記利用可能な帯域幅は、例えば高遅延モードで動作している新しい再生シーケンスからのビデオデータの再バッファリングに用いることができる。

【0145】

前記再バッファリング法は、そのデコーダがダミーPフレームの再生に使用されている間、そのバッファが前記新しい再生シーケンスを保管できるよう十分な時間にわたりその最後のトリックモード packets を保持することを試みる。前記再バッファリング法は、前記最後のトリックモード packets のGOPサイズを増すことによって、その帯域幅を徐々に生成することができる。選択されたGOPサイズが4であれば、前記最後のトリックモード packets のGOPサイズは5, 7, 10等となり、視覚的印象としては完全停止ではなくスローダウンとして目に映る。

【0146】

利用可能な帯域幅の合計が必要な帯域幅以上になるまで前記トリックモード packets を挿入することが可能であり、よって、その再生シーケンスの完全な再バッファリングが可能になる。前記最後のトリックモード packets について、そのIフレーム選択機構も変わる場合がある。これは、リクエストされた再生オフセットから離れすぎないようにするために必要である場合がある。その指数インクリメントを最小の1.0に設定することにより、各移行トリックモード packets がそのストリームを約 $1/I_r$ だけそのリクエストされた位置から離れて移動するようにすることができる。典型的な結果は、再バッファリングを可能にするには約3~5の移行 packets が必要であることを示しており、これは $I_r = 2$ Iフレーム/秒であるストリームでのリクエストされた再生位置からわずか1.5~2.5 sだけ離れていることを表す。

【0147】

別の可能な方法は、正確な再生オフセットから始まるトリックモード packets のシーケンスの利用可能な帯域幅を算定することであるが、この場合は指数インクリメントを-1.0として逆戻りする。その純帯域幅が必要な帯域幅以上になると、最後の packets はその移行シーケンスで使われる最初の packets になる。この方法により、その移行シーケンスは確実にリクエスト通りの再生オフセットで終了することが可能である。

【0148】

同じ方法で、トリックモードが負の速度（すなわちREW）で再生されると、十分な帯域幅が利用可能になるまでIフレームが前方（インクリメントは+1.0）から選択される場合がある。ここで、移行 packets のシーケンスを、実際の再生シーケンスが始まる位置の逆向きに現在位置から取り出すことができる。移行トリックモード packets のシーケンスがいったん決定され、このバッファ最適化法にロードされると、データサイズを0に、トリックモード packets のサイズを0に設定し、bw_balanceをバッファリングに必要なデータ量に設定して、この調整法に「仮想トリックモード packets」を挿入することが可能となる。

【0149】

前記バッファ最適化法の使用により、前記トリックモード packets の利用可能な帯域幅の消費および前記移行 packets の変更が生じ、新しい再生シーケンスのためのスペースが生成される。この方法は、図8の下グラフが示すような、6, 7, 8, 9というGOPサイズの4つの移行 packets が観察され得るバッファ移行を可能にする。この方法を用いてトリックモードから再生に移行することにより、そのフレームシーケンスの中断を起こすことなく、「スローダウン」の印象を与えることができる。

【0150】

上述の方法を早送りおよび巻き戻しに利用することができる。この例では、ビデオ・オ

10

20

30

40

50

ン・デマンド（VOD）サーバは、例えばユーザがFFボタンを解除したときに、セットトップボックスからフィードバックを受信することができる。しかし、前記サーバのバッファ管理アルゴリズムは、通常再生に戻る移行中に遅れを起こす場合がある。そのバッファを満たすために、入ってくるフレーム数が再生中のフレーム数を超過し、ユーザからの信号の受信後にバッファを満たすためにフレームの伝送が強化される場合がある。その時点で、通常のストリームに切り替わる場合がある。

【0151】

当該方法は、いくつかの余分なBフレームおよびPフレームを生成することによって最後のフレームの速度を下げることで、そのデコーダのバッファを満たすものであり、前記BフレームおよびPフレームは、比較的小さく容易に生成され、かつ表示時間未滿に伝送可能なものである。「スローダウン」は、毎秒送信されるIフレーム数を減少することができるため、一定の30fpsであることが通常要求される実際のフレームレートへの影響はない。あらゆるIフレームとともに送信されるBフレームおよびPフレームの数（これらは比較的小さい）を増やすことにより、平均ビットレートを前記トリックモード法によって減少することができる。次に、余分な帯域幅を使って、そのバッファを通常レベルに修復することができる。

10

【0152】

いくつかの例では、一時停止および再開モードは通常のトランスポートストリームを利用する移行であるため、これらに上述の方法を使わない場合がある。一方、ジャンプでは、その移行点でバッファレベルが異なる場合があるため、バッファ調整が必要である。また、トリックモードに適用されたものと同じ方法をジャンプに適用することもでき、これは、デバッファリングを可能にすることによって、または再バッファリングを可能にするためにダミーBフレームまたはPフレームを挿入することによって行われる。バッファ制御は、PCR、DTS、PTSといったストリームパラメータの十分な調整に関連するため、これら制御変数の一致を確保することによって速度移行およびジャンプを実施する必要がある場合がある。

20

【0153】

その移行点後のバッファレベルが、その前のバッファレベル未滿である場合、ヌルのシーケンスを挿入することによってバッファレベルを十分なレベルにすることができる。この方法は、前記移行後の元のストリームの差分（DTSからPCRを引いた差分）を維持する。この新シーケンスのデコードは、その前シーケンスの最後のフレームがデコードされてから約1フレーム後に始まる。フレームのシーケンスに中断が入らないようにするために、前記移行後の最初のパケットのPTSを、その前シーケンスの最後のフレームが表示されてから1フレーム後に起こすことができる。

30

【0154】

このジャンプ法は、いくつかのGOPが開いている場合があることを考慮することもできる。言い換えれば、前記新しい再生シーケンスのIフレームの前に表示される最初のBフレームは、伝送された前GOPからのフレームをフォワードレファレンスすることができる。さらに、前記ジャンプ法は、フィールドシーケンシングを修正し、シーケンスに2つのトップフィールドまたは2つのボトムフィールドが存在しないようにすることができる。間違ったフィールドシーケンシングは、例えばセットトップボックスに、好ましくない画面の横揺れを起こす場合がある。

40

【0155】

前記ジャンプ法は、その前GOPの完全な送信を確保するスプライシング法を含むことができる。それにより、そのデコーダの受信バッファに不完全な画像またはシーケンスが存在しないようにする。また、前記スプライシング法は、その新シーケンスのDTSからPCRを引いた差分が現在ストリームのDTSからPCRを引いた差分（すなわち古いシーケンスからのもの）より小さいことを判断し、それにより、そのバッファレベルの低下が必要であることを判断することができる。さらに、前記スプライシング法はそのIフレームを検索し、ダミーBフレームのシーケンスを添付することができる。ダミーBフレーム

50

の数を、その I フレームの前の元のシーケンスにある B フレームの数と一致させることができる。これにより、開いた GOP に関して説明した問題を避けることができる。

【0156】

前記スプライシング法は、「top field first」フラグおよび「repeat first field」フラグを最初のダミー B フレームの中で適切に設定することによってフィールドシーケンシングを調整することができる。フィールド調整が必要であれば、DTS および PTS を適宜調整することができる。この例では、その新しい再生シーケンスは 1 つの I フレームを有することができ、次に B フレームを再スタンプして前記新しい再生シーケンス (PBBPBBP 等) の残りとも一致させることにより、中断を避けることができる。前記スプライシング法は、バッファレベルを調整するために挿入可能なヌルの量を、次の等式を用いて算出する。

10

【0157】

【数 26】

$$\text{nulls} = (b_r/8) * ((\text{StreamDTS} - \text{StreamPCR}) - (\text{DTS}_{\text{new}} - \text{PCR}_{\text{new}})) / 27000000.0.$$

【0158】

前記スプライシング法は、ストリームの継続性を維持するために前記新シーケンスに追加する必要のある再スタンプオフセットを、次の等式を用いて算出することができる。

【0159】

【数 27】

20

$$\text{PCR_restamp} = (\text{StreamPCR} + (\text{nulls} * 8 / b_r) * 27000000.0) - \text{PCR}_{\text{new}}$$

【0160】

前記 PCR_restamp の量を PCR に加え、(PCR_restamp / 300) の量をそのプログラムに関連するオーディオを含むエレメンタリストリームにある DTS および PTS に加えることにより、その移行点に続くストリーミングデータを再スタンプすることができる。

【0161】

図 9 は、このスプライシング法がどのように働くかを図示する。その新シーケンスでのバッファレベルが、その前シーケンスでのバッファレベルより高い場合は、別の方法を採用することができる。例えば、当該 P フレーム挿入法およびトリックモードから再生への移行で説明した同様のプロセスを用いて最後の画像を「フリーズ」することにより、前記新シーケンスをバッファすることができる。

30

【0162】

前記再バッファリング法は、ヌルパケットの短いシーケンスが続く多数の P フレームを挿入する工程を含む場合がある。その前シーケンスの終了直後に、StreamPCR で始まる前記新シーケンスが伝送される場合、StreamPCR + (DTS_{new} - PCR_{new}) によって与えられるインスタントで最初のフレームをデコードすることができる。前記新シーケンスの最初のフレームは、StreamDTS によって与えられるインスタントでデコードされるものと期待され、これは、その前シーケンスからの最後のデコード時間後の 1 フレームである。これを等式によって表すと、StreamPCR + (DTS_{new} - PCR_{new}) > StreamDTS または (DTS_{new} - PCR_{new}) > (StreamDTS - StreamPCR) となり、次にインターバルがあり、ここでそのデコーダはデコードを停止する (すなわちバッファアンダーフローである)。

40

【0163】

フィールドが予測した、静止 P フレームおよび B フレームまたはダミーフレームを「フレーム」として画像構造でエンコードすること、およびマクロブロックを「フィールド」という予測タイプでエンコードすることが可能であると理解される。これは、P フレームのマクロブロックを、動きベクトル = (0, 0) である「トップフィールド」をフォワードレファレンスする「トップフィールド」および動きベクトル = (0, 0) である「トッ

50

プフィールド」をフォワードレファレンスする「ボトムフィールド」でエンコードすることによって達成可能である。Bフレームのマクロブロックは、動きベクトル＝(0, 0)である「トップフィールド」をバックワードレファレンスする「トップフィールド」および動きベクトル＝(0, 0)である「トップフィールド」をバックワードレファレンスする「ボトムフィールド」でエンコードすることができる。

【0164】

図10は、データストリーム通信のためのシステムを図示する。図10が示すように、セットトップボックス1002はデータサーバ1003および帯域幅調整モジュール1004と通信中である。帯域幅調整モジュール1004は、指定タイムスロットが前記データストリームを処理するのに十分な大きさでない場合に、前記データストリームの一部（例えばトリックモードパケット）を別のタイムスロットに移動することができる。この場合のトリックモードパケットの例として、毎秒約10フレームの速度で通信可能な一連のIフレームが含まれる。また、帯域幅調整モジュール1004は、前記トリックモードパケットにダミーデータ（例えばBフレームおよびPフレーム）を挿入することができる。

10

【0165】

また、システム1000は、セットトップボックス1002と通信中の圧縮／解凍コーダ1005を含むことができる。圧縮／解凍コーダ1005は、MPEG標準に沿って動作可能である。また、システム1000は、セットトップボックス1002からの画像を表示するためのディスプレイ1006、およびトリックモード再生（例えば早送り、巻き戻し、再生、一時停止、および停止）を開始するための、セットトップボックスと通信可能なユーザインタフェース1007を含むことができる。ディスプレイ1006は、従来のテレビである場合がある。ユーザインタフェース1007は、ワイヤレス方法を用いてセットトップボックスと通信することができる。ユーザインタフェース1007は、赤外線（IR）、無線周波（RF）、または他の任意の適切なタイプのリンクのようなワイヤレスリンクを使って通信可能な従来のリモートコントロールである場合がある。本書で開示する方法、装置、およびシステムは、例えばパーソナル・デジタル・アシスタント（PDA）、ラップトップ、および携帯電話など、ビデオデータを表示可能なコンピュータ、またはポータブル若しくは携帯装置と共に使用することもできる。

20

【0166】

図11は、データストリーム通信のための方法のフロー図である。1101では、第1のデータストリームの第1のタイムスロットであり、1102では、第2のデータストリームの第2のタイムスロットである。1103で、前記第2のデータストリームが前記第2のタイムスロットより大きいかが決定される。前記第2のデータストリームが前記第2のタイムスロットより大きくない場合、1104において、前記第2のデータストリームは前記第2のタイムスロットで伝送される。一方、前記第2のデータストリームが前記第2のタイムスロットより大きい場合、1105において、前記第2のデータストリームの一部が前記第1のタイムスロットに移動される。1106で、前記第2のデータストリームは伝送される。

30

【0167】

さらに、上述の方法は、さらに、前記移動部分の関数としてデータストレージ量を制御することができ、前記第2のデータストリームのサイズおよび前記第2のタイムスロットを監視することができる。これらの方法は、MPEG標準規格に準拠して前記データストリームを圧縮および解凍し、前記第1のタイムスロットの未使用の帯域幅を前記第2のタイムスロットに再分配することができる。上述の方法は、前記データストリームを監視することにより、前記データストリームを通信するための最大速度を決定することができる。

40

【0168】

図12は、データストレージまたはバッファレベルの制御方法のフロー図である。1201において、Iフレームを有するデータストリームにデータフレーム（例えばBまたはPフレームのダミーフレーム）が追加される。1202において、前記データストリーム

50

の伝送速度が変更される。1203において、前記第1のモードから第2のモードへの切り替えを求めるコマンドが受信され、1204において、第1のモードから第2のモードへの移行が行われる。前記第1、第2のモードは、トリックモード再生モードおよび／または通常再生モードである。

【0169】

本発明の開示の真の範囲は、本明細書にて図を用いて開示した実施形態に限定されない。例えば、効率的なトリックモードプレイバックを行うための前述の様々な方法の開示を、別個に用いること、あるいは相互の組み合わせによって用いることができる。さらに、開示した前記実施形態は、様々な画像サイズ（例えばHDTV）およびフレームレートで動作するものであると理解される。これらの考えられる方法は、ビジュアルアーチファクト、ブラックスクリーン、アンダーフロー、バッファオーバーフローに典型的に伴うマクロブロッキング、あるいはバッファ管理なしに実行される移行に一般に生じる中断を起こすことなく、異なる再生速度と通常再生速度との間の滑らかな移行を可能にするものであると理解される。また、これら開示された方法の一部として、異なるダミーBフレームおよびPフレームのエンコーディングを使うことができる。例えば、一部の実施形態では、フレーム予測されたBフレームおよびPフレームを「静止」フレームとして使う代わりに、上述したように、異なるエンコーディングを使う場合があり、これは本発明の範囲に含まれる。これは、インターレース画像のような特定のタイプのフォーマットのために提供され得るものである。

【0170】

さらに、当業者であれば理解するように、本明細書で開示された発明の態様の多くは、メディアストリーミングまたはビデオ・オン・デマンドのために採用されるのでないソフトウェアまたはハードウェアソリューションとして、コンピュータシステムに適用可能である。同様に、これらの実施形態は、VOD（ビデオ・オン・デマンド）コンセプトを採用するシステムあるいは特定のタイプのコンピュータ、プロセッサ、スイッチ、ストレージデバイス、メモリ、アルゴリズム等を採用するシステムに限定されない。デジタルプロセッシング機能、ネットワーク機能、およびストレージ機能にかかるコストの急速な低下を鑑み、例えば、前記システムの本発明の動作を変えずに、ある特定の機能のためのプロセッシングおよびストレージを、本明細書で説明した1機能的要素から別の機能的要素に移行させることは容易に可能である。多くの場合、本明細書で説明した実装の場所（すなわち機能的要素）は、単に設計者の好みの問題であり、絶対的な要件ではない。従って、限定されることが明示されていない限り、本発明の保護範囲は上述の具体的な実施形態に限定されることを意図しない。

【図面の簡単な説明】

【0171】

【図1】図1は、ビデオ・バッファ・レベルの一例である。

【図2】図2は、対象画像群（Group of Pictures）の長さを増す方法の一例である。

【図3】図3は、ビデオフレームのシーケンスがバッファレベルに与える影響を図示するものである。

【図4】図4は、ビデオストリームの分布曲線である。

【図5】図5は、トリックモードパケット調整を図示するものである。

【図6】図6は、トリックモードパケットの変化がバッファレベルに与える影響を示す図である。

【図7】図7は、トリックモードストリームを作成するためにバッファの最適化が使用される方法を図示するものである。

【図8】図8は、トリックモード・ビデオ・ストリームの出力を提供するものである。

【図9】図9は、スプライシング法を図示するものである。

【図10】図10は、データストリームを通信するためのシステムである。

【図11】図11は、データストリームを通信するための方法のフロー図である。

【図 1 2】 図 1 2 は、データ・ストレージ・レベルの制御方法のフロー図である。

【図 1】

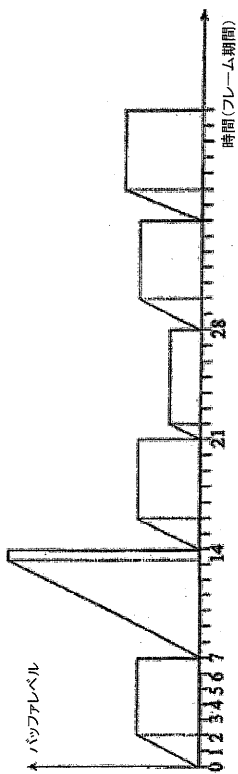


Figure 1

【図 2】

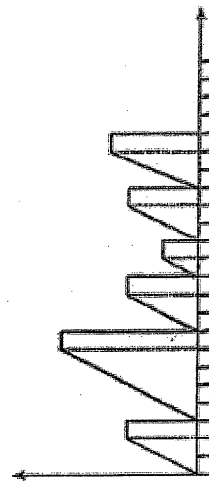


Figure 2

【図 3】

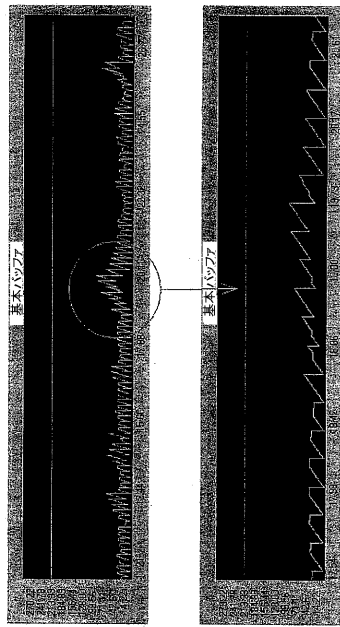


Figure 3

【図 4】

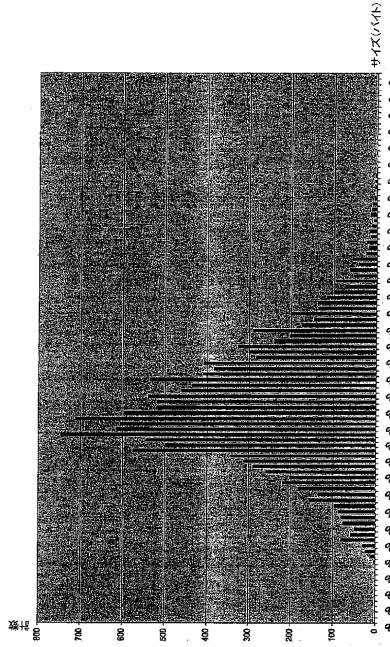


Figure 4

【図 5】

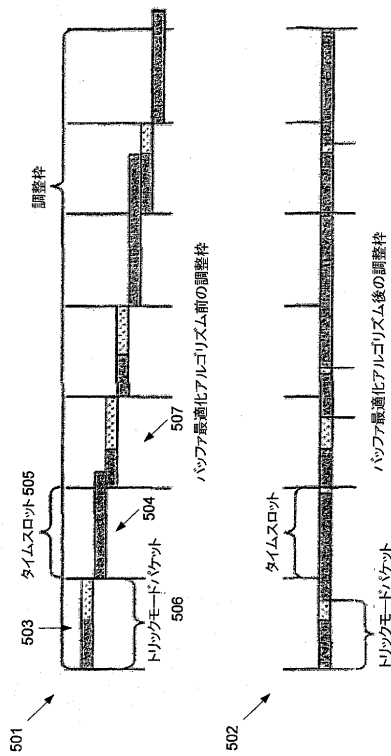


Figure 5

【図 6】

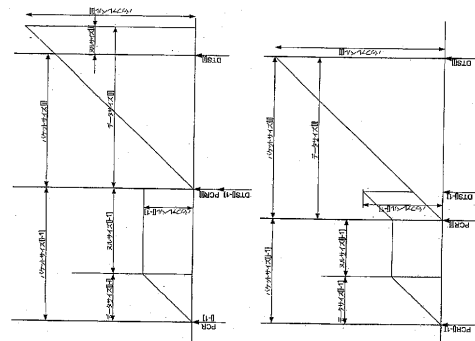


Figure 6

【図 7】

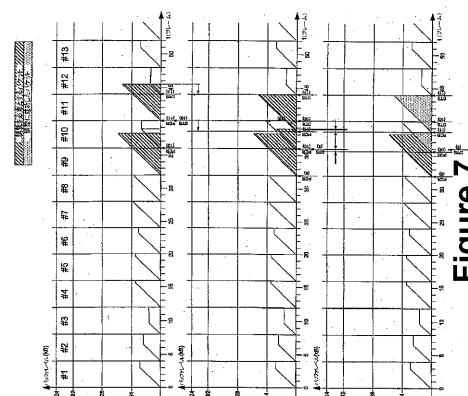


Figure 7

【図 8】

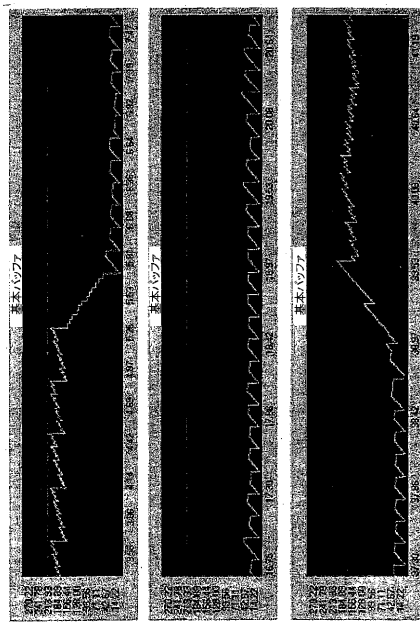


Figure 8

【図 9】

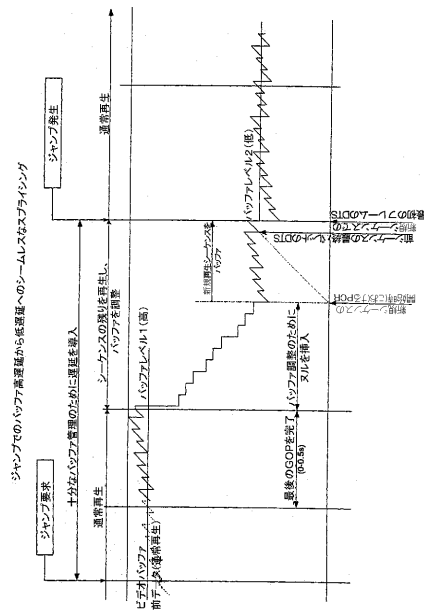


Figure 9

【図 10】

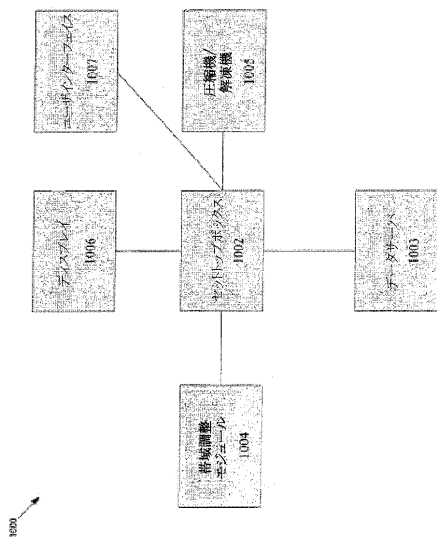


Figure 10

【図 11】

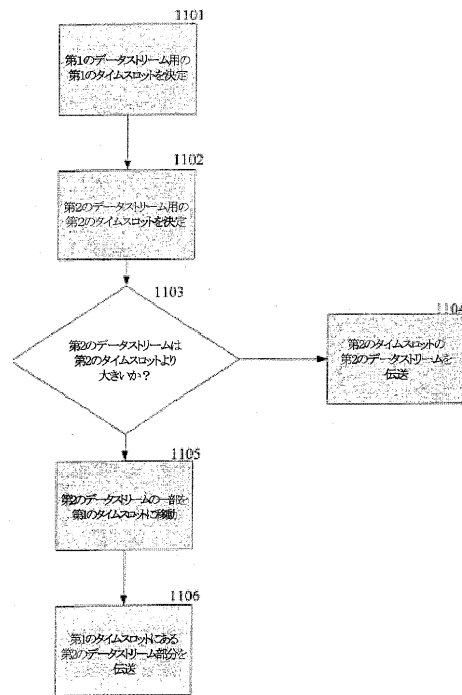


Figure 11

【図 12】

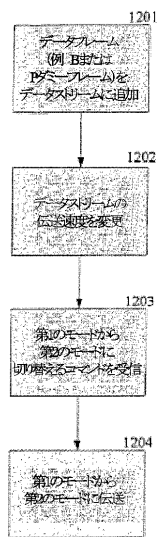


Figure 12

フロントページの続き

(72)発明者 ペイブス、ジャッキース

アメリカ合衆国、95630 カリフォルニア州、フォルサム、アパートメント ナンバー 28
23、アイロン ポイント ロード 1550

審査官 後藤 嘉宏

(56)参考文献 特開2000-175189 (JP, A)

国際公開第02/39744 (WO, A1)

(58)調査した分野(Int.Cl., DB名)

H04N 7/173