

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2010-41358

(P2010-41358A)

(43) 公開日 平成22年2月18日(2010.2.18)

(51) Int.Cl. F I テーマコード (参考)  
**H04N 7/26 (2006.01)** H04N 7/13 Z 5C059  
 5C159

審査請求 未請求 請求項の数 8 O L (全 18 頁)

|           |                              |          |                                |
|-----------|------------------------------|----------|--------------------------------|
| (21) 出願番号 | 特願2008-201468 (P2008-201468) | (71) 出願人 | 591128453                      |
| (22) 出願日  | 平成20年8月5日 (2008.8.5)         |          | 株式会社メガチップス                     |
|           |                              |          | 大阪府大阪市淀川区宮原4丁目1番6号             |
|           |                              | (74) 代理人 | 100125704                      |
|           |                              |          | 弁理士 坂根 剛                       |
|           |                              | (72) 発明者 | 成松 伸将                          |
|           |                              |          | 大阪市淀川区宮原4丁目1番6号 株式会            |
|           |                              |          | 社メガチップス内                       |
|           |                              | (72) 発明者 | 長谷川 弘                          |
|           |                              |          | 大阪市淀川区宮原4丁目1番6号 株式会            |
|           |                              |          | 社メガチップス内                       |
|           |                              | Fターム(参考) | 5C059 KK41 MA00 TA46 TB01 TC37 |
|           |                              |          | TC38 TD14 TD16 UA02 UA05       |
|           |                              |          | UA32                           |

最終頁に続く

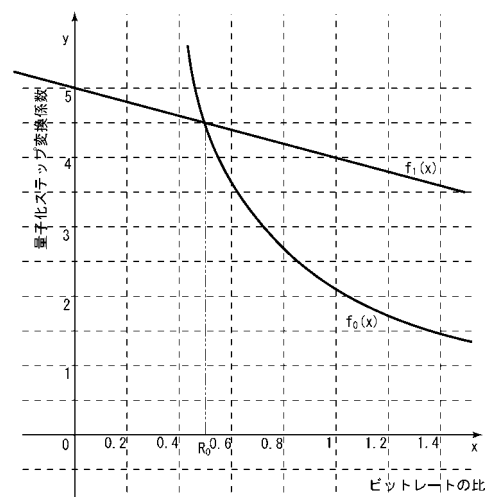
(54) 【発明の名称】 トランスコーダ

## (57) 【要約】

【課題】そこで、本発明は前記問題点に鑑み、量子化ステップ値の急激な変動を抑え、安定したレート制御を可能とする技術を提供することを目的とする。

【解決手段】ビットレート比から量子化ステップ変換係数を算出する関数  $f_1$  は、関数  $f_0$  と基準点  $(R_0, f_0(R_0))$  で交わる傾き - 1 の直線である。関数  $f_0$  は、単調減少の指数関数である。基準ビットレート比  $R_0$  は、第1ストリームの全体ビットレート  $S$ 、第2のストリームの全体目標ビットレート  $T$  を用いて  $R_0 = T/S$  で表される。関数  $f_0$  は、符号変換においてビットレート比と量子化ステップ変換係数の関係を適切に表現するが、ビットレート比が0.5程度の領域で変動率が大い。関数  $f_1$  は変動率が小さく、量子化ステップ変換値の急激な変動を抑制することができる。

【選択図】 図3



**【特許請求の範囲】****【請求項 1】**

第 1 のストリームを第 2 のストリームに変換するトランスコードであって、

第 1 のストリームと第 2 のストリームのビットレート比から第 1 のストリームと第 2 のストリームの量子化ステップ変換値を算出する変換値算出部と、

第 1 のストリームの量子化ステップ値に前記量子化ステップ変換値を乗算することで第 2 のストリームの量子化ステップ値を算出する量子化ステップ値算出部と、  
を備え、

前記変換値算出部は、前記ビットレート比と前記量子化ステップ変換値との関係が属する領域を指定する領域指定関数と、前記領域内において前記ビットレート比と前記量子化ステップ変換値との具体的な関係を決定する関係決定関数とを用いて、前記ビットレート比から前記量子化ステップ変換値を算出することを特徴とするトランスコード。

10

**【請求項 2】**

請求項 1 に記載のトランスコードにおいて、

第 1 のストリームの全体ビットレートと第 2 のストリームの全体目標ビットレートとの比を基準ビットレート比とすると、前記領域指定関数と前記関係決定関数とは、前記基準ビットレート比に対して同じ前記量子化ステップ変換値が対応付けられていることを特徴とするトランスコード。

**【請求項 3】**

請求項 2 に記載のトランスコードにおいて、

20

変換済みの過去の期間の第 1 のストリームのビットレートと第 2 のストリームの目標ビットレートとの比を現在ビットレート比とすると、前記領域指定関数を用いて前記現在ビットレート比に対して算出した前記量子化ステップ変換値と、前記関係決定関数を用いて前記現在ビットレート比に対して算出した前記量子化ステップ変換値との差が所定の閾値を越えた場合、前記現在ビットレート比により前記基準ビットレート比を更新することを特徴とするトランスコード。

**【請求項 4】**

請求項 2 または請求項 3 に記載のトランスコードにおいて、

前記基準ビットレート比の近傍においては、前記関係決定関数の変化率は、前記領域指定関数の変化率よりも小さいことを特徴とするトランスコード。

30

**【請求項 5】**

請求項 4 に記載のトランスコードにおいて、

前記領域指定関数は、  
単調減少の指数関数、

を含むことを特徴とするトランスコード。

**【請求項 6】**

請求項 4 または請求項 5 に記載のトランスコードにおいて、

前記関係決定関数は、

前記基準ビットレート比において前記領域指定関数と交わる負の傾きを有する直線であることを特徴とするトランスコード。

40

**【請求項 7】**

請求項 4 または請求項 5 に記載のトランスコードにおいて、

前記関係決定関数は、

前記基準ビットレート比において前記領域指定関数と交わり、前記基準ビットレート比において負の傾きを有する接線を持つ双曲線であることを特徴とするトランスコード。

**【請求項 8】**

請求項 4 または請求項 5 に記載のトランスコードにおいて、

前記関係決定関数は、

前記基準ビットレート比において前記領域指定関数と接する直線であることを特徴とするトランスコード。

50

## 【発明の詳細な説明】

## 【技術分野】

## 【0001】

本発明は、入力ストリームをデコードして異なる出力ストリームに変換するトランスコーダに関し、詳しくは、出力ストリームの発生符号量を目標ビットレートに向けて適正に制御する技術に関する。

## 【背景技術】

## 【0002】

デジタル放送で配信される画像や、DVD、ハードディスクなどに格納される画像などは、各種の符号化方式に従って圧縮される。これは、伝送帯域を圧迫しないため、伝送速度を上げるため、あるいは、記憶サイズを小さくするなどの目的のためである。

10

## 【0003】

画像の符号化方式には、MPEG2やH.264など様々な規格が存在する。そして、入力した符号化画像の符号量を削減するなどの目的で符号化方式の変換が行われる場合がある。トランスコーダは、入力した符号化画像を一旦デコードする。そして、トランスコーダは、デコードした画像を、再び異なる符号化方式（あるいは同じ符号化方式）で符号化する。このようにして、トランスコーダは、出力ストリームのビットレートを制御するのである。

## 【0004】

下記特許文献1では、トランスコーダにおいて出力ストリームの量子化ステップ値を算出するための技術が開示されている。特許文献1では、デコードされた画像のマクロブロック内の画素のばらつき度合いや、動き評価値を加味して量子化ステップ値を調整するようにしている。

20

## 【0005】

【特許文献1】特開2008-42426号公報

## 【発明の開示】

## 【発明が解決しようとする課題】

## 【0006】

ビットレートを一定に保つため、あるいはビットレートを削減することを目的として出力ストリームの量子化ステップ値を制御する場合、入力ストリームの状況、演算条件によっては量子化ステップの変動が大きくなるケースが発生する。量子化ステップの変動が大きくなると、レート制御が不安定になり、画質の劣化を招くことになる。

30

## 【0007】

そこで、本発明は前記問題点に鑑み、量子化ステップ値の急激な変動を抑え、安定したレート制御を可能とする技術を提供することを目的とする。

## 【課題を解決するための手段】

## 【0008】

上記課題を解決するため、請求項1記載の発明は、第1のストリームを第2のストリームに変換するトランスコーダであって、第1のストリームと第2のストリームのビットレート比から第1のストリームと第2のストリームの量子化ステップ変換値を算出する変換値算出部と、第1のストリームの量子化ステップ値に前記量子化ステップ変換値を乗算することで第2のストリームの量子化ステップ値を算出する量子化ステップ値算出部と、を備え、前記変換値算出部は、前記ビットレート比と前記量子化ステップ変換値との関係が属する領域を指定する領域指定関数と、前記領域内において前記ビットレート比と前記量子化ステップ変換値との具体的な関係を決定する関係決定関数とを用いて、前記ビットレート比から前記量子化ステップ変換値を算出することを特徴とする。

40

## 【0009】

請求項2記載の発明は、請求項1に記載のトランスコーダにおいて、第1のストリームの全体ビットレートと第2のストリームの全体目標ビットレートとの比を基準ビットレート比とすると、前記領域指定関数と前記関係決定関数とは、前記基準ビットレート比に対

50

して同じ前記量子化ステップ変換値が対応付けられていることを特徴とする。

【0010】

請求項3記載の発明は、請求項2に記載のトランスコードにおいて、変換済みの過去の期間の第1のストリームのビットレートと第2のストリームの目標ビットレートとの比を現在ビットレート比とすると、前記領域指定関数を用いて前記現在ビットレート比に対して算出した前記量子化ステップ変換値と、前記関係決定関数を用いて前記現在ビットレート比に対して算出した前記量子化ステップ変換値との差が所定の閾値を越えた場合、前記現在ビットレート比により前記基準ビットレート比を更新することを特徴とする。

【0011】

請求項4記載の発明は、請求項2または請求項3に記載のトランスコードにおいて、前記基準ビットレート比の近傍においては、前記関係決定関数の変化率は、前記領域指定関数の変化率よりも小さいことを特徴とする。

【0012】

請求項5記載の発明は、請求項4に記載のトランスコードにおいて、前記領域指定関数は、単調減少の指数関数、を含むことを特徴とする。

【0013】

請求項6記載の発明は、請求項4または請求項5に記載のトランスコードにおいて、前記関係決定関数は、前記基準ビットレート比において前記領域指定関数と交わる負の傾きを有する直線であることを特徴とする。

【0014】

請求項7記載の発明は、請求項4または請求項5に記載のトランスコードにおいて、前記関係決定関数は、前記基準ビットレート比において前記領域指定関数と交わり、前記基準ビットレート比において負の傾きを有する接線を持つ双曲線であることを特徴とする。

【0015】

請求項8記載の発明は、請求項4または請求項5に記載のトランスコードにおいて、前記関係決定関数は、前記基準ビットレート比において前記領域指定関数と接する直線であることを特徴とする。

【発明の効果】

【0016】

本発明のトランスコードは、ビットレート比と量子化ステップ変換値との関係が属する領域を指定する領域指定関数と、領域内においてビットレート比と量子化ステップ変換値との具体的な関係を決定する関係決定関数とを用いて、ビットレート比から量子化ステップ変換値を算出する。領域指定関数により、符号変換に最適な特性を維持しつつ、関係決定関数により、量子化ステップ変換値の変動を調整することができる。

【0017】

また、領域指定関数および関係決定関数を用いて現在ビットレート比に対して算出した量子化ステップ変換値の差が所定の閾値を越えた場合、現在ビットレート比により基準ビットレート比を更新する。これにより、関係決定関数の特性が、領域指定関数の特性から大きく乖離することを回避できる。

【0018】

そして、基準ビットレート比の近傍においては、関係決定関数の変化率は、領域指定関数の変化率よりも小さいので、領域指定関数の特性により符号変換に適した関係を維持しつつ、関係決定関数により量子化ステップ変換値の急激な変動を抑制することができる。

【発明を実施するための最良の形態】

【0019】

{第1の実施の形態}

<1. トランスコードの構成>

以下、図面を参照しつつ本発明の実施の形態について説明する。図1は、本実施の形態に係るトランスコード1のブロック図である。このトランスコード1は、CPU11、デコード12、画像メモリ13、ステップ値算出部14、エンコーダ15、バッファメモリ

10

20

30

40

50

16を備えている。

【0020】

デコーダ12は、第1ストリームを入力する。第1ストリームは、符号化画像のストリームである。デコーダ12は、第1ストリームをデコードし、非圧縮の画像データを画像メモリ13に出力する。エンコーダ15は、画像メモリ13に格納されたデコード後の非圧縮の画像データを再び符号化し、第2ストリームを出力する。

【0021】

たとえば、トランスコーダ1は、MPEG2で符号化された第1ストリームを入力し、H.264で符号化された第2ストリームを出力するなど、ストリームの符号化方式を変換する処理を行う。あるいは、トランスコーダ1は、MPEG2で符号化された第1ストリームを入力し、再びMPEG2で符号化された第2ストリームを出力するなど、同じ符号化方式のストリームを出力する。

【0022】

なお、デコーダ12、ステップ値算出部14、エンコーダ15における各演算処理は、ハードウェアにより実現されてもよいし、ソフトウェア処理により実現されていてもよい。つまり、これら演算処理部は、ハードウェア回路として構成されていてもよいし、CPUとメモリに格納されたプログラムとで構成されていてもよい。あるいは、一部の処理がハードウェアで処理され、一部の処理がソフトウェアで処理される形態であってもよい。

【0023】

< 2. 制御単位期間と期間内の情報 >

図2は、トランスコーダ1が入出力するストリームの制御単位期間ごとの情報を表した図である。トランスコーダ1は、処理時間軸を制御単位期間 $L_n$  ( $n = 1, 2, \dots$ ) に区分し、この制御単位期間 $L_n$ を処理単位としてレート制御を行う。以下の説明において、適宜、制御単位期間 $L_n$ を第 $n$ 期間と呼ぶことにする。制御単位期間 $L_n$ としては、1フレーム、連続する複数フレーム、1GOP、連続する複数GOPなどを1単位期間として設定することが可能である。

【0024】

第1ストリームの入力全体ビットレート $S$ は、シーケンスヘッダなどから取得される。入力平均ビットレート $S_n$ は、第 $n$ 期間における第1ストリームの平均ビットレートである。トランスコーダ1は、 $M$ 期間分の入力平均ビットレート $S_n$ に関する情報をバッファメモリ16に格納可能としている。つまり、バッファメモリ16は、第 $(n - M + 1)$ 期間から第 $n$ 期間までの入力平均ビットレート $S_n$ に関する情報を格納可能としている。期間平均ビットレート $AS_n$ は、第 $(n - M + 1)$ 期間から第 $n$ 期間における入力平均ビットレート $S_n$ を平均した値である。期間平均ビットレート $AS_n$ は、数(1)式で表される。

【0025】

【数1】

$$AS_n = \frac{1}{M} \sum_{i=n-M+1}^n S_i \quad \dots(1)$$

【0026】

第2ストリームの全体目標ビットレート $T$ は、ユーザにより設定される。たとえば、ユーザが、トランスコーダ1に設けられた図示せぬ操作部を用いて全体目標ビットレート $T$ を設定する。

【0027】

目標設定ビットレート $T_n$ は、第 $n$ 期間における第2ストリームの目標ビットレートである。出力平均ビットレート $C_n$ は、第 $n$ 期間における変換済みの第2ストリームの平均

10

20

30

40

50

ビットレートである。トランスコーダ 1 は、M 期間分の出力平均ビットレート  $C_n$  に関する情報をバッファメモリ 16 に格納可能としている。つまり、バッファメモリ 16 は、第  $(n - M + 1)$  期間から第  $n$  期間までの出力平均ビットレート  $C_n$  に関する情報を格納可能としている。期間平均ビットレート  $AC_n$  は、第  $(n - M + 1)$  から第  $n$  期間における出力平均ビットレート  $C_n$  を平均した値である。期間平均ビットレート  $AC_n$  は、数 (2) 式で表される。

【0028】

【数 2】

$$AC_n = \frac{1}{M} \sum_{i=n-M+1}^n C_i \quad \dots(2)$$

10

【0029】

なお、この実施の形態では、期間平均ビットレート  $AS_n$ 、 $AC_n$  を計算する場合に利用するバッファ期間を、制御単位期間  $L_n$  と連動させたが、バッファ期間の設定方法は、これに限るものではない。たとえば、符号化が終わった時点の過去 1 フレーム、連続する過去複数フレーム、過去 1 GOP、連続する過去複数 GOP などをバッファ期間として設定することが可能である。

20

【0030】

量子化ステップ変換係数  $\alpha_n$  は、第  $(n - 1)$  期間の終了時に算出される係数である。第 1 ストリームの量子化ステップ値そのもの、あるいは第 1 ストリームの量子化ステップ値から計算される値  $P$  に、量子化ステップ変換係数  $\alpha_n$  を乗算することで、第 2 ストリームの量子化ステップ値  $Q$  が決定される。この関係を数 (3) 式に示す。

【0031】

【数 3】

$$Q = \alpha_n P \quad \dots(3)$$

30

【0032】

再び、図 1 を参照する。CPU 11 は、デコーダ 12 から第 1 ストリームのピクチャごとのビットレート  $s\_b$  を取得し、バッファメモリ 16 に格納する。CPU 11 は、バッファメモリ 16 に格納されたビットレート  $s\_b$  に基づき、第  $n$  期間の入力平均ビットレート  $S_n$  や期間平均ビットレート  $AS_n$  を算出する。

【0033】

デコーダ 12 は、第 1 ストリームのマクロブロックごとの量子化ステップ値  $P$  をステップ値算出部 14 に出力する。

【0034】

40

CPU 11 は、エンコーダ 15 から変換済みの第 2 ストリームのピクチャごとのビットレート  $c\_b$  を取得し、バッファメモリ 16 に格納する。CPU 11 は、バッファメモリ 16 に格納されたビットレート  $c\_b$  に基づき、第  $n$  期間の出力平均ビットレート  $C_n$  や期間平均ビットレート  $AC_n$  を算出する。

【0035】

CPU 11 は、また、ユーザにより設定された全体目標ビットレート  $T$  を取得する。

【0036】

CPU 11 は、全体目標ビットレート  $T$ 、出力平均ビットレート  $C_n$ 、期間平均ビットレート  $AC_n$ 、入力平均ビットレート  $S_n$ 、期間平均ビットレート  $AS_n$  などの情報を利用し、第  $(n + 1)$  期間の第 2 ストリームの目標設定ビットレート  $T_{n+1}$  を決定する。

50

目標設定ビットレート  $T_{n+1}$  の算出方法は、次の項<3．目標ビットレートの算出例>において説明する。

【0037】

CPU11は、さらに、第 $(n+1)$ 期間の第2ストリームの目標設定ビットレート  $T_{n+1}$  から本発明の特徴的な処理である関数  $f$  を利用することで、量子化ステップ変換係数  $\alpha_{n+1}$  を算出し、ステップ値算出部14に出力する。関数  $f$  については、後の<4．変換係数の算出>において説明する。

【0038】

ステップ値算出部14は、量子化ステップ値  $P$  と量子化ステップ変換係数  $\alpha_{n+1}$  を入力し、数(3)式に示す演算を実行することで、第2ストリームの量子化ステップ値  $Q$  を算出する。ステップ値算出部14は、算出した量子化ステップ値  $Q$  をエンコーダ15に与える。

10

【0039】

エンコーダ15は、ステップ値算出部14から入力した量子化ステップ値  $Q$  に基づいて非圧縮の画像データを符号化し、第2ストリームを生成する。

【0040】

<3．目標ビットレートの算出例>

次に、第 $(n+1)$ 期間における第2ストリームの目標設定ビットレート  $T_{n+1}$  の算出例について説明する。トランスコーダ1は、第 $n$ 期間経過時において、第 $(n+1)$ 期間における第2ストリームの目標設定ビットレート  $T_{n+1}$  を決定する。

20

【0041】

数(4)式は、第 $(n+1)$ 期間の目標設定ビットレート  $T_{n+1}$  を算出する式である。具体的には、第 $n$ 期間の変換済みの第2ストリームの出力平均ビットレート  $C_n$  を第 $n$ 期間の目標設定ビットレート  $T_n$  で除算した値を目標比と考えると、第 $(n+1)$ 期間の目標設定ビットレート  $T_{n+1}$  は、第2ストリームの全体目標ビットレート  $T$  を目標比で除算することで算出される。

【0042】

【数4】

$$T_{n+1} = k * \frac{T}{\frac{C_n}{T_n}} \quad \dots(4)$$

30

【0043】

数(4)式中、 $k$  は正の係数であり、目標設定ビットレート  $T_{n+1}$  を調整するための係数である。このように、 $C_n / T_n$  により第 $n$ 期間における目標との比を算出し、全体目標ビットレート  $T$  を目標比で除算することで第 $(n+1)$ 期間の目標設定ビットレート  $T_{n+1}$  を調整し、全体として目標ビットレートに近づくよう制御するのである。

【0044】

数(4)式では、第 $n$ 期間の第2ストリームの出力平均ビットレート  $C_n$  を目標設定ビットレート  $T_n$  で除算した値を目標比として利用しているが、この目標比をさらに調整するようにしたのが、数(5)式である。

40

【0045】

【数 5】

$$T_{n+1} = k * \frac{T}{\frac{S_{n-1} * C_n}{S_n T_n}} \quad \dots(5)$$

【0046】

数(5)式において、目標比に  $S_{n-1} / S_n$  が乗算されている。 $S_{n-1} / S_n$  は、第(n-1)期間の第1ストリームの入力平均ビットレート  $S_{n-1}$  を、第n期間の第1ストリームの入力平均ビットレート  $S_n$  で除算した値であり、入力平均ビットレートの期間比と呼ぶことができる。つまり、目標比に期間比を乗算することで、目標比を調整しているのである。

10

【0047】

このように目標比に期間比を乗算することで、局所的な目標比の変化を補正することができる。たとえば、入力平均ビットレート  $S_n$  が局所的に小さく変化した場合には、それに伴い、目標比  $C_n / T_n$  も小さく変動する場合がある。このような場合にも、期間比  $S_{n-1} / S_n$  (この場合、期間比が1より大きな値となる。) を乗算することで、目標比を調整し、目標設定ビットレート  $T_{n+1}$  が大きく変動することを回避できるのである。逆に、入力平均ビットレート  $S_n$  が局所的に大きく変化した場合には、期間比  $S_{n-1} / S_n$  が1より小さい値となり、目標比が急激に大きくなることを抑制することができる。

20

【0048】

数(5)式では、第(n-1)期間の第1ストリームの入力平均ビットレート  $S_{n-1}$  を、第n期間の第1ストリームの入力平均ビットレート  $S_n$  で除算した値を期間比とした。これに対して、第1ストリームの第(n+1)期間の入力平均ビットレート  $S_{n+1}$  を先読みできる場合には、入力平均ビットレート  $S_{n+1}$  を利用してもよい。入力平均ビットレート  $S_{n+1}$  を先読みできる場合とは、入力平均ビットレート  $S_{n+1}$  に関する情報をバッファリングした上で、第(n+1)期間の目標設定ビットレート  $T_{n+1}$  を算出できる時間的余裕がある場合である。つまり、ある程度の処理遅延が許されるような場合である。数(6)式に、入力平均ビットレート  $S_{n+1}$  を用いて目標設定ビットレート  $T_{n+1}$  を算出する式を示す。

30

【0049】

【数 6】

$$T_{n+1} = k * \frac{T}{\frac{S_n * C_n}{S_{n+1} T_n}} \quad \dots(6)$$

【0050】

数(6)式に示すように、期間比として、 $S_n / S_{n+1}$ 、つまり、第n期間の第1ストリームの入力平均ビットレート  $S_n$  を、第(n+1)期間の第1ストリームの入力平均ビットレート  $S_{n+1}$  で除算した値を用いている。このように、第(n+1)期間の第1ストリームの入力平均ビットレート  $S_{n+1}$  を利用することで、目標設定ビットレート  $T_{n+1}$  をより高い精度で制御することが可能となる。つまり、より最新の入力情報を用いることで、目標ビットレートに対する最適制御を行うことができるのである。

40

【0051】

また、数(5)式では、期間比として  $S_{n-1} / S_n$  を利用した。これに代えて、第(n-1)期間を含む過去M期間の期間平均ビットレート  $A S_{n-1}$  と、第n期間を含む過去M期間の期間平均ビットレート  $A S_n$  の比を期間比として用いてもよい。つまり、数(

50

5) 式において、期間比として、 $S_{n-1} / S_n$  に代えて、 $AS_{n-1} / AS_n$  を用いてもよい。これにより、局所的な変動の影響を小さくし、目標設定ビットレート  $T_{n+1}$  を最適制御することが可能である。

【0052】

あるいは、第1ストリームの第  $(n+1)$  期間の入力平均ビットレート  $S_{n+1}$  を先読みできる場合には、第  $n$  期間を含む過去  $M$  期間の期間平均ビットレート  $AS_n$  と、第  $(n+1)$  期間を含む過去  $M$  期間の期間平均ビットレート  $AS_{n+1}$  の比を期間比として用いてもよい。つまり、数(5)式において、期間比として、 $S_{n-1} / S_n$  に代えて、 $AS_n / AS_{n+1}$  を用いてもよい。これにより、局所的な変動の影響を小さくするとともに、最新の入力ビットレートを利用することで、より精度高く目標設定ビットレート  $T_{n+1}$  を制御することが可能である。

10

【0053】

また、第  $n$  期間の出力平均ビットレート  $C_n$  に代えて、第  $(n-M+1)$  期間から第  $n$  期間までの期間平均ビットレート  $AC_n$  を用いても良い。つまり、数(4)式～数(6)式等、上述した算出例において、 $C_n$  に代えて  $AC_n$  を用いるのである。これにより、目標設定ビットレート  $T_{n+1}$  を、より緩やかに制御することが可能である。

【0054】

第  $(n+1)$  期間の目標設定ビットレート  $T_{n+1}$  を算出する別の算出例を数(7)式に示す。第  $n$  期間の目標設定ビットレート  $T_n$  から第  $n$  期間の変換済みの第2ストリームの出力平均ビットレート  $C_n$  を減算した値を目標差と考えると、第  $(n+1)$  期間の目標設定ビットレート  $T_{n+1}$  は、第2ストリームの全体目標ビットレート  $T$  に目標差を加算することで算出されるのである。

20

【0055】

【数7】

$$T_{n+1} = T + k * (T_n - C_n) \quad \dots(7)$$

【0056】

数(7)式中、 $k$  は正の係数であり、目標設定ビットレート  $T_{n+1}$  を調整するための係数である。このように、 $(T_n - C_n)$  により、第  $n$  期間における目標との差を算出し、全体目標ビットレート  $T$  に目標差を加算することで、出力ストリームが目標ビットレートに近づくような制御を行うのである。

30

【0057】

また、目標差  $(T_n - C_n)$  に、期間比  $S_{n-1} / S_n$  や、期間比  $AS_{n-1} / AS_n$  を乗算するようにしてもよい。あるいは、入力平均ビットレート  $S_{n+1}$  を先読みできる場合には、目標差  $(T_n - C_n)$  に、期間比  $S_n / S_{n+1}$  や、期間比  $AS_n / AS_{n+1}$  を乗算するようにしてもよい。また、 $C_n$  に代えて  $AC_n$  を用いても良い。

【0058】

< 4. 変換係数の算出 >

40

第  $(n+1)$  期間の目標設定ビットレート  $T_{n+1}$  が算出されると、数(8)式に示す演算を行うことにより、量子化ステップ変換係数  $\alpha_{n+1}$  を算出する。つまり、現在ビットレート比  $= T_{n+1} / S_n (= R_B)$  を関数  $f$  に代入することで、第  $(n+1)$  期間の量子化ステップ変換係数  $\alpha_{n+1}$  を算出する。

【0059】

【数 8】

$$\alpha_{n+1} = f(T_{n+1}/S_n) \quad \dots(8)$$

【0060】

数(8)式中、関数  $f$  は、“ビットレートの比”から“量子化ステップ値の比”(量子化ステップ変換係数)を求める関数である。“ビットレートの比”とは、第1ストリームのビットレートに対する第2ストリームのビットレートの比である。第1ストリームのビットレートとしては、 $S$ 、 $S_n$  などを用いる。第2ストリームのビットレートとしては  $T$ 、 $T_n$  などを用いる。“量子化ステップ値の比”とは、第1ストリームの量子化ステップ値に対する第2ストリームの量子化ステップ値の比である。“ビットレートの比”を  $R_B$ 、“量子化ステップ値の比”を  $R_Q$  とすると、関数  $f$  は、一般的に、数(9)式で表される。

10

【0061】

【数 9】

$$R_Q = f(R_B) \quad \dots(9)$$

20

【0062】

量子化ステップ変換係数  $\alpha_n$  の初期値  $\alpha_1$  は、数(10)式で与えられる。つまり、第2ストリームの全体目標ビットレート  $T$  を第1ストリームの入力全体ビットレート  $S$  で除算した値、つまりビットレートの比を関数  $f$  に入力することによって、量子化ステップ変換係数の初期値  $\alpha_1$  を求めている。

【0063】

【数 10】

$$\alpha_1 = f(T/S) \quad \dots(10)$$

30

【0064】

第2ストリームの全体目標ビットレート  $T$  を第1ストリームの入力全体ビットレート  $S$  で除算した値を、数(11)式に示すように、基準ビットレート比  $R_0$  と呼ぶことにする。

【0065】

【数 11】

$$R_0 = T/S \quad \dots(11)$$

40

【0066】

関数  $f$  について詳しく説明する。上述したように、関数  $f$  は、“ビットレートの比”から量子化ステップ変換係数  $\alpha_n$  を算出する関数である。数(12)式は、第1の実施の形態に係る関数  $f$  として関数  $f_1$  を示す。

【0067】

【数 1 2】

$$f_1(x) = -x + R_0 + \varepsilon R_0^{-\delta} \quad \dots (12)$$

【0 0 6 8】

関数  $f_1$  は、点  $(R_0, f_0(R_0))$  を通る傾き - 1 の直線である。ここで、関数  $f_0$  は、数 (1 3) 式で表される。関数  $f_0$  をそのまま関数  $f$  として利用することも可能であるが、本実施の形態においては、関数  $f_0$  を領域指定関数として利用し、この領域指定関数  $f_0$  の局所的な傾きを調整した関係決定関数  $f_1$  を利用して量子化ステップ変換係数を算出することとしている。 10

【0 0 6 9】

【数 1 3】

$$f_0(x) = \varepsilon x^{-\delta} = 2.1 * x^{-1.1} \quad \dots (13)$$

【0 0 7 0】

数 (1 3) 式においては、MPEG2 から H. 264 ヘストリーム変換する場合の  $\varepsilon$ 、 $\delta$  の値として、実験等に基づいて算出された具体的数値を例示した。つまり、実験の結果、 $\varepsilon = 2.1$ 、 $\delta = 1.1$  とすることで、良好な特性が得られた。本実施の形態においては、全ピクチャに共通の関数  $f_0$  を利用しているが、ピクチャタイプ (I ピクチャ、P ピクチャ、B ピクチャ) でそれぞれに異なる特性の関数を利用してもよい。 20

【0 0 7 1】

図 3 は、関数  $f_1$  と関数  $f_0$  を示す図である。図中、 $x$  軸はビットレートの比を示し、 $y$  軸は量子化ステップ変換係数を示している。関数  $f_1$ 、関数  $f_0$  として具体的に  $\varepsilon = 2.1$ 、 $\delta = 1.1$  の値を用いている。関数  $f_0$  は、指数関数であり、ビットレートの比が小さい領域については、その傾きが大きくなっている。ビットレートの比が、おおよそ 0.4 ~ 0.6 程度で推移することを想定すると、関数  $f_0$  はこの領域において大きく量子化ステップ変換係数が変動することになる。したがって、関数  $f_0$  をそのまま関数  $f$  として用いた場合には、ビットレートの比が小さい領域において、量子化ステップ変換係数の変動が大きくなる。そこで、基準ビットレート比  $R_0$  の近傍領域において関数  $f_0$  よりも変動の小さい関数  $f_1$  を用いて量子化ステップ変換係数を算出することとしている。なお、図 3 においては、基準ビットレート比  $R_0 = 0.5$  の場合を例に図示している。 30

【0 0 7 2】

つまり、領域指定関数  $f_0$  は、ビットレートの比と量子化ステップ変換係数との関係を決めるおおよそのエリアを規定する役割を有し、そのエリアの中で、関数  $f_1$  は、ビットレートの比と量子化ステップ変換係数との関係を具体的に決定する役割を有する。上述したように、関数  $f_0$  は、実験等に基づいて求められた関数であり、MPEG2 から H. 264 ヘストリーム変換する場合におけるビットレートの比と量子化ステップ変換係数の関係を良好に保つことができる。しかし、上述したように、ビットレートの比が 0.4 ~ 0.6 程度において、量子化ステップ変換係数の変動が大きくならないように、関数  $f_1$  を利用するのである。 40

【0 0 7 3】

関数  $f_1$  は、点  $(R_0, f_0(R_0))$  において、関数  $f_0$  と交わっている。つまり、基準ビットレート比  $R_0$  の近傍においては、関数  $f_0$  と比較的近い特性を維持している。しかし、関数  $f_1$  が、点  $(R_0, f_0(R_0))$  から離れるにつれて、関数  $f_1$  は関数  $f_0$  と特性が乖離する。関数  $f_1$  が関数  $f_0$  から特性が大きくなると、関数  $f_0$  の特性を維持できなくなり、関数  $f_1$  を使用し続けるのは不適切となる。そこで、数 (1 4) 式の関係を満たす限りにおいて、関数  $f_1$  を利用するのである。 50

【 0 0 7 4 】

【 数 1 4 】

$$|f_0(R_B) - f_1(R_B)| \leq 0.5 \quad \dots(14)$$

【 0 0 7 5 】

数 ( 1 4 ) 式において、 $R_B$  は、現在ビットレート比であり、数 ( 1 5 ) 式で示される。

【 0 0 7 6 】

10

そして、数 ( 1 4 ) 式の関係を満たさなくなった場合には、数 ( 1 5 ) 式に示す現在ビットレート比  $R_B$  により、基準ビットレート比  $R_0$  を更新するのである。

【 0 0 7 7 】

【 数 1 5 】

$$R_B = T_{n+1} / S_n \quad \dots(15)$$

【 0 0 7 8 】

20

これにより、図 4 に示すように、基準点 ( $R_0$ ,  $f_0(R_0)$ ) が更新され、あわせて関数  $f_1$  も更新される。つまり、関数  $f_0$  で指定されるエリアに沿って、関数  $f_1$  を移動させながら、関数  $f_1$  を用いて量子化ステップ変換係数を算出するのである。これにより、関数  $f_0$  と関数  $f_1$  の両方の特性を利用して、最適なレート制御を行うことが可能である。

【 0 0 7 9 】

上記の実施の形態においては、数 ( 8 ) 式に示したように、ビットレートの比として  $T_{n+1} / S_n$  を用いたが、数 ( 1 6 ) 式で示すように、第  $n$  期間における入力平均ビットレート  $S_n$  に代えて、第 (  $n - M + 1$  ) 期間から第  $n$  期間までの期間平均ビットレート  $AS_n$  を用いて量子化ステップ変換係数  $\alpha_{n+1}$  を算出してもよい。

【 0 0 8 0 】

30

【 数 1 6 】

$$\alpha_{n+1} = f(T_{n+1} / AS_n) \quad \dots(16)$$

【 0 0 8 1 】

また、目標設定ビットレート  $T_{n+1}$  を算出する演算過程において、数 ( 6 ) 式で示した場合のように、期間比として第 (  $n + 1$  ) 期間の第 1 ストリームの入力平均ビットレート  $S_{n+1}$  を利用した場合には、数 ( 8 ) 式に代えて、数 ( 1 7 ) 式で示すように、第 (  $n + 1$  ) 期間の目標設定ビットレート  $T_{n+1}$  と、第 (  $n + 1$  ) 期間の入力平均ビットレート  $S_{n+1}$  の比を関数  $f$  (関数  $f_1$ ) に入力することで、第 (  $n + 1$  ) 期間の量子化ステップ変換係数  $\alpha_{n+1}$  を算出するようにすればよい。

40

【 0 0 8 2 】

【 数 1 7 】

$$\alpha_{n+1} = f(T_{n+1} / S_{n+1}) \quad \dots(17)$$

【 0 0 8 3 】

また、目標設定ビットレート  $T_{n+1}$  を算出する演算過程において、期間比として、第

50

( $n + 1$ ) 期間を含む過去  $M$  期間の期間平均ビットレート  $AS_{n+1}$  を利用した場合には、数 (16) 式に代えて、数 (18) 式で示すように、第 ( $n + 1$ ) 期間の目標設定ビットレート  $T_{n+1}$  と、第 ( $n + 1$ ) 期間を含む過去  $M$  期間の期間平均ビットレート  $AS_{n+1}$  の比を関数  $f$  (関数  $f_1$ ) に入力することで、第 ( $n + 1$ ) 期間の量子化ステップ変換係数  $\alpha_{n+1}$  を算出するようにすればよい。

【0084】

【数18】

$$\alpha_{n+1} = f(T_{n+1} / AS_{n+1}) \quad \dots(18)$$

10

【0085】

なお、第1の実施の形態において、関数  $f_1$  の傾きを -1 としたが、これは一例である。関数  $f_0$  よりも変動の小さい負の傾きで最適なものを選択すればよい。また、数 (14) 式において、閾値を 0.5 に設定したが、これも一例である。この閾値を自由に設定することで基準ビットレート比  $R_0$  (同様に更新された基準ビットレート比  $R_0$ ) の近傍領域を自由に設定可能である。

【0086】

{ 第2の実施の形態 }

次に、本発明の第2の実施の形態について説明する。第2の実施の形態と第1の実施の形態とでは関数  $f$  が異なる。数 (19) 式は、第2の実施の形態において関数  $f$  として利用される関数  $f_2$  を示す。

20

【0087】

【数19】

$$f_2(x) = \frac{R_0^2}{x} - R_0 + \varepsilon R_0^{-\delta} \quad \dots(19)$$

【0088】

関数  $f_2$  は、点 ( $R_0, f_0(R_0)$ ) を通り、かつ、点 ( $R_0, f_0(R_0)$ ) において傾き -1 の接線  $p(x)$  を有する直交双曲線 (直角双曲線) である。関数  $f_0$  は、数 (13) 式で表したものと同一である。

30

【0089】

図5は、関数  $f_2$  と関数  $f_0$  を示す図である。図中、 $x$  軸はビットレートの比を示し、 $y$  軸は量子化ステップ変換係数を示している。関数  $f_2$ 、関数  $f_0$  として具体的に  $\varepsilon = 2$ 、 $\delta = 1$  の値を用いている。上述したように、関数  $f_0$  は、指数関数であり、ビットレートの比が小さい領域については、その傾きが大きくなっている。そこで、基準ビットレート比  $R_0$  の近傍において、関数  $f_0$  よりも変動の小さい関数  $f_2$  を用いて量子化ステップ変換係数を算出することとしている。

40

【0090】

第1の実施の形態と同様、関数  $f_0$  は、ビットレートの比と量子化ステップ変換係数との関係を決めるおおよそのエリアを規定する領域指定関数としての役割を有し、そのエリアの中で、関数  $f_2$  は、ビットレートの比と量子化ステップ変換係数との関係を具体的に決定する関係決定関数としての役割を有する。

【0091】

関数  $f_2$  は、点 ( $R_0, f_0(R_0)$ ) において、関数  $f_0$  と交わっている。つまり、基準ビットレート比  $R_0$  の近傍においては、関数  $f_0$  と比較的近い特性を維持している。しかし、関数  $f_2$  が、点 ( $R_0, f_0(R_0)$ ) から離れるにつれて、関数  $f_2$  は関数  $f_0$  と特性が乖離する。関数  $f_2$  が関数  $f_0$  から特性が大きくなると、関数  $f_0$  の特性を

50

維持できなくなり、関数  $f_2$  を使用し続けるのは不適切となる。そこで、数 (20) 式の関係を満たす限りにおいて、関数  $f_2$  を利用するのである。

【0092】

【数20】

$$|f_0(R_B) - f_2(R_B)| \leq 0.5 \quad \dots(20)$$

【0093】

そして、数 (20) 式の関係を満たさなくなった場合には、数 (15) 式に示す現在ビットレート比  $R_B$  により、基準ビットレート比  $R_0$  を更新するのである。これにより、基準点 ( $R_0, f_0(R_0)$ ) が更新され、あわせて関数  $f_2$  も更新される。つまり、関数  $f_0$  のエリアに沿って、関数  $f_2$  を移動させながら、関数  $f_2$  を用いて量子化ステップ変換係数を算出するのである。これにより、関数  $f_0$  と関数  $f_2$  の両方の特性を利用して、最適なレート制御を行うことが可能である。

10

【0094】

なお、第2の実施の形態において、接線  $p(x)$  の傾きを -1 としたが、これは一例である。関数  $f_2$  の変動が関数  $f_0$  よりも小さくなるような負の傾きで最適なものを選択すればよい。また、数 (20) 式において、閾値を 0.5 に設定したが、これも一例である。この閾値を自由に設定することで基準ビットレート比  $R_0$  (同様に更新された基準ビットレート比  $R_0$ ) の近傍領域を自由に設定可能である。

20

【0095】

{ 第3の実施の形態 }

次に、本発明の第3の実施の形態について説明する。第3の実施の形態と第1の実施の形態とは関数  $f$  が異なる。数 (21) 式は、第3の実施の形態において関数  $f$  として利用される関数  $f_3$  を示す。

【0096】

【数21】

$$f_3(x) = -\varepsilon \delta R_0^{-(1+\delta)} \left( x - \frac{1+\delta}{\delta} R_0 \right) \quad \dots(21)$$

30

【0097】

関数  $f_3$  は、点 ( $R_0, f_0(R_0)$ ) を通り、かつ、点 ( $R_0, f_0(R_0)$ ) において関数  $f_0$  と接する直線である。関数  $f_0$  は、数 (13) 式で表したものと同一である。

【0098】

図6は、関数  $f_3$  と関数  $f_0$  を示す図である。図中、 $x$  軸はビットレートの比を示し、 $y$  軸は量子化ステップ変換係数を示している。関数  $f_3$ 、関数  $f_0$  として具体的に  $\varepsilon = 2$ 、 $\delta = 1$  の値を用いている。

40

【0099】

ここで、関数  $f_3$  は、関数  $f_1$  や関数  $f_2$  とは異なる特性を有する。つまり、点 ( $R_0, f_0(R_0)$ ) よりビットレート比が小さい領域については、量子化ステップ変換係数の変動を小さくするよう作用するが、点 ( $R_0, f_0(R_0)$ ) よりビットレート比が大きい領域については、逆に変動が大きくなる。

【0100】

第1、第2の実施の形態と同様、関数  $f_0$  は、ビットレートの比と量子化ステップ変換係数との関係を決めるおおよそのエリアを規定する領域指定関数としての役割を有し、そのエリアの中で、関数  $f_3$  は、ビットレートの比と量子化ステップ変換係数との関係を具

50

体的に決定する関係決定関数としての役割を有する。

【 0 1 0 1 】

また、関数  $f_3$  を利用する条件も第 1、第 2 の実施の形態と同様である。数 ( 2 2 ) 式の関係を満たす限りにおいて、関数  $f_3$  を利用するのである。

【 0 1 0 2 】

【 数 2 2 】

$$|f_0(R_B) - f_3(R_B)| \leq 0.5 \quad \dots(22)$$

10

【 0 1 0 3 】

ただし、第 3 の実施の形態においては、関数  $f_3$  の利用条件を限定的にする方が好ましい。上述したように、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比が大きい領域については、量子化ステップ変換係数の変動幅が大きくなるが、この現象は、ビットレートの比が点 (  $R_0$  ,  $f_0(R_0)$  ) より離れるにつれて顕著になる。したがって、数 ( 2 2 ) 式において、閾値の値を 0.5 より小さい値、0.2 や 0.3 などを設定し、より点 (  $R_0$  ,  $f_0(R_0)$  ) の近傍でのみ関数を利用するようにした方が好ましい。

【 0 1 0 4 】

数 ( 2 2 ) 式の関係を満たさなくなった場合には、数 ( 1 5 ) 式に示す現在ビットレート比  $R_B$  により、基準ビットレート比  $R_0$  を更新する点も同様である。

20

【 0 1 0 5 】

{ 変形例 }

上記第 1 ~ 第 3 の実施の形態においては、数 ( 1 4 )、( 2 0 )、( 2 2 ) 式で示したように、関数  $f_1$ 、 $f_2$ 、 $f_3$  の利用条件を設けた。数 ( 1 4 )、( 2 0 )、( 2 2 ) 式では、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比の大きい領域、小さい領域で同じ条件としているが、異なる条件としてもよい。たとえば、関数  $f_1$  について、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比の小さい領域では、数 ( 1 4 ) 式をそのまま用い、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比の大きい領域では、数 ( 1 4 ) 式において閾値 0.5 を 0.3 など置き換え、利用条件を厳しくするなどの方法が考えられる。

【 0 1 0 6 】

関数  $f_3$  については、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比が大きい領域については、量子化ステップ変換係数の変動幅が大きくなることを述べた。そして、この現象は、ビットレート比が点 (  $R_0$  ,  $f_0(R_0)$  ) より離れるにつれて顕著になる。そこで、関数  $f_3$  については、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比の小さい領域では、数 ( 2 2 ) 式をそのまま用い、点 (  $R_0$  ,  $f_0(R_0)$  ) よりビットレート比の大きい領域では、数 ( 2 2 ) 式において閾値 0.5 を 0.2 など置き換え、利用条件を厳しくすればよい。関数  $f_3$  は、関数  $f_0$  の接線であるので、ビットレート比の小さい領域では、比較的関数  $f_0$  に近い特性を有し、関数  $f_0$  よりも緩やかな変動特性を持たせることができ、ビットレート比の大きい領域では、点 (  $R_0$  ,  $f_0(R_0)$  ) のごく近傍でのみ関数  $f_3$  を利用することで、関数  $f_0$  の特性から大きく乖離することを防止できる。

30

40

【 0 1 0 7 】

点 (  $R_0$  ,  $f_0(R_0)$  ) を境界として利用する関数  $f$  を異ならせることもできる。たとえば、点 (  $R_0$  ,  $f_0(R_0)$  ) よりもビットレート比の小さい領域では、関数  $f_3$  を利用し、点 (  $R_0$  ,  $f_0(R_0)$  ) よりもビットレート比の大きい領域では、関数  $f_2$  を利用するのである。これにより、ビットレート比の小さい領域では、関数  $f_3$  を利用することで、比較的関数  $f_0$  に近い特性を有し、関数  $f_0$  よりも緩やかな変動特性を持たせることができ、ビットレート比の大きい領域では、関数  $f_2$  を利用することで、関数  $f_0$  の特性から大きく乖離することなく、緩やかな変動特性を持たせることができる。

【 0 1 0 8 】

あるいは、点 (  $R_0$  ,  $f_0(R_0)$  ) よりもビットレート比の小さい領域では、関数  $f$

50

$f_1$  や関数  $f_2$  利用し、点  $(R_0, f_0(R_0))$  よりもビットレート比の大きい領域では、関数  $f_0$  をそのまま利用することもできる。

【図面の簡単な説明】

【0109】

【図1】トランスコーダのブロック図である。

【図2】入力ストリーム（第1ストリーム）と出力ストリーム（第2ストリーム）の情報を制御単位期間ごとに示した図である。

【図3】関数  $f_1$  と関数  $f_0$  の特性を示す図である。

【図4】基準ビットレート比  $R_0$  の更新と関数  $f_1$  の更新を示す図である。

【図5】関数  $f_2$  と関数  $f_0$  の特性を示す図である。

【図6】関数  $f_3$  と関数  $f_0$  の特性を示す図である。

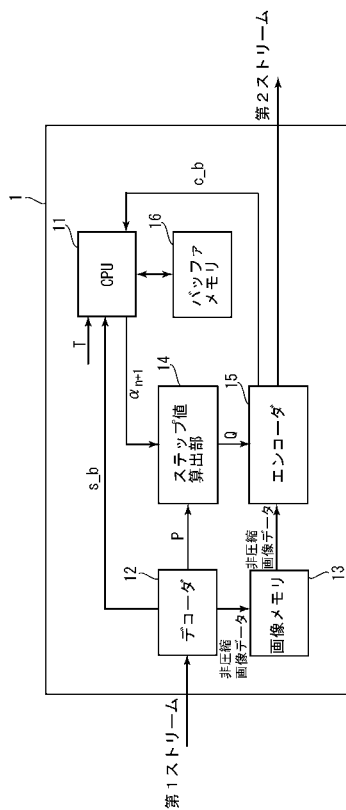
【符号の説明】

【0110】

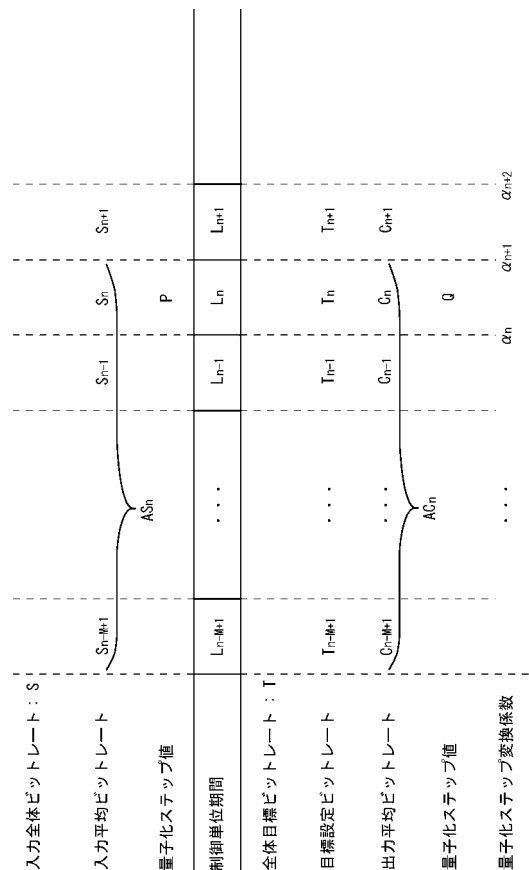
- 1 トランスコーダ
- 11 CPU
- 12 デコーダ
- 14 ステップ値算出部
- 15 エンコーダ

10

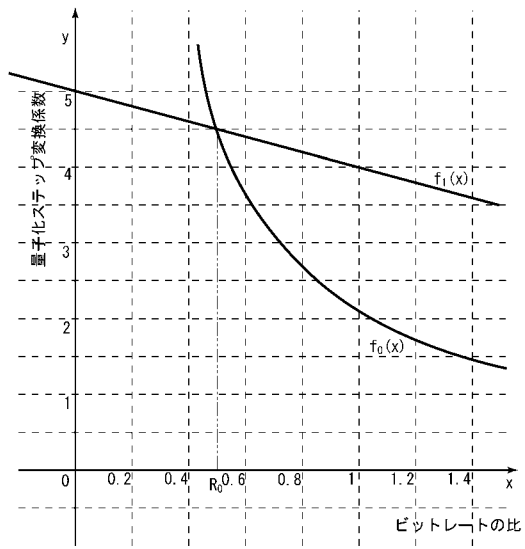
【図1】



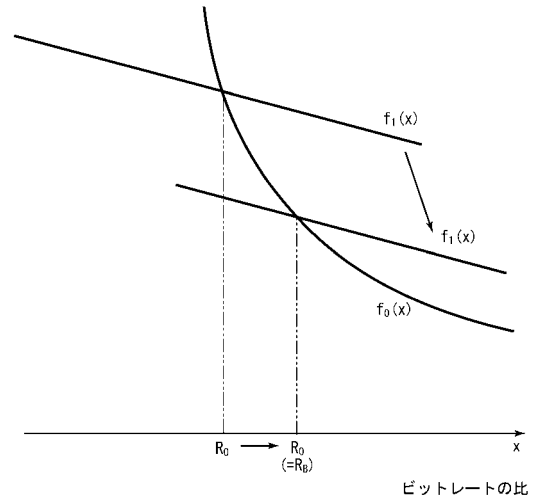
【図2】



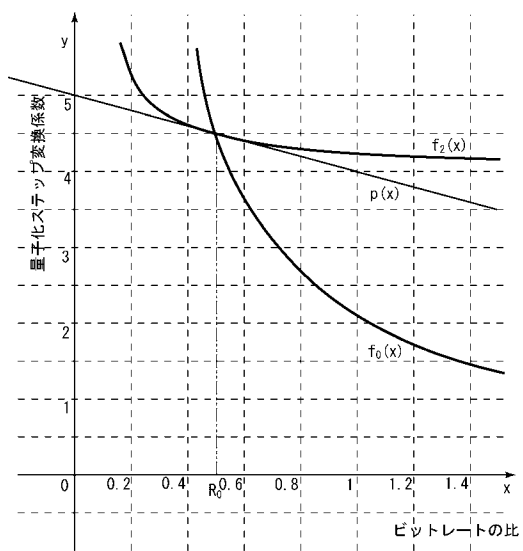
【図 3】



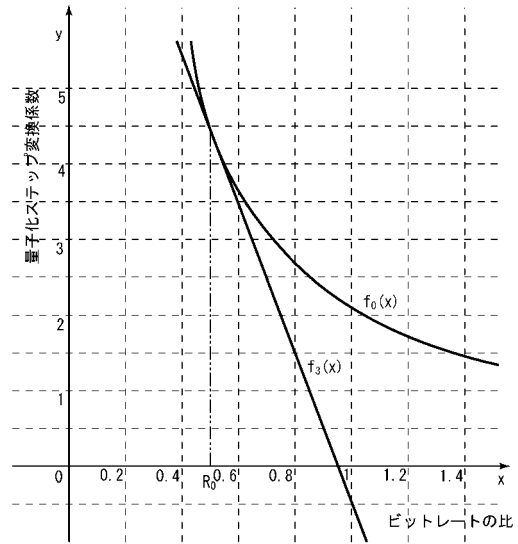
【図 4】



【図 5】



【図 6】



---

フロントページの続き

Fターム(参考) 5C159 KK52 MA00 TA46 TB01 TC37 TC38 TD14 TD16 UA02 UA05  
UA32