

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第4728568号  
(P4728568)

(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

G 1 O L 19/00 (2006.01)

G 1 O L 19/00 2 5 0

G 1 O L 19/00 3 3 0 B

請求項の数 25 外国語出願 (全 42 頁)

(21) 出願番号 特願2003-311939 (P2003-311939)  
 (22) 出願日 平成15年9月3日 (2003.9.3)  
 (65) 公開番号 特開2004-258603 (P2004-258603A)  
 (43) 公開日 平成16年9月16日 (2004.9.16)  
 審査請求日 平成18年8月29日 (2006.8.29)  
 (31) 優先権主張番号 60/408,538  
 (32) 優先日 平成14年9月4日 (2002.9.4)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 500046438  
 マイクロソフト コーポレーション  
 アメリカ合衆国 ワシントン州 9805  
 2-6399 レッドモンド ワン マイ  
 クロソフト ウェイ  
 (74) 代理人 100077481  
 弁理士 谷 義一  
 (74) 代理人 100088915  
 弁理士 阿部 和夫  
 (72) 発明者 サンジーブ メロトラ  
 アメリカ合衆国 98033 ワシントン  
 州 カークランド ノースイースト 12  
 7 アベニュー 10204

最終頁に続く

(54) 【発明の名称】 レベル・モードとラン・レンジス／レベル・モードの間での符号化を適応させるエントロピー符  
 号化

(57) 【特許請求の範囲】

【請求項 1】

コンピュータ・システムにおいてオーディオ・データを符号化する方法であって、  
 オーディオ・データ・シーケンスの第1の部分、異なる次元の複数のハフマン符号テ  
 ーブル間の変更を示すために複数のエスケープ符号を使用する直接可変次元ベクトル・ハ  
 フマン符号化モードで符号化するステップであって、高次元ベクトル・ハフマン符号テ  
 ーブルにおいて前記オーディオ・データ・シーケンスの前記第1の部分からの複数のシンボ  
 ルのベクトルにハフマン符号が割り当てられていない場合に、前記複数のエスケープ符号  
のうちの1つのエスケープ符号を追加することにより、前記複数のハフマン符号テーブル  
のうちの前記高次元ベクトル・ハフマン符号テーブルから前記複数のハフマン符号テーブル  
のうちの低次元ベクトル・ハフマン符号テーブルに変更して、前記複数のシンボルのベ  
クトルを符号化することを含むステップと、

切替えポイントでラン・レベル符号化モードに切り換えるステップと、

前記オーディオ・データ・シーケンスの第2の部分の前記ラン・レベル符号化モードで  
 符号化するステップと  
 を備えることを特徴とする方法。

【請求項 2】

前記切替えポイントを示すフラグを、符号化されたビットストリームの中で送ることを  
 さらに備えることを特徴とする請求項1に記載の方法。

【請求項 3】

前記オーディオ・データ・シーケンスの前記第 1 の部分は、主にゼロでない量子化されたオーディオ係数から成り、前記オーディオ・データ・シーケンスの前記第 2 の部分は、主にゼロ値の量子化されたオーディオ係数から成ることを特徴とする請求項 1 に記載の方法。

【請求項 4】

前記切替えポイントは、予め決められた切替えポイントであることを特徴とする請求項 1 に記載の方法。

【請求項 5】

前記予め決められた切替えポイントを使用して前記オーディオ・データ・シーケンスの符号化効率を実験することによって、前記予め決められた切替えポイントは実験的に決められることを特徴とする請求項 4 に記載の方法。

10

【請求項 6】

前記切替えポイントは、適応して決定されることを特徴とする請求項 1 に記載の方法。

【請求項 7】

第 2 の切替えポイントで第 3 の符号化モードに切り換えることをさらに備えることを特徴とする請求項 1 に記載の方法。

【請求項 8】

前記ラン・レベル符号化モードは、ラン・レンジおよびレベルのコンテキスト・ベースの算術符号化を備えることを特徴とする請求項 1 に記載の方法。

【請求項 9】

20

前記ラン・レベル符号化モードは、ラン・レンジおよびレベルのハフマン符号化を備えることを特徴とする請求項 1 に記載の方法。

【請求項 10】

前記ラン・レベル符号化モードは、ラン・レンジおよびレベルのベクトル・ハフマン符号化を備えることを特徴とする請求項 1 に記載の方法。

【請求項 11】

コンピュータに、請求項 1 に記載の方法を行わせるためのプログラムを記憶していることを特徴とするコンピュータ可読媒体。

【請求項 12】

前記オーディオ・データ・シーケンスの前記第 1 の部分を前記直接可変次元ベクトル・ハフマン符号化モードで符号化するステップは、

30

前記複数のシンボルのベクトルを符号化するために使用するハフマン符号を、前記複数のシンボルの値の合計に基づいて決定するステップと、

前記ハフマン符号を使用して前記複数のシンボルのベクトルを符号化するステップとを備えることを特徴とする請求項 1 に記載の方法。

【請求項 13】

前記ハフマン符号はエスケープ符号であり、前記複数のシンボルのベクトルは、 $n$  次元ベクトルであり、前記エスケープ符号は、前記  $n$  次元ベクトルが、 $x$  個の  $n/x$  次元ベクトルとして符号化されるべきことを示すことを特徴とする請求項 12 に記載の方法。

【請求項 14】

40

前記オーディオ・データ・シーケンスの前記第 1 の部分を前記直接可変次元ベクトル・ハフマン符号化モードで符号化するステップは、

前記オーディオ・データ・シーケンスの前記第 1 の部分からの複数のシンボルの第 1 の  $n$  次元ベクトルに、前記複数のハフマン符号テーブルのうちの  $n$  次元ベクトルハフマン符号テーブルにおいてハフマン符号が割り当てられていることを判定するステップであって、 $n$  は少なくとも 2 であり、前記  $n$  次元ベクトルハフマン符号テーブルは、すべての可能な  $n$  次元ベクトルよりも少ない数のハフマン符号を含むものであるステップと、

前記  $n$  次元ベクトルハフマン符号テーブルから割り当てられたハフマン符号を使用して前記第 1 の  $n$  次元ベクトルを符号化するステップとを備え、

50

前記オーディオ・データ・シーケンスの前記第 1 の部分からの複数のシンボルの第 2 の  $n$  次元ベクトルに、前記  $n$  次元ベクトルハフマン符号テーブルにおいてハフマン符号が割り当てられていないことの判定に応答して、

前記複数のハフマン符号テーブルのうちの  $n / 2$  次元ベクトルハフマン符号テーブルへの変更を示すエスケープ符号を追加するステップと、

前記第 2 の  $n$  次元ベクトルを 2 つの  $n / 2$  次元ベクトルに分割するステップと、

前記 2 つの  $n / 2$  次元ベクトルに、前記  $n / 2$  次元ベクトルハフマン符号テーブルにおいてハフマン符号が割り当てられていることを判定するステップであって、前記  $n / 2$  次元ベクトルハフマン符号テーブルはすべての可能な  $n / 2$  次元ベクトルよりも少ない数のハフマン符号を含むものであるステップと、

前記  $n / 2$  次元ベクトルハフマン符号テーブルから割り当てられたハフマン符号を使用して前記 2 つの  $n / 2$  次元ベクトルを符号化するステップと  
を備えることを特徴とする請求項 1 に記載の方法。

【請求項 15】

コンピュータ・システムにおいてオーディオ・データを復号化するための方法であって、

符号化されたオーディオ・データ・シーケンスの第 1 の部分を、異なる次元の複数のハフマン符号テーブル間の変更を示すために複数のエスケープ符号を使用する直接可変次元ベクトル・ハフマン復号化モードで復号化するステップであって、高次元ベクトル・ハフマン符号テーブルのエスケープ符号が前記符号化されたオーディオ・データ・シーケンス内に存在する場合に、前記複数のハフマン符号テーブルのうちの前記高次元ベクトル・ハフマン符号テーブルから前記複数のハフマン符号テーブルのうちの低次元ベクトル・ハフマン符号テーブルに変更して復号化することを含むステップと、

切替えポイントでラン・レベル復号化モードに切り換えるステップと、

前記符号化されたオーディオ・データ・シーケンスの第 2 の部分を前記ラン・レベル復号化モードで復号化するステップと  
を備えることを特徴とする方法。

【請求項 16】

復号化モードの前記切り換えに先立って、前記切替えポイントを示すフラグを受け取ることをさらに備えることを特徴とする請求項 15 に記載の方法。

【請求項 17】

前記符号化されたオーディオ・データ・シーケンスの前記第 1 の部分は、主にゼロでない量子化されたオーディオ係数から成り、前記符号化されたオーディオ・データ・シーケンスの前記第 2 の部分は、主にゼロ値の量子化されたオーディオ係数から成ることを特徴とする請求項 15 に記載の方法。

【請求項 18】

前記切替えポイントは、予め決められた切替えポイントであることを特徴とする請求項 15 に記載の方法。

【請求項 19】

前記切替えポイントは、適応して決定されることを特徴とする請求項 15 に記載の方法。

【請求項 20】

第 2 の切替えポイントで第 3 の復号化モードに切り換えることをさらに備えることを特徴とする請求項 15 に記載の方法。

【請求項 21】

前記ラン・レベル復号化モードは、ラン・レンジおよびレベルのコンテキスト・ベースの算術復号化を備えることを特徴とする請求項 15 に記載の方法。

【請求項 22】

前記ラン・レベル復号化モードは、ラン・レンジおよびレベルのハフマン復号化を備えることを特徴とする請求項 15 に記載の方法。

## 【請求項 2 3】

前記ラン・レベル復号化モードは、ラン・レンジスおよびレベルのベクトル・ハフマン復号化を備えることを特徴とする請求項 1 5 に記載の方法。

## 【請求項 2 4】

コンピュータに、請求項 1 5 に記載の方法を行わせるためのプログラムを記憶していることを特徴とするコンピュータ可読媒体。

## 【請求項 2 5】

前記符号化されたオーディオ・データ・シーケンスの前記第 1 の部分を前記直接可変次元ベクトル・ハフマン復号化モードで復号化するステップは、

前記符号化されたオーディオ・データ・シーケンスの第 1 のハフマン符号が前記複数のハフマン符号テーブルのうちの  $n$  次元ベクトルハフマン符号テーブルのエスケープ符号であると判定するステップであって、 $n$  は少なくとも 2 であり、前記  $n$  次元ベクトルハフマン符号テーブルはすべての可能な  $n$  次元ベクトルよりも少ない数のハフマン符号を含むものであるステップと、

前記符号化されたオーディオ・データ・シーケンスの前記第 1 のハフマン符号が前記  $n$  次元ベクトルハフマン符号テーブルの前記エスケープ符号であるとの判定にตอบสนองして、前記複数のハフマン符号テーブルのうちの  $n / 2$  次元ベクトルハフマン符号テーブルを使用して、前記符号化されたオーディオ・データ・シーケンスの第 2 のハフマン符号を復号化するステップと

を備えることを特徴とする請求項 1 5 に記載の方法。

## 【発明の詳細な説明】

## 【技術分野】

## 【0 0 0 1】

本発明は、オーディオ・データの適応エントロピー符号化に関する。例えば、オーディオ符号器は、量子化されたオーディオ・データの直接レベルのハフマン符号化と、量子化されたオーディオ・データのラン・レンジスおよびレベルの算術符号化の間で切替えを行う。

## 【背景技術】

## 【0 0 0 2】

コンパクト・ディスク、デジタル無線電話網、およびインターネットを介するオーディオの配信の導入とともに、デジタル・オーディオが一般的になっている。技術者は、様々なテクニック (technique; 技術、方法) を使用して、デジタル・オーディオの品質を維持しながらも、デジタル・オーディオを効率的に処理している。これらのテクニックを理解することは、オーディオ情報がコンピュータにおいてどのように表わされ、処理されるかを理解することに役立つ。

## 【0 0 0 3】

## I. コンピュータにおけるオーディオ情報の表現

コンピュータは、オーディオ情報を、オーディオ情報を表わす複数の数の連続として処理する。例えば、一つの数が、オーディオ・サンプルを表わすことが可能であり、それは、特定の時点における振幅値 (すなわち、ラウドネス) である。サンプル深度 (sample depth)、サンプリング・レート、およびチャネル・モードを含むいくつかのファクタ (factor) が、オーディオ情報の品質に影響を与える。

## 【0 0 0 4】

サンプル深度 (または精度) は、サンプルを表わすのに使用される数の範囲を示す。サンプルに関して使用可能であるその値が多いほど、品質が高くなる。というのは、その数が、振幅のより微妙な変動をキャプチャすることができるからである。例えば、8 ビットのサンプルは、256 の使用可能な値を有するが、16 ビットのサンプルは、65,536 の使用可能な値を有する。

## 【0 0 0 5】

また、サンプリング・レート (通常、毎秒のサンプル数で測定される) も、品質に影響

10

20

30

40

50

を与える。サンプリング・レートが高いほど、品質が高くなる。というのは、サウンドについてのより多くの周波数を表わすことができるからである。いくつかの一般的なサンプリング・レートは、毎秒 8,000 サンプル、毎秒 11,025 サンプル、毎秒 22,050 サンプル、毎秒 32,000 サンプル、毎秒 44,100 サンプル、毎秒 48,000 サンプル、および毎秒 96,000 サンプルである。

【0006】

表 1 は、異なる品質レベルを有するいくつかのオーディオ形式、ならびに対応する生のビットレート・コストを示している。

【0007】

【表 1】

表 1：異なる品質のオーディオ情報に関するビット・レート

品質	サンプル深度 (ビット/ サンプル)	サンプリング・ レート (サンプル/秒)	モード	生の ビット・レート (ビット/秒)
インターネット電話	8	8,000	モノ	64,000
電話	8	11,025	モノ	88,200
CDオーディオ	16	44,100	ステレオ	1,411,200
高品質オーディオ	16	48,000	ステレオ	1,536,000

【0008】

表 1 が示すとおり、CDオーディオなどの高品質オーディオ情報の費用は、高いビット・レートである。高品質オーディオ情報は、大量のコンピュータ・ストレージおよび伝送容量を消費する（使い尽くす）。ただし、企業および消費者は、高品質のオーディオ・コンテンツを作成し、配信し、再生するのに、ますますコンピュータに頼っている。

【0009】

II. オーディオの圧縮および圧縮解除

多くのコンピュータおよびコンピュータ網は、生のデジタル・オーディオを処理するリソースを欠いている。圧縮（符号化またはコーディングとも呼ばれる）は、情報をより低いビット・レートの形態に変換することによって、オーディオ情報の記憶および伝送の費用を低減する。圧縮は、ロスレス(lossless：損失がない)（品質が低下しない）、またはロッシー(lossy：損失がある)（品質が低下するが、ロスレス圧縮を超えたビット・レート低減が、より著しい）とすることが可能である。圧縮解除（復号化とも呼ばれる）が、圧縮された形態から元の情報の再構成されたバージョンを取り出す。

【0010】

一般に、オーディオ圧縮の目的は、可能な限り少量のビットで最大限の信号品質を提供するようにオーディオ信号をデジタル式に表わすことである。従来のオーディオ符号器/復号器[「コーデック」]システムは、サブバンド/変換符号化、量子化、速度制御、および可変長符号化を使用して圧縮を実現する。量子化およびその他のロッシー圧縮テクニックにより、聴き取られる可能性がある雑音はオーディオ信号に導入される。雑音の可聴性は、存在する雑音の量、また聴取者が知覚する雑音の量に依存する。第 1 のファクタは、主に客観的品質に関するが、第 2 のファクタは、サウンドの人間による知覚に依存する。したがって、従来のオーディオ符号器は、ビット・レートをさらに低下させるために、可変長符号化を使用して量子化されたデータを、損失なく、圧縮していた。

【0011】

A. オーディオ・データのロッシー圧縮および圧縮解除

従来、オーディオ符号器は、様々な異なるロッシーの圧縮テクニック技術を使用する。このロッシーの圧縮テクニックは、通常、周波数変換、知覚モデル化/重み付け、および量子化を必要とする。対応する圧縮解除には、逆量子化、逆重み付け、および逆周波数変換を必要とする。

【0012】

10

20

30

40

50

周波数変換テクニックは、知覚的に重要な情報を知覚的に重要でない情報から分けることをより容易にする形態にデータを変換する。次に、それほど重要でない情報は、より損失の多い圧縮を受けるようにし、他方、より重要な情報は、保存されるようにして、所与のビット・レートに関して最良の知覚される品質を提供するようにすることができる。周波数変換器が、通常、オーディオ・サンプルを受け取り、ときとして周波数係数またはスペクトル係数と呼ばれる周波数領域のデータに、それを変換する。

#### 【 0 0 1 3 】

会話や音楽などの自然のサウンドにおけるほとんどのエネルギーは、低周波数範囲に集中している。このことは、統計上、より高い周波数範囲におけるエネルギーの欠如を反映して、より高い周波数範囲がゼロまたはゼロに近い周波数係数をより多く有することになる、ことを意味する。

10

#### 【 0 0 1 4 】

知覚モデル化は、所与のビット・レートで再構成されたオーディオ信号の知覚される品質を向上させるように、人間聴覚システムのモデルに従ってオーディオ・データを処理することに意味する。例えば、聴覚モデルは、通常、人間の聴力の範囲、および臨界帯域 (critical bands) を考慮する。知覚モデル化の結果を使用して、符号器は、所与のビット・レートに対して雑音の可聴性を最小限に抑える目的にもかかわらず、オーディオ・データ中に雑音 (例えば、量子化雑音) を成形する。符号器は、ときとして、ビット・レートを低下させるために、雑音 (例えば、量子化雑音) を導入しなければならないが、重み付けにより、符号器は、雑音がそれほど聞き取れない帯域により多くの雑音を入れ、雑音がよく聞き取れる帯域にはより少ない雑音しか入れないことが可能になる。

20

#### 【 0 0 1 5 】

量子化は、入力値それぞれの範囲それぞれを、複数の単一の値にマップして不可逆性の情報の損失つまり量子化雑音を導入するが、符号器が、出力の品質とビット・レートを調整することも可能にする。ときとして、符号器は、ビット・レートおよび/または品質を調整するために量子化を調整する速度コントローラと連携して、量子化を達成する。適応量子化および非適応量子化、スカラー量子化およびベクトル量子化、一様な量子化および一様でない量子化を含め、様々な種類の量子化が存在する。知覚重み付け (perceptual weighting) は、ある形態の一様でない量子化と考えることができる。

30

#### 【 0 0 1 6 】

逆量子化および逆重み付けにより、重みが付けられて量子化された周波数係数データが、元の周波数係数データの近似値に再構成される。次に、逆周波数変換器が、再構成された周波数係数データを再構成された時間領域オーディオ・サンプルに変換する。

#### 【 0 0 1 7 】

##### B . オーディオ・データのロスレス圧縮および圧縮解除

従来、オーディオ符号器は、様々な異なるロスレス圧縮テクニックの1つまたは複数を使用する。一般に、ロスレス圧縮テクニックには、ラン・レンジス符号化、ハフマン符号化、および算術符号化が含まれる。対応する圧縮解除テクニックには、ラン・レンジス復号化、ハフマン復号化、および算術復号化が含まれる。

#### 【 0 0 1 8 】

40

ラン・レンジス符号化は、カメラ・ビデオ、テキスト、およびその他のタイプのコンテンツに対して使用される単純な周知の圧縮テクニックである。一般に、ラン・レンジス符号化は、同一の値を有する連続するシンボルのシーケンス (すなわち、ラン (run)) をそのシーケンスの値および長さで置き換える。ラン・レンジス復号化では、連続するシンボルのシーケンスが、ランの値およびランの長さ (run value and run length) から再構成される。ラン・レンジス符号化/復号化の多数の変種が開発されている。ラン・レンジス符号化/復号化、およびラン・レンジス符号化/復号化の変種のいくつかに関するさらなる情報については、例えば、非特許文献1、非特許文献2、M a i r s 他に発行された特許文献1、G i l l 他に発行された特許文献2、およびC h a d d h a に発行された特許文献3を参照されたい。

50

## 【 0 0 1 9 】

ラン・レベル符号化は、同一の値を有する連続するシンボルのランが、ランの長さで置き換えられることで、ラン・レンジ符号化と同様である。ランの値は、データにおける顕著な値（例えば、0）であり、ランは、異なる値（例えば、ゼロでない値）を有する1つまたは複数のレベルによって隔てられる。

## 【 0 0 2 0 】

ラン・レンジ符号化（例えば、ランの値およびランの長さ）またはラン・レベル符号化の結果をハフマン符号化して、さらにビット・レートを低下させることが可能である。これが行われた場合、ハフマン符号化されたデータは、ラン・レンジ復号化に先立ってハフマン復号化される。

10

## 【 0 0 2 1 】

ハフマン符号化は、カメラ・ビデオ、テキスト、およびその他のタイプのコンテンツに対して使用される別の周知の圧縮テクニックである。一般に、ハフマン符号テーブルは、可変長のハフマン符号を一意的なシンボル値（または、値の一意的な組合せ）に関連付ける。より短い符号に、より確率の高いシンボル値が割り当てられ、より長い符号に、それほど確率の低いシンボル値が割り当てられる。この確率は、いくつかの種類のコンテンツの典型的な例に対して計算される。あるいは、この確率は、符号化されたばかりのデータ、または符号化されるデータに関して計算され、その場合、ハフマン符号は、一意的なシンボル値に関する確率の変化に適応する。静的なハフマン符号化と比べて、適応ハフマン符号化は、通常、データに関するより正確な確率を組み込むことによって圧縮されたデータのビット・レートを低下させるが、ハフマン符号を指定する追加の情報も伝送される必要がある可能性がある。

20

## 【 0 0 2 2 】

シンボルを符号化するのに、ハフマン符号器は、シンボル値を、ハフマン符号テーブルの中でシンボル値に関連づけられた可変長のハフマン符号に置き換える。復号化するのに、ハフマン復号器は、ハフマン符号を、ハフマン符号に関連づけられたシンボル値に置き換える。

## 【 0 0 2 3 】

スカラー・ハフマン符号化では、ハフマン符号テーブルは、1つの値に、例えば、量子化されたデータ値の直接レベルに、単一のハフマン符号を関連付ける。ベクトル・ハフマン符号化では、ハフマン符号テーブルは、値の組合せに、例えば、特定の順序の量子化されたデータ値の一群の直接レベルに単一のハフマン符号を関連付ける。ベクトル・ハフマン符号化は、スカラー・ハフマン符号化より（例えば、符号器が、2進ハフマン符号における確率を分数として利用することを可能にすることにより）良好なビット・レート低下をもたらす可能性がある。他方、ベクトル・ハフマン符号化用のコードブックは、単一の符号が大きいグループのシンボルを表わす場合、またはシンボルが大きい範囲の可能な値を有する（可能な組合せの数が多いことにより）場合、極端に大きい可能性がある。例えば、アルファベット・サイズが256（シンボル当たり0ないし255の値として）であり、ベクトル当たりのシンボルの数が4である場合、可能な組合せの数は、 $256^4 = 4,294,967,296$ である。これは、コードブックを計算し、ハフマン符号を見出す際に、メモリおよび処理資源（processing resources）を消費し（使い尽くし）、コードブックを伝送する際に伝送リソースを消費する（使い尽くす）。

30

40

## 【 0 0 2 4 】

ハフマン符号化／復号化の多数の変種が開発されている。ハフマン符号化／復号化、およびハフマン符号化／復号化の変種のいくつかに関するさらなる情報については、例えば、非特許文献1、非特許文献2を参照されたい。

## 【 0 0 2 5 】

Chen他に発行された特許文献4は、オーディオ・データのマルチ・レベル・ラン・レンジ符号化を説明している。周波数変換は、一続きの周波数係数値をもたらす。顕著な値がゼロである周波数スペクトルの部分に関して、マルチ・レベル・ラン・レンジ符

50

号器は、ゼロ値のランを隣接するゼロでない値に統計的に相関させ、可変長の符号語を割り当てる。符号器は、ゼロ値を有するスペクトル係数の後にゼロでない係数が続く入力ランを受け取る確率に関して生成された、特殊化されたコードブックを使用する。対応する復号器は、可変長の符号語を、ゼロ値の係数と隣接するゼロでない値の係数のランに関連付ける。

【0026】

C h e n 他に発行された特許文献5は、オーディオ・データの可変 - 可変長符号化を説明している。符号器は、周波数係数値の可変サイズのグループに可変長の符号を割り当てる。

【0027】

C h e n 他に発行された特許文献6は、周波数領域オーディオ符号化に関するエントロピー符号モード切替えを説明している。周波数領域オーディオ符号器は、入力ストリームの特性に従って異なるエントロピー符号化モードのなかから選択を行う。詳細には、入力ストリームは、符号化されるべき典型的な入力または実際の入力の統計解析から導出された統計基準に従って複数の周波数範囲に区分される。各範囲には、その範囲のタイプのデータを符号化するように最適化されたエントロピー符号器が割り当てられる。符号化中および復号化中、モード選択器は、的確な方法を個々の周波数範囲に適用する。区分境界は、前もって決められ、復号器が、どの復号化方法が符号化されたデータに適用されるかを暗黙に知ることを可能にしている。あるいは、適応構成が使用されて、その構成において、出力ストリーム中で境界にフラグが設定されて、後続のデータに関する符号化モードの変更が示されるようにすることが可能である。例えば、区分境界は、主にゼロの量子化された周波数係数を、主にゼロでない量子化された係数から分離し、次に、そのようなデータに合わせて最適化された符号器を適用する。

C h e n の特許に関するさらなる詳細については、特許自体を参照されたい。

【0028】

算術符号化は、カメラ・ビデオ、およびその他のタイプのコンテンツに対して使用される別の周知の圧縮テクニックである。いくつかの個々の入力シンボルの間で統計上の相関が存在するケースで、算術符号化は、ときとして、所与の入力シンボルを符号化する最適なビット数が、分数のビット数である適用例において使用される。算術符号化は、一般に、入力シーケンスを所与の範囲内の単一の数として表わすことに関わる。通常、その数は、0と1の中間の分数である。入力シーケンス中のシンボルは、0と1の間の空間の部分を示す範囲に関連付けられる。この範囲は、特定のシンボルが入力シーケンスの中で生起する確率に基づいて計算される。入力シーケンスを表わすのに使用される分数は、その範囲に関連して構成される。したがって、入力シンボルの確率分布は、算術符号化スキームにおいて重要である。

【0029】

コンテキスト・ベースの算術符号化では、入力シンボルの異なる確率分布が、異なるコンテキストに関連付けられる。入力シーケンスを符号化するのに使用される確率分布は、コンテキストが変化する際に変化する。コンテキストは、入力シーケンスの中で生起する特定の入力シンボルの確率に影響を与えると予期される異なるファクタを測定することによって計算することができる。算術符号化/復号化、および算術符号化の変種のいくつかに関するさらなる情報については、非特許文献3を参照されたい。

【0030】

マイクロソフト・コーポレーションのWindows (登録商標) Media Audio [「WMA」] 符号器および復号器のバージョンを含む様々なコーデック・システムおよびコーデック標準が、ロスレス圧縮および圧縮解除を使用している。他のコーデック・システムは、Motion Picture Experts Group、Audio Layer 3 [「MP3」] 標準、Motion Picture Experts Group 2、Advanced Audio Coding [「AAC」] 標準、およびドルビー (Dolby) AC3によって提供されるか、または規定されている。さら

10

20

30

40

50



なる情報に関しては、それぞれの標準または技術刊行物を参照されたい。

【0031】

【特許文献1】米国特許第6,304,928号明細書

【特許文献2】米国特許第5,883,633号明細書

【特許文献3】米国特許第6,233,017号明細書

【特許文献4】米国特許第6,233,162号明細書

【特許文献5】米国特許第6,377,930号明細書

【特許文献6】米国特許第6,300,888号明細書

【非特許文献1】Bell et al., Text Compression, Prentice Hall PTR, pages 105-107, 1990

10

【非特許文献2】Gibson et al., Digital Compression for Multimedia, Morgan Kaufmann, pages 17-62, 1998

【非特許文献3】Nelson, The Data Compression Book, "Huffman One Better: Arithmetic Coding," Chapter 5, pp. 123-65(1992)

【発明の開示】

【発明が解決しようとする課題】

【0032】

オーディオ・データのロスレス圧縮についての従来の技術およびシステムの利点がどのようなものであれ、従来のテクニックおよびシステムは、本発明の利点を有さない。

【課題を解決するための手段】

20

【0033】

要約すると、詳細な説明は、オーディオ・データの適応エントロピー符号化および適応エントロピー復号化のための様々なテクニックおよびツールを対象としている。このようなテクニックおよびツールは、組合せで、または独立して使用することができる。

【0034】

一態様では、符号器は、オーディオ・データ・シーケンスの第1の部分を直接可変次元ベクトル・ハフマン符号化モード(variable-dimension vector Huffman encoding)で符号化し、切替えポイントでラン・レベル符号化モードに切り換わり、第2の部分をラン・レベル符号化モード(例えば、コンテキスト・ベースの算術符号化、ハフマン符号化、ベクトル・ハフマン符号化)で符号化する。例えば、第1の部分は、主にゼロでない量子化されたオーディオ係数から成り、第2の部分は、主にゼロ値の量子化されたオーディオ係数から成る。切替えポイントは、予め決定されていること(例えば、その切替えポイントを使用してシーケンスを符号化する効率を試験することにより)、または適応して(adaptively)決定することができる。符号器は、符号化されたビットストリームで、切替えポイントを示すフラグを送ることができる。

30

【0035】

代替の態様では、復号器は、符号化されたシーケンスの第1の部分を直接可変次元ベクトル・ハフマン復号化モードで復号化し、切替えポイントでラン・レベル復号化モードに切り換わり、第2の部分をラン・レベル復号化モード(例えば、コンテキスト・ベースの算術復号化、ハフマン復号化、ベクトル・ハフマン復号化)で復号化する。切替えを行うのに先立ち、復号器は、切替えポイントを示すフラグを受け取ることができる。

40

【0036】

別の態様では、符号器または復号器が、シーケンスの第1の部分を直接コンテキスト・ベースの算術モードで符号化または復号化し、切替えポイントでラン・レベル・モードに切り換わり、第2の部分をラン・レベル・モードで符号化または復号化する。ラン・レベル・モードは、コンテキスト・ベースの算術モードであることが可能である。

【0037】

別の態様では、符号器が、第1のベクトルにおけるシンボルの数に基づいて1組の複数の符号テーブルから第1の符号テーブルを選択し、第1のベクトルを第1の符号テーブルからの符号で表わす。第1の符号テーブルは、そのシンボル数を有する可能性の高いベク

50

トルを表わすための符号、およびそれほど可能性の高くないベクトルを表わすエスケープ符号を含むことが可能である。また、符号器は、異なるシンボル数を有する第2のベクトルも符号化する。例えば、第1のベクトルは、第2のベクトルより多い数のシンボルを有し、第2のベクトルより高い生起確率を有する。第2のベクトルを符号化するため、符号器は、第2のベクトルにおけるシンボルの数に基づいて第2の異なる符号テーブルを選択することができる。第2のベクトルが1つのシンボルを有する場合、符号器は、テーブルのない符号化テクニックを使用して第2のベクトルを表わすことができる。

【0038】

別の態様では、復号器が、第1の符号を受け取り、第1の符号テーブルの中で第1の符号をルックアップすることによって第1のベクトルを復号化する。第1の符号がエスケープ符号である場合、復号器は、第1のテーブルの中に存在しない第2の符号を受け取って復号化する。第1の符号がエスケープ符号ではない場合、復号器は、第1のテーブルにおいて第1のベクトルに関するシンボル（複数）をルックアップし、そのシンボル（複数）を復号化されたデータストリームの中を含める。第1のベクトルにおけるシンボルの数は、第1の符号がエスケープ符号であるかどうかについての基準である。復号器は、第2の符号を第2のテーブルの中でルックアップすることによって第2の符号を復号化することができる。第2の符号がエスケープ符号である場合、復号器は、第2のテーブルの中に存在しない第1のベクトルを表わす第3の符号を受け取り、復号化する。第2の符号がエスケープ符号でない場合、復号器は、第2のテーブルの中で第1のベクトルに関するシンボルをルックアップし、そのシンボルを復号化されたデータストリームの中を含める。

【0039】

別の態様では、符号器が、テーブルのない符号化テクニックを使用してオーディオ・データ係数を符号化する。係数が第1の値範囲（value range）内にある場合、符号器は、その係数を1ビット符号の後に8ビットの符号化された値が続く符号で符号化する。その他の値範囲の場合、符号器は、その係数を2ビット符号の後に16ビットの符号化された値が続く符号、3ビット符号の後に24ビットの符号化された値が続く符号、または異なる3ビット符号の後に31ビットの符号化された値が続く符号で符号化する。

【0040】

別の態様では、ベクトル・ハフマン符号化スキームにおいて、符号器が、一群のハフマン符号からベクトルを符号化するために使用するハフマン符号を決め、そのハフマン符号を使用してベクトルを符号化する。符号の決定は、ベクトルにおけるオーディオ・データのシンボルの値の合計に基づく。ハフマン符号がエスケープ符号である場合、ハフマン符号は、 $n$ 次元ベクトルが、少なくとも1つの異なる符号テーブルを使用して $x$ 個の $n/x$ 次元のベクトルとして符号化されるべきことを示す。符号器は、その合計をベクトルにおけるシンボルの数に依存するしきい値と比較することができる。例えば、しきい値は、4つのシンボルに対して6であり、2つのシンボルに対して16であり、あるいは1つのシンボルに対して100である。

【0041】

別の態様では、符号器が、オーディオ・データのシーケンスを受け取り、コンテキスト・ベースの算術符号化を使用してそのシーケンスの少なくとも一部を符号化する。復号器が、オーディオ・データ係数の符号化されたシーケンスを受け取り、コンテキスト・ベースの算術復号化を使用して符号化されたシーケンスの少なくとも一部を復号化する。

【0042】

別の態様では、符号器が、コンテキスト・ベースの算術符号化を使用してオーディオ・データ係数を符号化する。1つまたは複数のコンテキストが、係数の確率を表わす関連する確率分布を有する。符号器は、現在の係数の表現のモードに少なくともある程度、基づいて現在の係数に関するコンテキストを、適応して（adaptively）特定し、そのコンテキストを使用して現在の係数を符号化する。例えば、表現のモードが直接である場合、符号器は、前の係数（例えば、現在の係数の直前の2つの係数）の直接レベルに少なくともある程度、基づいてコンテキストを、適応して（adaptively）特定する。表現のモードがラ

10

20

30

40

50

ン・レベルである場合、現在のラン・レンジスに対して、符号器は、オーディオ入力シーケンス中のゼロ値係数の前のラン・レンジス、ゼロ値係数のパーセンテージに少なくともある程度基づいてそのコンテキストを、適応して (adaptively) 特定する。表現のモードがラン・レベルである場合、現在の係数に対して、符号器は、ゼロ値係数の現在のラン・レンジス、ゼロ値係数の前のラン・レンジス、および前の係数の直接レベルに少なくともある程度基づいて、そのコンテキストを、適応して (adaptively) 特定する。

#### 【 0 0 4 3 】

別の態様では、符号器または復号器が、直接符号化または直接復号化を使用し、顕著な値 (predominant value) (例えば、0) に等しい連続する係数のカウントを維持してオーディオ・データの第1の部分を符号化、または復号化する。カウントがしきい値を超えた場合、符号器または復号器は、ラン・レベル符号化またはラン・レベル復号化を使用してオーディオ・データの第2の部分を符号化、または復号化する。しきい値は、静的であること、または適応して (adaptively) 決定されることが可能である。しきい値は、係数のブロックのサイズに依存することが可能である。例えば、しきい値は、256の係数のブロックに対して4であり、あるいは512の係数のブロックに対して8であることが可能である。

#### 【 0 0 4 4 】

別の態様では、符号器または復号器が、第1の符号テーブルを使用してシーケンスの第1の部分を符号化、または復号化し、第2の符号テーブルを使用してシーケンスの第2の部分を符号化、または復号化する。第1のテーブルは、顕著な値 (例えば、0) に等しい連続する係数のより長いランの可能性が高い場合に使用され、第2のテーブルは、等しい値の連続する係数のより短いランの可能性が高い場合に使用される。使用されるテーブルは、シグナル・ビットによって示されることが可能である。

#### 【 0 0 4 5 】

適応エントロピー符号化テクニックおよび適応エントロピー復号化テクニックの特徴および利点は、添付の図面を参照して行われる様々な実施形態の以下の詳細な説明から明白となる。

#### 【 発明を実施するための最良の形態 】

#### 【 0 0 4 6 】

説明される実施形態では、オーディオ符号器は、いくつかの適応エントロピー符号化テクニックを実行する。ビット・レートを低減し、かつ/または品質を向上させる適応エントロピー符号化テクニックは、符号器のパフォーマンスを改良する。復号器は、対応するエントロピー復号化テクニックを行う。これらのテクニックは、単一の統合されたシステムの一環としてところどころで説明しているが、これらのテクニックは、別々に、場合により、他のテクニックと組み合わせて適用することも可能である。

#### 【 0 0 4 7 】

オーディオ符号器およびオーディオ復号器は、離散オーディオ信号を処理する。説明される実施形態では、オーディオ信号は、周波数変換されたオーディオ信号からの量子化された係数である。代替として、符号器および復号器は、別の種類の離散オーディオ信号、あるいはビデオまたは別の種類の情報を表わす離散信号を処理する。

#### 【 0 0 4 8 】

一部の実施形態では、オーディオ符号器は、直接信号レベル (direct signal levels) の符号化と、ラン・レンジスおよび信号レベルの符号化の間で、適応して切替えを行う。符号器は、スカラー・ハフマン符号化、ベクトル・ハフマン符号化、算術符号化、または別のテクニックを使用して直接信号レベルを符号化する。ラン・レンジス/レベル符号化 (ラン・レベル符号化とも呼ばれる) では、各ラン・レンジスが、ゼロ、またはより多くのゼロのランを表わし、各信号レベルが、ゼロでない値を表わす。ラン・レベル・イベント空間において、符号器は、ハフマン符号化、算術符号化、または別のテクニックを使用してそのイベント空間内でラン・レンジスおよびレベルを符号化する。復号器は、復号化中に、対応する適応切替えを行う。適応切替えは、ゼロ値レベルのしきい値数に達した際

10

20

30

40

50

に行われる。代替として、符号器および復号器は、さらなる基準または他の基準に基づいて切替えを行う。

【0049】

一部の実施形態では、オーディオ符号器は、可変次元ベクトル・ハフマン符号化を使用する。可変次元ベクトル・ハフマン符号化は、符号器が、シンボルのより確率の高い組合せをより大きい次元のベクトルを使用して表わし、シンボルのそれほど確率の高くない組合せをより小さい次元のベクトルまたはスカラーを使用して表わす、ハフマン符号を使用することができるようにする。復号器は、対応する可変次元のハフマン復号化を行う。

【0050】

一部の実施形態では、オーディオ符号器は、コンテキスト・ベースの算術符号化を使用する。符号器によって使用されるコンテキストは、オーディオ・データの個々の種類についての効率的な圧縮を可能にする。復号器は、対応するコンテキスト・ベースの算術復号化を行う。

【0051】

説明する実施形態では、オーディオ符号器およびオーディオ復号器は、様々なテクニックを実行する。これらのテクニックに関する動作は、通常、提示のため特定の順序で説明するが、この説明の仕方は、動作の順序のいくらかの並べ替えも包含するものと理解されたい。さらに簡明にするため、流れ図は、通常、特定のテクニックを他のテクニックと併せて使用することができる様々なやり方は示していない。

【0052】

I. コンピューティング環境

図1は、説明する実施形態を実装することが可能な適切なコンピューティング環境(100)の一般化された例を示している。コンピューティング環境(100)は、本発明の使用または機能の範囲に関して何ら限定を示唆するものではない。というのは、本発明は、多種多様な汎用または特殊目的のコンピューティング環境において実装できるからである。

【0053】

図1を参照すると、コンピューティング環境(100)は、少なくとも1つの処理ユニット(110)およびメモリ(120)を含む。図1で、この最も基本的な構成(130)が、破線内に含まれている。処理ユニット(110)は、コンピュータ実行可能命令を実行し、現実のプロセッサまたは仮想プロセッサであることが可能である。多重処理システムでは、複数の処理ユニットが、コンピュータ実行可能命令を実行して処理能力を高める。メモリ(120)は、揮発性メモリ(例えば、レジスタ、キャッシュ、RAM)、不揮発性メモリ(例えば、ROM、EEPROM、フラッシュ・メモリ等)、または揮発性メモリと不揮発性メモリの何らかの組合せであることが可能である。メモリ(120)は、オーディオ・データの適応エントロピー符号化/復号化を行うオーディオ符号器/復号器を実装するソフトウェア(180)を記憶する。

【0054】

コンピューティング環境は、さらなる特徴を有することが可能である。例えば、コンピューティング環境(100)は、ストレージ(140)と、1つまたは複数の入力デバイス(150)と、1つまたは複数の出力デバイス(160)と、1つまたは複数の通信接続(170)とを含む。バス、コントローラ、またはネットワークなどの相互接続機構(図示せず)が、コンピューティング環境(100)の構成要素を互いに接続する。通常、オペレーティング・システム・ソフトウェア(図示せず)は、コンピューティング環境(100)において実行されている他のソフトウェアのための動作環境を提供し、コンピューティング環境(100)の構成要素の活動を協調させる。

【0055】

ストレージ(140)は、取外し可能、または取外し不可能であることが可能であり、磁気ディスク、磁気テープまたは磁気カセット、CD-ROM、CD-RW、DVD、または情報を記憶するのに使用することができ、コンピューティング環境(100)内でア

10

20

30

40

50

クセスすることができる任意の他の媒体が含まれる。ストレージ(140)は、オーディオ・データの適応エントロピー符号化/復号化を行うオーディオ符号器/復号器を実装するソフトウェア(180)のための命令を記憶する。

#### 【0056】

入力デバイス(150)は、キーボード、マウス、ペン、またはトラックボールなどのタッチ入力デバイスであること、走査デバイス、ネットワーク・アダプタ、またはコンピューティング環境(100)に入力を提供する別のデバイスであることが可能である。オーディオの場合、入力デバイス(150)は、サウンド・カードであること、またはアナログ形態またはデジタル形態でオーディオ入力を受け入れる同様のデバイスであること、またはオーディオ・サンプルをコンピューティング環境に提供するCD-ROM読取り装置であることが可能である。出力デバイス(160)は、ディスプレイ、プリンタ、スピーカ、CD/DVD書き込み装置、ネットワーク・アダプタ、またはコンピューティング環境(100)から出力を提供する別のデバイスであることが可能である。

#### 【0057】

通信接続(170)は、通信媒体を介して別のコンピューティング・エンティティに対する通信を可能にする。通信媒体は、コンピュータ実行可能命令、圧縮されたオーディオ情報、または変調されたデータ信号中にその他のデータなどの情報を伝える。変調されたデータ信号とは、信号の中に情報を符号化するように特性の1つまたは複数が設定されている、または変更されている信号である。例として、限定としてではなく、通信媒体には、電気、光、RF、赤外線、音響、またはその他の搬送波を使用して実装される有線または無線のテクニックが含まれる。

#### 【0058】

本発明は、コンピュータ可読媒体の一般的な文脈で説明することができる。コンピュータ可読媒体は、コンピューティング環境内でアクセスすることが可能な任意の可用な媒体である。例として、限定としてではなく、コンピューティング環境(100)内で、コンピュータ可読媒体には、メモリ(120)、ストレージ(140)、通信媒体、および以上のいずれの組合せも含まれる。

#### 【0059】

本発明は、プログラム・モジュールの中に含まれ、コンピューティング環境において、目標の現実のプロセッサ上または仮想のプロセッサ上で実行されるコンピュータ実行可能命令の一般的な文脈で説明することができる。一般に、プログラム・モジュールには、特定のタスクを行う、または特定の抽象データ・タイプを実装するルーチン、プログラム、ライブラリ、オブジェクト、クラス、構成要素、データ構造等が含まれる。プログラム・モジュールの機能は、様々な実施形態において所望に応じて組み合わせること、またはプログラム・モジュールの間で分割することができる。プログラム・モジュールのためのコンピュータ実行可能命令は、ローカルのコンピューティング環境内で、または分散コンピューティング環境において実行することができる。

#### 【0060】

提示のため、詳細な説明では、コンピューティング環境におけるコンピュータ動作を説明するのに「解析する(analyze)」、「送る(send)」、「比較する(compare)」、および「調べる(check)」のような用語を使用する。これらの用語は、コンピュータによって行われる動作の高レベルの抽象化であり、人間によって行われる動作と混同してはならない。これらの用語に対応する実際のコンピュータ動作は、実装形態に応じて異なる。

#### 【0061】

##### II. 一般化されたオーディオ符号器およびオーディオ復号器

図2は、説明する実施形態を実装することが可能な一般化されたオーディオ符号器(200)を示すブロック図である。符号器(200)は、オーディオ・データの適応エントロピー符号化を行う。図3は、説明する実施形態を実装することが可能な一般化されたオーディオ復号器(300)を示すブロック図である。復号器(300)は、符号化されたオーディオ・データを復号化する。

## 【 0 0 6 2 】

符号器内部のモジュール間の関係、および復号器内部のモジュール間の関係は、例示的な符号器および例示的な復号器における情報の流れを示しており、他の関係は、簡明にするため、図示していない。実装形態に、および所望される圧縮のタイプに応じて、符号器または復号器のモジュールは、追加すること、省略すること、複数のモジュールに分割すること、他のモジュールと組み合わせること、および/または同様のモジュールで置き換えることができる。代替の実施形態では、異なるモジュールおよび/または他の構成を有する符号器または復号器が、オーディオ・データの適応エントロピー符号化、および適応エントロピー復号化を行う。

## 【 0 0 6 3 】

## A . 一般化されたオーディオ符号器

一般化されたオーディオ符号器 ( 2 0 0 ) は、選択器 ( 2 0 8 )、マルチ・チャネル・プリ・プロセッサ ( 2 1 0 )、区分器/タイル構成器 ( partitioner / tile configurer ) ( 2 2 0 )、周波数変換器 ( 2 3 0 )、知覚モデル化器 ( perception modeler ) ( 2 4 0 )、重み付け器 ( weighter ) ( 2 4 2 )、マルチ・チャネル変換器 ( 2 5 0 )、量子化器 ( 2 6 0 )、エントロピー符号器 ( 2 7 0 )、コントローラ ( 2 8 0 )、複合/純粋 ( mixed / pure ) ロスレス符号器 ( 2 7 2 ) および関連するエントロピー符号器 ( 2 7 4 )、ならびにビットストリーム・マルチプレクサ [ 「 M U X 」 ] ( 2 9 0 ) を含む。符号器 ( 2 0 0 ) のモジュールのいくつかについて以下に説明する。一部の実施形態における符号器 ( 2 0 0 ) のその他のモジュールに関する説明については、関連出願データのセクションで言及している出願を参照されたい。

## 【 0 0 6 4 】

符号器 ( 2 0 0 ) は、何らかのサンプリング深度およびサンプリング・レートで時系列の入力オーディオ・サンプル ( 2 0 5 ) をパルス符号変調された ( pulse code modulated ) [ 「 P C M 」 ] 形式で受け取る。入力オーディオサンプル ( 2 0 5 ) は、マルチ・チャネル・オーディオ (例えば、ステレオ・モード、サラウンド ( surround ) ) であること、またはモノラルであることが可能である。符号器 ( 2 0 0 ) は、オーディオ・サンプル ( 2 0 5 ) を圧縮し、符号器 ( 2 0 0 ) の様々なモジュールによって生成される情報を多重化して、Windows (登録商標) Media Audio [ 「 W M A 」 ] 形式または Advanced Streaming Format [ 「 A S F 」 ] などの形式でビットストリーム ( 2 9 5 ) を出力する。代替として、符号器 ( 2 0 0 ) は、他の入力形式および/または出力形式で機能する。

## 【 0 0 6 5 】

最初、選択器 ( 2 0 8 ) が、オーディオ・サンプル ( 2 0 5 ) に対して複数の符号化モードのなかから選択を行う。図 2 で、選択器 ( 2 0 8 ) は、2 つのモード、すなわち、複合/純粋ロスレス符号化モード ( mixed / pure lossless coding mode ) とロッシー符号化モード ( lossy coding mode ) の間で切替えを行う。ロスレス符号化モードは、複合/純粋ロスレス符号器 ( 2 7 2 ) を含み、通常、高品質 (および高いビットレート) の圧縮のために使用される。ロッシー符号化モードは、重み付け器 ( 2 4 2 ) や量子化器 ( 2 6 0 ) などの構成要素を含み、通常、調整可能な品質 (また規制されたビットレート) の圧縮のために使用される。選択器 ( 2 0 8 ) における選択決定は、ユーザ入力 (例えば、ユーザが、高品質のオーディオ・コピーを作成するためにロスレス符号化を選択すること) に、またはその他の基準に、依存する。その他の状況 (例えば、ロッシー圧縮が、十分な品質をもたらすことができない場合、またはビットを過剰に生じさせる場合) では、符号器 ( 2 0 0 ) は、フレームまたは 1 組のフレームに関してロッシー符号化から複合/純粋ロスレス符号化に切り換えることが可能である。

## 【 0 0 6 6 】

周波数変換器 ( 2 3 0 ) は、オーディオ・サンプル ( 2 0 5 ) を受け取り、オーディオ・サンプル ( 2 0 5 ) を周波数領域内のデータに変換する。周波数変換器 ( 2 3 0 ) は、周波数係数データのブロックを重み付け器 ( 2 4 2 ) に出力し、ブロックサイズなどの副

10

20

30

40

50

次情報をM U X ( 2 9 0 ) に出力する。周波数変換器 ( 2 3 0 ) は、周波数係数と副次情報をともに知覚モデル化器 ( 2 4 0 ) に出力する。

【 0 0 6 7 】

知覚モデル化器 ( 2 4 0 ) は、人間聴覚システムの特性をモデル化して、所与のビット・レートで再構成されるオーディオ信号の知覚される品質を向上させる。一般に、知覚モデル化器 ( 2 4 0 ) は、聴覚モデルに従ってオーディオ・データを処理した後、オーディオ・データに関する重み付け係数を生成するのに使用することができる重み付け器 ( 2 4 2 ) に情報を提供する。知覚モデル化器 ( 2 4 0 ) は、様々な聴覚モデルのいずれかを使用し、励起パターン情報、またはその他の情報を重み付け器 ( 2 4 2 ) に送る。

【 0 0 6 8 】

量子化バンド重み付け器として、重み付け器 ( 2 4 2 ) は、知覚モデル化器 ( 2 4 0 ) から受け取られた情報に基づいて量子化マトリックスのための重み付け係数を生成し、その重み付け係数を周波数変換器 ( 2 3 0 ) から受け取られたデータに適用する。重み付け器 ( 2 4 2 ) は、重み付け係数のセットなどの副次情報をM U X ( 2 9 0 ) に出力する。次に、チャンネル重み付け器として、重み付け器 ( 2 4 2 ) は、知覚モデル化器 ( 2 4 0 ) から受け取られた情報に基づき、またローカルで再構成された信号の品質にも基づいてチャンネル特有の重み付け係数を生成する。このスカラー重み ( scalar weight ) により、再構成されたチャンネルが、ほぼ様な品質を有することが可能になる。重み付け器 ( 2 4 2 ) は、係数データの重み付けされたブロックをマルチ・チャンネル変換器 ( 2 5 0 ) に出力し、チャンネル重み係数のセットなどの副次情報をM U X ( 2 9 0 ) に出力する。代替として、符号器 ( 2 0 0 ) は、別の形態の重み付けを使用するか、重み付けを省く。

【 0 0 6 9 】

マルチ・チャンネル・オーディオ・データの場合、重み付け器 ( 2 4 2 ) によって生成される雑音形状の周波数係数データについて複数のチャンネルは、しばしば、相関する。この相関を活用するため、マルチ・チャンネル変換器 ( 2 5 0 ) は、オーディオ・データにマルチ・チャンネル変換を適用することができる。マルチ・チャンネル変換器 ( 2 5 0 ) は、例えば、使用されたマルチ・チャンネル変換、およびフレームのマルチ・チャンネル変換された部分、を示す副次情報をM U X ( 2 9 0 ) に対して生成する。

【 0 0 7 0 】

量子化器 ( 2 6 0 ) は、マルチ・チャンネル変換器 ( 2 5 0 ) の出力を量子化し、エントロピー符号器 ( 2 7 0 ) への量子化された係数データを生成し、M U X ( 2 9 0 ) への量子化ステップ・サイズを含む副次情報を生成する。量子化により、情報の不可逆な損失が導入されるが、コントローラ ( 2 8 0 ) と連携して出力ビットストリーム ( 2 9 5 ) の品質およびビット・レートを符号器 ( 2 0 0 ) が調整することも可能になる。一部の実施形態では、量子化器 ( 2 6 0 ) は、適応型の、様な、スカラー量子化器である。代替の実施形態では、量子化器は、様なでない量子化器、ベクトル量子化器、および/または適応型でない量子化器であるか、または適応型の、様な、スカラー量子化の異なる形態を使用する。

【 0 0 7 1 】

エントロピー符号器 ( 2 7 0 ) は、量子化器 ( 2 6 0 ) から受け取られた量子化された係数データを損失なく圧縮する。一部の実施形態では、エントロピー符号器 ( 2 7 0 ) は、以下のセクションで説明するとおり、適応型のエントロピー符号化を使用する。エントロピー符号器 ( 2 7 0 ) は、オーディオ情報を符号化するのに費やされたビットの数を計算し、その情報を速度/品質コントローラ ( 2 8 0 ) に送ることができる。

【 0 0 7 2 】

コントローラ ( 2 8 0 ) は、量子化器 ( 2 6 0 ) と協働して、符号器 ( 2 0 0 ) の出力のビット・レートおよび/または品質を調整する。コントローラ ( 2 8 0 ) は、符号器 ( 2 0 0 ) の他のモジュールから情報を受け取り、受け取られた情報を処理して、現行の条件に与えられた所望の量子化係数を決定する。コントローラ ( 2 8 0 ) は、品質制約および/またはビット・レート制約を満たす目標を使用して、その量子化ファクタ ( quantiza

10

20

30

40

50

tion factors) を量子化器 (260) に出力する。

【0073】

複合ロスレス / 純粋ロスレス符号器 (272) および関連するエントロピー符号器 (274) が、複合 / 純粋ロスレス符号化モードの間オーディオ・データを圧縮する。符号器 (200) は、シーケンス全体に関して複合 / 純粋ロスレス符号化モードを使用するか、またはフレームごとに、またはその他の基準で、符号化モード間で切替えを行う。

【0074】

MUX (290) が、オーディオ符号器 (200) のその他のモジュールから受け取られた副次情報を、エントロピー符号器 (270) から受け取られたエントロピー符号化されたデータとともに多重化する。MUX (290) は、その情報を WMA 形式で、またはオーディオ復号器が認識する別の形式で出力する。MUX (290) は、符号器 (200) によって出力されるビットストリーム (295) を記憶する仮想バッファを含む。バッファの現在の充満度 (fullness)、バッファの充満度の変化の速度、およびバッファのその他の特性が、コントローラ (280) によって使用されて、個々の適用例に合わせて品質および / またはビット・レートが調整される (例えば、一定の品質 / 可変のビット・レートに、または一定のビット・レート / 可変の品質に、またはそれを下回るように) ことが可能である。

【0075】

B. 一般化されたオーディオ復号器

図3を参照すると、一般化されたオーディオ復号器 (300) が、ビットストリーム・デマルチプレクサ [「DEMUX」] (310) と、1つまたは複数の復号器 (320) と、複合 / 純粋ロスレス復号器 (322) と、タイル構成 (tile configuration) 復号器 (330) と、逆マルチ/チャンネル変換器 (340) と、逆量子化器 / 逆重み付け器 (350) と、逆周波数変換器 (360) と、重ね合わせ器 (overlapper) / 追加器 (adder) (370) と、マルチ・チャンネル・ポスト・プロセッサ (380) とを含む。復号器 (300) は、速度 / 品質制御または知覚モデル化のためのモジュールを含まないため、符号器 (300) よりいくぶん単純である。復号器 (300) のモジュールのいくつかについて以下に説明する。一部の実施形態における復号器 (300) のその他のモジュールに関する説明については、関連出願データのセクションにおいて言及している出願を参照されたい。

【0076】

復号器 (300) は、圧縮されたオーディオ情報のビットストリーム (305) を WMA 形式または別の形式で受け取る。ビットストリーム (305) は、エントロピー符号化されたデータ、ならびに復号器 (300) がオーディオ・サンプル (395) を再構成する元にする副次情報、を含む。

【0077】

DEMUX (310) が、ビットストリーム (305) 中の情報を構文解析し、情報を復号器 (300) の各モジュールに送る。DEMUX (310) は、オーディオの複雑さの変動、ネットワーク・ジッタ、および / またはその他のファクタ、に起因するビット・レートの短期の変動を補償するように1つまたは複数のバッファを含む。

【0078】

1つまたは複数のエントロピー復号器 (320) が、DEMUX (310) から受け取られたエントロピー符号を損失なく圧縮解除する。簡明にするため、1つのエントロピー復号器を図3に示しているが、異なるエントロピー復号器を、ロスレス符号化モードとロッシー符号化モードのために、または各モード内に、使用してもよい。また、簡明にするため、図3は、モード選択論理を示していない。エントロピー復号器 (320) は、通常、符号器 (200) において使用されたエントロピー符号化テクニックの逆を適用する。ロッシー符号化モードで圧縮されたデータを復号化する際、エントロピー復号器 (320) は、量子化された周波数係数データを生成する。

【0079】



複合／純粹ロスレス復号器（３２２）および関連するエントロピー復号器（３２０）が、複合／純粹ロスレス符号化モードに関して符号化されたオーディオ・データを損失なく圧縮解除する。復号器（３００）は、シーケンス全体に関して特定の復号化モードを使用するか、あるいはフレームごとに、またはその他の基準で復号化モードを切り換える。

【００８０】

逆マルチ・チャネル変換器（３４０）が、エントロピー復号器（３２０）からのエントロピー復号化された量子化された周波数係数データ、ならびに例えば、使用されたマルチ・チャネル変換、およびフレームの変換された部分を示すＤＥＭＵＸ（３１０）からの副次情報を受け取る。

【００８１】

逆量子化器／逆重み付け器（３５０）は、ＤＥＭＵＸ（３１０）から量子化係数ならびに量子化マトリックスを受け取り、逆マルチ・チャネル変換器（３４０）から量子化された周波数係数データを受け取る。逆量子化器／重み付け器（３５０）は、受け取られた量子化係数／マトリックス情報を必要に応じて圧縮解除し、次に、逆量子化および逆重み付けを行う。

【００８２】

逆周波数変換器（３６０）が、逆量子化器／逆重み付け器（３５０）によって出力された周波数係数データ、ならびにＤＥＭＵＸ（３１０）からの副次情報を受け取る。逆周波数変換器（３６０）は、符号器において使用された周波数変換の逆を適用し、ブロックを重ね合わせ器／追加器（３７０）に出力する。

【００８３】

重ね合わせ器／追加器（３７０）は、逆周波数変換器（３６０）および／または複合／純粹ロスレス復号器（３２２）から復号化された情報を受け取る。重ね合わせ器／追加器（overlapper/adder）（３７０）は、必要に応じてオーディオ・データを重ね合わせ、追加し、異なるモードで符号化されたオーディオ・データのフレームまたはその他のシーケンスをインターリーブする。

【００８４】

ＩＩＩ．適応エントロピー符号化／復号化モード切替え（Adaptive Entropy Encoding/Decoding Mode Switching）

入力シーケンスが単一の値（例えば、０）の多数の生起を含む場合、ラン・レベル符号化法が、しばしば、レベルの直接符号化よりも効果的である。ただし、ゼロでない量子化された変換係数がオーディオ・データ入力シーケンスにおいて、特に低い方の周波数で、一般的であるため、ラン・レベル符号化は、周波数範囲全体にわたって有効なわけではない。さらに、より高い品質のオーディオでは、ゼロでない量子化された変換係数が、高い方の周波数においてさえも、より一般的になる。（より高い品質のオーディオでは、量子化ステップ・サイズは、通常、より小さい。）したがって、一部の実施形態では、図２の符号器（２００）などの符号器が、オーディオ・データ入力シーケンスのある部分に関してラン・レベル符号化を使用し、そのシーケンスの別の部分に関してレベルの直接符号化を使用することができるマルチ・モード符号化テクニックを実施する。図３の復号器（３００）のような復号器が、対応するマルチ・モード復号化テクニックを実施する。

【００８５】

A．適応エントロピー符号化モード切替え

図４を参照すると、マルチ・モード符号化テクニック（technique；技術、方法）４００において、符号器は、まず、入力ストリーム中の信号レベルを直接に符号化する（４１０）。例えば、符号器は、信号レベルに対して直接に可変次元ハフマン符号化、コンテキスト・ベースの算術符号化、または別のエントロピー符号化テクニックを実施する。

【００８６】

符号化中の切替えポイントにおいて、符号器は、符号化スキームを切り換える（４２０）。符号器は、予め決められた切替えポイントにおいて符号化スキーム（scheme；仕組み、方式）を切り換えること、または入力データを解析して符号化スキームを切り換える適

10

20

30

40

50

切なポイントを決定することができる。例えば、符号器は、入力シーケンスを解析して、ラン・レベル符号化に切り換える最良のポイントを見つけ出し、その切替えポイントを出カビットストリームの中で復号器に送ることができる。あるいは、符号器は、入力データ中の連続するゼロ（または代替として、別の顕著な値）をカウントすることによって切替えポイントを、適応して計算し、連続するゼロの特定のしきい値数がカウントされた場合にラン・レベル符号化に切り換えることができる。復号器も同じやり方で切替えポイントを計算することができ、したがって、切替えポイントは、ビットストリームに含められる必要がない。あるいは、符号器および復号器は、何らかの他の基準を使用して切替えポイントを決定する。

【 0 0 8 7 】

10

切替えポイント後、符号器は、残りの信号レベルをラン・レベル符号化を使用して符号化する（ 4 3 0 ）。例えば、符号器は、ラン・レンジおよび信号レベルに対してハフマン符号化、コンテキスト・ベースの算術符号化、または別のエントロピー符号化テクニックを実施する。符号器は、切替えポイントの前と後で同一のテクニック（例えば、コンテキスト・ベースの算術符号化）を使用しても、または異なるテクニックを使用してもよい。

【 0 0 8 8 】

さらに、図 4、および適用例の様々な他の図は、単一の切替えポイントを示しているが、追加の切替えポイントを使用して入力データを 2 つより多くの部分に分割することもできる。例えば、連続するゼロのより高いしきい値に対して追加の適応切替えポイントを設定することができる。これにより、異なる符号化スキームを異なる部分に適用することができる。あるいは、符号器は、シーケンスにおける様々な分割ポイントを試して、異なる分割構成に関する符号化効率を、復号器にその異なる構成を通知する（signaling）費用を加え、評価することができる。

20

【 0 0 8 9 】

図 5 は、一実装形態による適応切替えポイント計算を伴うマルチ・モード符号化テクニック（ 5 0 0 ）を示している。適応切替えポイントは、連続するゼロ値係数のカウントに依存する。入力データは、最低周波数係数から最高周波数係数まで進む量子化された変換係数それぞれの信号レベルである。実際には、切替えポイントの位置は、圧縮されている信号、および符号化のビット・レート／品質に依存する。代替として、入力データは、別の形態および／または編成のオーディオ・データである。

30

【 0 0 9 0 】

最初に、符号器は、いくつかの変数を初期設定する。具体的には、符号器は、ラン・カウント変数を 0 に設定し（ 5 1 0 ）、符号化状態変数を「直接（direct）」に設定する（ 5 1 2 ）。

【 0 0 9 1 】

符号器は、次の係数 Q C を入力として受け取る（ 5 2 0 ）。次に、符号器は、その係数 Q C がゼロであるかどうかを調べる（ 5 3 0 ）。係数 Q C がゼロでない場合、符号器は、ラン・カウントをリセットする（ 5 3 8 ）。それ以外の場合（すなわち、係数 Q C がゼロである場合）、符号器は、ラン・カウント変数を増分し（ 5 3 2 ）、現在のラン・カウントがラン・カウントしきい値を超えているかどうかを調べる（ 5 3 4 ）。ラン・カウントしきい値は、静的であること、または係数のブロックのサイズなどのファクタに依存すること（例えば、 2 5 6 の係数のシーケンスに対して 4 のラン・カウントしきい値、 5 1 2 の係数のシーケンスに対して 8 のラン・カウントしきい値等）、または何らかの別の形で適応することが可能である。ラン・カウントがしきい値を超えた場合、符号器は、符号化状態をラン・レベル符号化(run-level encoding) [「 R L E 」] に切り換える（ 5 3 6 ）。

40

【 0 0 9 2 】

次に、符号器は、適切な場合、係数 Q C を符号化する（ 5 4 0 ）。（一部のケースでは、グループの係数が、ベクトル・ハフマン符号化などのテクニックを使用して一緒に符号

50

化される。そのようなケースでは、符号器は、係数  $Q_C$  を符号化することを延期することができる。) )

【 0 0 9 3 】

次に、符号器は、符号化モードを切り換えるべきかどうかを調べる ( 5 5 0 )。詳細には、符号器は、符号化状態を調べる。符号化状態がもはや直接ではない場合 (例えば、ゼロ係数がしきい値数に達した結果として、符号器が符号化状態を  $RLE$  に切り換えた場合)、符号器は、係数のラン・レベル符号化を開始する ( 5 6 0 )。(やはり、係数のグループと一緒に符号化されるケースでは、符号器は、一群の係数に関して好都合なブレイクポイントに達するまで、切替えの決定を延期することができる。)

【 0 0 9 4 】

符号器は、符号化モードを切り換えない場合、係数を符号化することを終えたかどうかを調べる ( 5 7 0 )。終えている場合、符号器は、終了する。終えていない場合、符号器は、次の係数を入力して ( 5 2 0 ) 符号化プロセスを続ける。

【 0 0 9 5 】

B . 適合エントロピー復号化モード切替え

図 6 を参照すると、マルチ・モード復号化テクニック ( 6 0 0 ) において、復号器は、符号化された信号レベルを直接に復号化する ( 6 1 0 )。例えば、復号器は、符号化された信号レベルに対して直接に、可変次元ハフマン復号化、コンテキスト・ベースの算術復号化、または別のエントロピー復号化テクニックを実施する。

【 0 0 9 6 】

復号化中に切替えポイントにおいて、復号器は、復号化スキームを切り換える ( 6 2 0 )。切替えポイントが予め決められている場合、復号器は、フラグまたは他の通知機構の形態で、復号化スキームをいつ切り換えるかを復号器に明示的に告げるデータを受け取ることができる。あるいは、復号器は、自らが受け取る入力データに基づいて復号化スキームをいつ切り換えるかを、適応して計算することができる。復号器は、切替えポイントを計算する場合、符号器によって使用されたのと同じ計算テクニックを使用して、復号化スキームが正しいポイントで切り換わることを確実にする。例えば、復号器は、連続するゼロ (または代替として、別の顕著な値) をカウントして切替えポイントを、適応して決定する。一実装形態では、復号器は、図 5 に示した符号器テクニックに対応する技術を使用する。あるいは、復号器は、何らかの他の基準を使用して切替えポイントを決定する。

【 0 0 9 7 】

切替えポイント後、復号器は、残りのラン・レベル符号化された信号レベルを復号化する ( 6 3 0 )。例えば、復号器は、符号化されたラン・レンジおよび信号レベルに対して、ハフマン復号化技術、コンテキスト・ベースの算術復号化技術、または別のエントロピー復号化テクニックを実施する。復号器は、切替えポイントの前と後で同じテクニック (例えば、コンテキスト・ベースの算術復号化) を使用しても、または異なるテクニックを使用してもよい。

【 0 0 9 8 】

I V . 可変次元ハフマン符号化および可変次元ハフマン復号化

直接信号レベルなどのシンボルを、スカラー・ハフマン符号化を使用して符号化することができるが、そのような手法は、シンボルを符号化するための最適なビット数が分数である場合、限界がある。また、スカラー・ハフマン符号化は、またスカラー・ハフマン符号がシンボル間の統計上の相関を考慮に入れることができないことによって、限界がある。ベクトル・ハフマン符号化は、スカラー・ハフマン符号化よりも良好なビット・レート低下をもたらす (例えば、符号器が、2 進ハフマン符号における確率を分数として活用できるようにすることにより)。また、一般に、より高い次元のベクトルは、より小さい次元のベクトルより良好なビット・レート低下をもたらす。ただし、符号 (code) がそれぞれ可能なシンボルの組合せに割り当てられる場合、ベクトル次元が増加するにつれてコードブック・サイズが指数関数的に増大する。例えば、3 2 ビットのシステムでは、4 次元ベクトルに関する可能な組合せの数は、 $(2^3)^4$  である。また、ベクトルを照合して

10

20

30

40

50

、ハフマン符号を見つけ出すのにかかる探索時間も、コードブック・サイズが増大するにつれて劇的に増加する。

【 0 0 9 9 】

一部の実施形態では、コードブック・サイズを小さくするため、図 2 の符号器 ( 2 0 0 ) のような符号器が、可変次元ベクトル・ハフマン符号化テクニックを使用する。コードブック符号 ( codebook code ) をそれぞれの可能な  $n$  次元の組合せに割り当ててのではなく、限られた数の最も確率の高い  $n$  次元ベクトルに符号が割り当てられる。特定の  $n$  次元ベクトルに符号 ( code ) が割り当てられない場合、 $n$  次元ベクトルは、代わりに、より小さい次元のベクトル ( 例えば、2 つの  $n / 2$  次元ベクトル ) として、ハフマン符号を有するスカラーとして、または離散値を表わすために何らかのテーブルのないテクニックを使用するスカラーとして、符号化される。図 3 の復号器 ( 3 0 0 ) のような復号器が、ベクトルに対する符号を見つけ出し、関連する値を見つけ出すことによってベクトルを再構成する。

10

【 0 1 0 0 】

例えば、シンボル当たり 2 5 6 の値が可能な 4 次元ベクトルの場合、符号器は、5 0 0 の最も確率の高い 4 次元ベクトルをハフマン符号を使用して符号化し、その他のベクトルを示すのにエスケープ符号を使用する。符号器は、そのような他のベクトルを 2 次元ベクトルに分割する。符号器は、5 0 0 の最も確率の高い 2 次元ベクトルをハフマン符号を使用して符号化し、その他のベクトルを示すのにエスケープ符号を使用し、このその他のベクトルは、分割されて、スカラー・ハフマン符号を使用して符号化される。したがって、符号器は、5 0 1 + 5 0 1 + 2 5 6 個の符号を使用する。

20

【 0 1 0 1 】

どのベクトルまたはどのスカラーがテーブル中のハフマン符号で表わされるかを決定することに関して、またハフマン符号自体をテーブルに割り当てることに関して、コードブック構成は、静的であること、以前に符号化されたデータに適応すること、または符号化されるべきデータに適応することが可能である。

【 0 1 0 2 】

A . 可変次元ベクトル・ハフマン符号化

図 7 を参照すると、符号器が、可変次元ベクトル・ハフマン ( variable-dimension vector Huffman ) [ 「 V D V H 」 ] 符号化テクニック ( 7 0 0 ) を使用する。例えば、符号器は、このテクニック ( 7 0 0 ) を使用して、オーディオ・データの周波数係数に関する信号レベルを直接に符号化する。代替として、符号器は、このテクニック ( 7 0 0 ) を使用して別の形態のオーディオ・データを符号化する。簡明にするため、図 7 は、コードブック構成を示していない。コードブック構成は、静的であること、以前に符号化されたデータに適応すること、または符号化されるべきデータに適応することが可能である。

30

【 0 1 0 3 】

符号器は、 $n$  個のシンボルの次のベクトルを獲得する ( 7 1 0 ) 。例えば、符号器は、次の 4 個のシンボルを順次に獲得する。

【 0 1 0 4 】

符号器は、コードブックがそのベクトルに対する符号を含むかどうかを調べる ( 7 2 0 ) 。含む場合、符号器は、単一のハフマン符号を使用して ( 7 3 0 ) そのベクトルを符号化する。例えば、 $n$  次元ベクトルをどのように符号化するかを決定するのに、符号器は、そのベクトルに関連する符号に関して  $n$  次元ベクトル符号テーブルを調べる。より大きい次元のベクトルは、通常、より大幅なビット・レート節約をもたらすため、符号器は、最も確率の高い  $n$  次元ベクトルに対してハフマン符号を使用する。ただし、テーブルのサイズを制限するため、 $n$  次元ベクトルの一部だけが、関連する符号を有する。

40

【 0 1 0 5 】

コードブックがベクトルに対する符号を含まない場合、符号器は、そのベクトルをより小さいベクトルおよび / またはスカラーに分割し ( 7 4 0 ) 、そのより小さいベクトルおよび / またはスカラーを符号化する。例えば、符号器は、 $n$  個のシンボルのベクトルを  $x$

50

個の  $n/x$  シンボル・ベクトルに分割する。各  $n/x$  シンボル・ベクトルに関して、符号器は、符号化テクニックを再帰的に繰り返し、 $n/x$  シンボル・ベクトル、または  $n/x$  シンボル・ベクトルを構成するベクトル/スカラーが、ハフマン符号で、または（スカラーに関して）離散値を表わすために、テーブルのないテクニックを使用して符号化された時点で終了する。

#### 【0106】

次に、符号器は、符号化されるべきさらなるベクトルが存在するかどうかを調べる（750）。存在しない場合、符号器は、終了する。存在する場合、符号器は、 $n$  個のシンボルの次のベクトルを獲得する（710）。

#### 【0107】

##### 1. 例としての実装形態

図8は、一実装形態におけるVDVH符号化を使用してベクトルを符号化するための詳細なテクニック（800）を示している。このテクニック（800）では、符号器は、複数シンボルから成るベクトル中のシンボルの整数値を合計して、そのベクトルを単一のハフマン符号を使用して符号化するか、またはそのベクトルをより小さいベクトル/スカラーに分割するかを決定する。これにより、コードブック・サイズが実質的に制限され、符号の探索が迅速になる。

#### 【0108】

$n$  次元「 $n$ -dim」ベクトルに対するコードブック・テーブルは、 $L_1$  個の  $n$  次元ベクトルに対するハフマン符号を含む。また、コードブック・テーブルは、エスケープ符号も含む。 $L_1$  個の符号は、ベクトル成分（整数である）の合計が特定のしきい値  $T_1$  を下回る各ベクトルに対するものである。例えば、 $n$  が4であり、4次元ベクトルに関するしきい値  $T_1$  が6であるものと想定する。4次元ベクトルに関するそのコードブック・テーブルは、エスケープ符号および126個の符号を含み、それぞれの可能なベクトルの成分（例えば、成分の絶対値）の合計が6より小さい  $-(0, 0, 0, 0)$ 、 $(0, 0, 0, 1)$  などである。ベクトルの成分合計に基づいてテーブルサイズを制限することは、一般に、最も確率の高いベクトルがより小さい成分合計を有するベクトルであるため、有効である。

#### 【0109】

$n$  次元ベクトルに対するコードブック・テーブルが、特定の  $n$  次元ベクトルに対するハフマン符号を有さない場合、符号器は、出力ビットストリームにエスケープ符号を追加し、他のコードブック・テーブルの中でより小さい次元の複数のベクトルまたは複数のスカラーをルックアップして、その  $n$  次元ベクトルをより小さい次元の複数のベクトルまたは複数のスカラーとして符号化する。例えば、より小さい次元は、 $n/2$  が1でない限り、 $n/2$  であり、 $n/2$  が1である場合、 $n$  次元ベクトルは、複数のスカラーに分割される。代替として、 $n$  次元ベクトルは、何らかの別の仕方でも分割される。

#### 【0110】

より小さい次元のベクトルに対するコードブック・テーブルは、 $L_2$  個のより小さい次元のベクトルに対するハフマン符号、ならびにエスケープ符号を含む。 $L_2$  個の符号は、より小さい次元のテーブルに対してベクトル成分の合計が特定のしきい値  $T_2$  を下回る各ベクトルに対する符号である。例えば、より小さい次元が2であり、2次元ベクトルに関するしきい値  $T_2$  が16であるものと想定する。2次元ベクトルに対するコードブック・テーブルは、エスケープ符号、および成分（例えば、成分の絶対値）の合計が16を下回るそれぞれの可能なベクトルごとに1つの136個の符号、 $-(0, 0)$ 、 $(0, 1)$ 、などを含む。

#### 【0111】

より小さい次元のベクトルに対するコードブック・テーブルが特定のより小さい次元のベクトルに対するハフマン符号を有さない場合、符号器は、エスケープ符号を出力ビットストリームに追加し、他のコードブック・テーブルを使用してそのベクトルをさらに小さい次元のベクトルまたはスカラーとして符号化する。このプロセスは、スカラー・レベル

10

20

30

40

50

に至るまで繰り返される。例えば、分割は、スカラー・レベルに至るまで2の累乗で行われる。代替として、ベクトルは、何らかの別の仕方分割される。

#### 【0112】

スカラー・レベルで、コードブック・テーブルは、 $L_3$  個のスカラー、ならびにエスケープ符号を含む。 $L_3$  個の符号は、特定のしきい値  $T_3$  (小さい値の確率が高いと想定する) を下回る各スカラーに対するものである。例えば、スカラーに関するしきい値  $T_3$  が 100 であるものと想定する。スカラーに関するコードブック・テーブルは、100 個の符号、およびエスケープ符号を含む。スカラーは、スカラー符号テーブルの中で関連する符号を有さない場合、エスケープ符号、およびテーブルなしのテクニックによる値 (例えば、リテラル (literal)) を使用して符号化される。このセクションで与えた数値的な例のすべてを使用すると、テーブルは、合計で  $126 + 1 + 136 + 1 + 100 + 1 = 365$  個の符号を含む。

10

#### 【0113】

テーブルの次元サイズ、ベクトル分割ファクタ、ベクトル成分合計に関するしきい値は、実装形態に依存する。その他の実装形態は、異なるベクトル・サイズ、異なる分割係数、および/または異なるしきい値を使用する。代替として、符号器は、VDVH符号化を行う際にベクトル成分合計以外の基準を使用してベクトル・サイズ/コードブック・テーブル切替えを行う。

#### 【0114】

図8を参照すると、符号器はまず、 $n$  次元ベクトルを獲得する (810)。 $n$  次元ベクトルは、各シンボルが、例えば、オーディオ・データの周波数係数に関する量子化されたレベルを表わす値を有する  $n$  個のシンボルを含む。

20

#### 【0115】

符号器は、ベクトル成分を合計して (812)、その合計を  $n$  次元ベクトルに関するしきい値と比較する (820)。合計がしきい値より小さいか、またはしきい値に等しい場合、符号器は、符号テーブルからのハフマン符号を使用してその  $n$  次元ベクトルを符号化し (822)、符号化が完了するまで継続する (824)。合計がしきい値より大きい場合、符号器は、エスケープ符号を送り (826)、その  $n$  次元ベクトルを  $n/2$  の次元を有する2つのより小さいベクトルに分割する (830)。

#### 【0116】

30

符号器は、次の  $n/2$  次元ベクトルを獲得し (840)、その  $n/2$  次元ベクトルの成分を合計する (842)。符号器は、 $n/2$  次元ベクトルに関連するしきい値に対してその合計をチェックする (850)。合計がしきい値より小さいか、またはしきい値に等しい場合、符号器は、 $n/2$  次元ベクトルに関する符号テーブルからのハフマン符号を使用してその  $n/2$  次元ベクトルを符号化し (852)、符号器が  $n/2$  次元ベクトルを符号化することを終えていない場合 (854)、次の  $n/2$  次元ベクトルを獲得する (840)。合計が  $n/2$  次元ベクトルに関するしきい値より大きい場合、符号器は、別のエスケープ符号を送る (856)。

#### 【0117】

符号器は、一般に、ベクトルを処理する際にこのパターンに従い、各ベクトルを符号化するか、またはそのベクトルをより小さい次元のベクトルに分割する。符号器がベクトルを2つのスカラー (1次元) 成分に分割する場合 (860)、符号器は、次のスカラーを獲得し (870)、そのスカラーの値をスカラー値に関連するしきい値と比較する (880)。スカラー値がしきい値より小さいか、しきい値に等しい場合 (880)、符号器は、スカラーに関する符号テーブル (882) からのハフマン符号を使用してスカラーを符号化する。スカラー値がしきい値より大きい場合、符号器は、テーブルのないテクニックを使用してそのスカラーを符号化する (884)。次に、符号器は、スカラーを処理するのを終えていない場合 (886)、次のスカラーを獲得する (870)。

40

#### 【0118】

代替として、符号器は、VDVH符号化を行う際に、異なる次元サイズを有するテーブ

50

ルを使用し、ベクトルを2の累乗による以外の何らかの形で分割し、かつ/またはベクトル成分合計以外の基準を使用してベクトル・サイズ/コードブック・テーブルを切り換える。

【0119】

## 2. 適応切替え

図9は、符号器が、ラン・レンジスおよび信号レベルの符号化に切り換えるための切替えポイントを一実装形態に従って適応して決定する態様で、直接信号レベルの係数のV D V H符号化をするテクニック(900)を示している。適応切替えポイントは、連続するゼロ値係数のカウントに依存する。入力データは、最低周波数係数から最高周波数係数まで進む量子化された変換係数に関する信号レベルである。代替として、入力データは、別の形態および/または編成のオーディオ・データである。

10

【0120】

最初に、符号器は、いくつかの変数を初期設定する。具体的には、符号器は、ラン・カウント変数を0に設定し(910)、現在のベクトル変数を空(empty)に設定し(912)、符号化状態変数を直接可変次元ベクトル・ハフマン[「D V D V H」]に設定する(914)。

【0121】

符号器は、次の係数Q Cを入力として受け取る(920)。次に、符号器は、その係数がゼロであるかどうかを調べる(930)。係数Q Cがゼロでない場合、符号器は、ラン・カウントをリセットし(938)、その係数Q Cを現在のベクトルに追加する(940)。それ以外の場合(すなわち、係数Q Cがゼロである場合)、符号器は、ラン・カウント変数を増分し(932)、現在のラン・カウントがラン・カウントしきい値を超えているかどうかを調べる(934)。ラン・カウントしきい値は、静的であること、または係数のブロックのサイズ(例えば、256個の係数の入力シーケンス中の4つのゼロ)などのファクタに依存すること、または何らかの他の形で適応することが可能である。例えば、しきい値は、入力シーケンス中の係数に関連して、または関連せずに増分、または減分することができる。ラン・カウントがしきい値を超えた場合、符号器は、符号化状態をラン・レベル符号化[「R L E」]に切り換え(936)、係数Q Cは、現在のベクトルの成分として追加される(940)。

20

【0122】

係数Q Cを現在のベクトルに追加することにより、ベクトルの次元が増分される。符号器は、現在のベクトル中の成分の数を現在のベクトルの最大次元と比較することにより、現在のベクトルが符号化される用意ができていないかどうかを判定する(950)。用意ができていない場合、符号器は、D V D V H符号化を使用して現在のベクトルを符号化する(960)。現在のベクトルが最大次元より小さいが、係数Q Cがシーケンスの最後である場合、符号器は、現在のベクトルに埋込みを行い、D V D V H符号化を使用して現在のベクトルを符号化する(960)。最大次元は、実装形態に依存する。一実装形態では、最大次元は、8である。ただし、最大次元は、例えば、コードブックを作成する、記憶する、または伝送するのに利用可能なリソースの量に応じて、大きくする、または小さくすることができる。

30

40

【0123】

ベクトルを符号化した後、符号器は、符号化状態を調べる(970)。符号化状態がもはやD V D V Hではない場合(例えば、ゼロ係数のしきい値数を超えた結果、符号器が、符号化状態をR L Eに切り換えている場合)、符号器は、ラン・レンジスおよびレベルとして係数の符号化を開始する(980)。ラン・レベル符号化は、例えば、ハフマン符号化、ベクトル・ハフマン符号化、またはコンテキスト・ベースの算術符号化を含め、いくつかの仕方で行うことができる。一部の実施形態では、ラン・レベル符号化は、1つのテーブルが、より短いランの可能性が高いデータを符号化するのに使用され、もう1つのテーブルが、より長いランの可能性が高いデータを符号化するために使用される2つのハフマン符号テーブルを有するハフマン符号化を使用して行われる。符号器は、各テーブルを

50

試みて、どちらのテーブルを符号器が使用したかを示すシグナル・ビットと共に、ひとつのテーブルからの符号を選択する。

【 0 1 2 4 】

符号化状態が切り換わっていない、または現在のベクトルが、符号化の用意ができていない場合、符号器は、符号化されるべきさらなる係数が存在するかどうかを判定する ( 9 9 0 )。存在する場合、符号器は、次の係数を入力し ( 9 2 0 )、符号化プロセスを続ける。

【 0 1 2 5 】

B . 可変次元ベクトル・ハフマン復号化

図 1 0 は、図 7 に示した V D V H 符号化テクニック ( 7 0 0 ) に対応する V D V H 復号化テクニック ( 1 0 0 0 ) を示している。例えば、復号器は、このテクニック ( 1 0 0 0 ) を使用して、オーディオ・データの周波数係数に関する符号化された信号レベルを直接に復号化する。代替として、復号器は、このテクニックを使用して別の形態のオーディオ・データを復号化する。

【 0 1 2 6 】

復号器は、n 次元ベクトル・ハフマン符号テーブルに関する次のハフマン符号を獲得する ( 1 0 1 0 )。例えば、復号器は、シーケンス内の 4 つのシンボルの対する次のハフマン符号を獲得する。

【 0 1 2 7 】

復号器は、ハフマン符号が n 次元ベクトル・ハフマン符号テーブルに関するエスケープ符号であるかどうかを調べる ( 1 0 2 0 )。エスケープ符号ではない場合、復号器は、そのハフマン符号によって表わされる n 個のシンボルを獲得する ( 1 0 3 0 )。例えば、復号器は、4 次元ベクトル・ハフマン・コードブックの中でそのハフマン符号に関連する 4 つのシンボルを獲得する。

【 0 1 2 8 】

符号がエスケープ符号である場合、n 次元のコードブックはそのベクトルに対する符号を含まず、復号器は、より小さいベクトルおよび / またはスカラーに対するハフマン符号を獲得する ( 1 0 4 0 )。例えば、復号器は、x 個の n / x シンボル・ベクトルに対する符号を獲得する。各 n / x シンボル・ベクトルに関して、復号器は、復号化テクニックを再帰的に繰り返し、n / x シンボル・ベクトル、または n / x シンボル・ベクトルを構成するベクトル / スカラーが復号化された時点で終了する。

【 0 1 2 9 】

次に、復号器は、復号化されるべき n 次元ベクトル・ハフマン符号テーブルに関するさらなる符号が存在するかどうかを調べる ( 1 0 5 0 )。存在しない場合、復号器は終了する。存在する場合、復号器は、次のそのようなハフマン符号を獲得する ( 1 0 1 0 )。

【 0 1 3 0 】

1 . 例示的な実装形態

図 1 1 は、一実装形態における V D V H 符号化を使用して符号化されたベクトルを復号化するための詳細なテクニック ( 1 1 0 0 ) を示している。この復号化テクニック ( 1 1 0 0 ) は、図 8 に示した符号化テクニック ( 8 0 0 ) に対応する。

【 0 1 3 1 】

図 1 1 を参照すると、復号器は、n 次元ベクトル・ハフマン符号テーブルに関する次の符号を獲得する ( 1 1 1 0 )。復号器は、その符号が、n 次元ベクトル・ハフマン符号テーブルに関するエスケープ符号であるかどうかを調べる ( 1 1 2 0 )。エスケープ符号ではない場合、復号器は、n 次元ベクトルテーブルの中でその符号によって表わされる n 個のシンボルを獲得する ( 1 1 2 2 )。復号器は、符号化されたデータを処理することを終えるまで継続する ( 1 1 2 4 )。

【 0 1 3 2 】

符号が n 次元ベクトル・ハフマン符号テーブルに関するエスケープ符号である場合、復号器は、n / 2 次元ベクトル・ハフマン符号テーブルを使用して、その n 次元ベクトルを

10

20

30

40

50



2つの $n/2$ 次元ベクトルとして復号化する。具体的には、復号器は、 $n/2$ 次元ベクトル・ハフマン符号テーブルに関する次の符号を獲得する(1130)。復号器は、符号が $n/2$ 次元ベクトル・ハフマン符号テーブルに関するエスケープ符号であるかどうかを調べる(1140)。エスケープ符号ではない場合、復号器は、 $n/2$ 次元ベクトル・ハフマン符号テーブルの中でその符号によって表わされる $n/2$ 個のシンボルを獲得する(1142)。復号器は、そのような符号の処理が完了するまで、 $n/2$ 次元ベクトル・ハフマン符号テーブルに関する符号を処理することを続ける(1144)。

【0133】

符号が $n/2$ 次元ベクトル・ハフマン符号テーブルに関するエスケープ符号である場合、復号器は、その $n/2$ 次元ベクトルを2つの $n/4$ 次元ベクトルとして復号化する。この際、この2つの $n/4$ 次元ベクトルは、スカラーであることが可能であるといった具合である。

10

【0134】

復号器は、一般に、エスケープ符号が検出された場合、復号化されるべきベクトルがスカラー(1次元ベクトル)になるまで、より大きい次元のベクトルを2つのより小さい次元のベクトルとして復号化するこのパターンに従う。スカラーになった時点で、復号器は、スカラー・ハフマン符号テーブルに関する次の符号を獲得する(1150)。復号器は、その符号がスカラー・ハフマン符号テーブルに関するエスケープ符号であるかどうかを調べる(1160)。エスケープ符号ではない場合、復号器は、スカラー・ハフマン符号テーブルの中でその符号によって表されるスカラーを獲得する(1162)。復号器は、そのような符号の処理が完了するまで、スカラーに対する符号を処理することを続ける(1164)。符号がスカラー・ハフマン符号テーブルに関するエスケープ符号であった場合、そのスカラーは、テーブルのないテクニックを使用して符号化され、復号器は、その値を獲得する(1170)。

20

【0135】

代替として、復号器は、VDVH復号化を行う際、異なる次元サイズを有するテーブルを使用し、かつ/または2の累乗による以外の何らかの別の形でベクトルを分割する。

【0136】

## 2. 適応切替え

図12は、復号器が、ラン・レンジスおよび信号レベルの復号化に切り換えるための切替えポイントを適応して決定する態様で、VDVH符号化を使用して符号化されたベクトルを、復号化するための一実装形態によるテクニック(1200)を示している。適応切替えポイントは、最低周波数係数から最高周波数係数まで進む量子化された変換係数に関する信号レベルであるデータ中の連続するゼロ値係数のカウンタに依存する。代替として、データは、別の形態および/または編成のオーディオ・データである。

30

【0137】

最初に、復号器は、いくつかの変数を初期設定する。具体的には、復号器は、ラン・カウントを0に設定し(1210)、復号化状態をDDVHに設定する(1212)。

【0138】

復号器は、次のベクトルを、ハフマン符号テーブルの中でそのベクトルに対する符号をルックアップすることによって復号化する(1220)。例えば、復号器は、図11に示した復号化テクニック(1100)を行う。次に、復号器は、復号化されたベクトルに基づいてラン・カウントを更新する(1230)(具体的には、復号化されたベクトルにおけるゼロ値の数を使用して、ラン・カウントをリセットする、増分する、または別の仕方で調整する)。

40

【0139】

復号器は、ラン・カウントがしきい値を超えているかどうかを調べる(1240)。ラン・カウントしきい値は、静的であること、または係数のブロックのサイズ(例えば、256個の係数の入力シーケンス中の4つのゼロ)などのファクタに依存すること、または何らかの別の形で適応することが可能である。ラン・カウントがしきい値を超えた場合、

50

復号器は、ラン・レベル復号化を使用して符号化されている係数を復号化することを開始する(1250)。ラン・レベル復号化は、例えば、ハフマン復号化、ベクトル・ハフマン復号化、またはコンテキスト・ベースの算術復号化を含むいくつかのやり方で行うことができる。

#### 【0140】

一部の実施形態では、ラン・レベル復号化は、1つのテーブルが、より短いランの可能性が高いデータを復号化するために使用され、もう1つのテーブルが、より長いランの可能性が高いデータを復号化するために使用される、2つの可能なハフマン符号テーブルを有するハフマン復号化を使用して行われる。復号器が符号を受け取った際、その符号中のシグナル・ビットは、どちらのテーブルを復号器が使用したかを示し、復号器は、適切なテーブルの中でその符号をルックアップする。

10

#### 【0141】

ラン・カウントがしきい値を超えていない場合、復号器は、復号化が終了するまでベクトルを処理すること続ける(1260)。

#### 【0142】

##### V. コンテキスト・ベースの算術符号化およびコンテキスト・ベースの算術復号化

一部の実施形態では、図2の符号器(200)のような符号器が、コンテキスト・ベースの算術(context-based arithmetic) [「CBA」] 符号化を使用してオーディオ・データのシーケンスを符号化する。CBA符号化では、入力シンボルに関する異なる確率分布が、異なるコンテキストに関連付けられる。入力シーケンスを符号化するのに使用される確率分布は、コンテキストが変化した際に変化する。コンテキストは、入力シーケンスの中で出現する特定の入力シンボルの確率に影響を及ぼすと予期される様々なファクタを測定することによって計算することができる。図3の復号器(300)のような復号器が、対応する算術復号化(arithmetic decoding)を行う。

20

#### 【0143】

係数を直接に(すなわち、直接レベルとして)符号化する際、符号器は、シーケンス中の前の係数の値を含むファクタを使用してコンテキストを計算する。ラン・レベル符号化を使用して係数を符号化する際、符号器は、前の係数の値に加え、現在のランの長さおよび前のランの長さを含むファクタを使用してコンテキストを計算する。符号器は、計算されたコンテキストに関連する確率分布を使用して、そのデータに適切な算術符号を決定する。したがって、コンテキストを計算する際に様々なファクタを使用することにより、符号器は、データに対して、またそのデータの表現のモード(すなわち、直接、ランレベル)に対して適応するようにコンテキストを決定することができる。

30

#### 【0144】

代替の実施形態では、符号器は、追加のファクタを使用すること、いくつかのファクタを省くこと、または前述したファクタを他の組合せでを使用することができる。

#### 【0145】

##### A. コンテキストの例示的な実装形態

表2~5および図13A~13D、14A~14H、および15A~15Hは、例示的な実装形態において、それぞれ、CBA符号化を行う際、およびCBA復号化を行う際に使用されるコンテキストおよび確率分布を示している。代替として、CBA符号化およびCBA復号化は、異なるコンテキストおよび/または異なる確率分布を使用する。

40

#### 【0146】

以下の説明は、例示的な実装形態における符号器におけるコンテキスト計算を中心的に扱うが、復号器は、以前に復号化されたオーディオ・データを使用して復号化中に対応するコンテキスト計算を行う。

#### 【0147】

前述したとおり、符号器は、直接レベルだけを符号化していても、またはラン・レングスおよび直接レベルを符号化していても、CBA符号化を使用して係数を符号化することができる。ただし、一実施形態では、コンテキストを計算するためのテクニックは、符号

50

器が直接レベルだけを符号化しているか、またはラン・レンジおよび直接レベルを符号化しているかに応じて異なる。さらに、ラン・レンジおよび直接レベルを符号化する際、符号器は、ラン・レンジを符号化しているか、または直接レベルを符号化しているかに応じて異なるコンテキストを使用する。

#### 【 0 1 4 8 】

符号器は、因果（関係を示す）コンテキスト（causal context）を使用する直接レベルの算術符号化中に、4 コンテキスト・システムを、コンテキストを計算するために使用する。符号器は、前のレベル（ $L[n-1]$ ）の値、および前のレベルの直前のレベル（ $L[n-2]$ ）に基づいて現在のレベル  $L[n]$  に関するコンテキストを計算する。このコンテキスト計算は、

10

1) 前のレベルが低である場合、現在のレベルは、低である可能性が高いという想定、および

2) 前の2つのレベルは、その他のレベルより良好な現在のレベルの予測因子（predictor）である可能性が高いという想定、

に基づいている。表2は、4 コンテキスト・システムにおける前の2つのレベルの値に関連するコンテキストを示している。図13A～13Dは、それらのコンテキストに関する現在のレベルに関する確率分布を示している。

#### 【 0 1 4 9 】

##### 【表2】

表2：直接レベルのCBA符号化／復号化のためのコンテキスト

20

$L[n-1]$	$L[n-2]$	コンテキスト
= 0	= 0	0
= 0	$\geq 1$	1
= 1	不定 (any)	2
$\geq 2$	不定	3

#### 【 0 1 5 0 】

図13A～13Dの確率分布は、前の2つのレベルがゼロか、またはゼロに近い場合、現在のレベルは、ゼロであるか、またはゼロに近い可能性がより高い、という傾向を示している。

30

#### 【 0 1 5 1 】

また、符号器は、レベルのラン・レンジ符号化を行う際にもCBA符号化を使用することができる。ラン・レンジを符号化する際、コンテキストを計算するために符号器によって使用されるファクタには、入力シーケンス中のゼロのパーセンテージ（シーケンスの一部、またはすべてにわたるその時点までの合計（running total））、およびゼロの前のランの長さ（ $R[n-1]$ ）が含まれる。符号器は、表3で以下に示すとおり、入力シーケンス中のゼロのパーセンテージに基づいてゼロ・パーセンテージ指標を計算する。

#### 【 0 1 5 2 】

##### 【表3】

40

表3：ラン・レンジのCBA符号化／復号化に関するゼロ・パーセンテージ指標

ゼロ%	ゼロ%指標
$\geq 90$	0
$\geq 80$	1
$\geq 60$	2
< 60	3

#### 【 0 1 5 3 】

符号器は、ゼロ・パーセンテージ指標を前のランの長さとともに使用して、表4に示す

50

とおり、現在のラン・レングスを符号化するためのコンテキストを計算する。図 1 4 A ~ 1 4 H は、それらのコンテキストに関連するラン・レングス値に関する確率分布を示している。

【 0 1 5 4 】

【表 4】

表 4 : ラン・レングスの C B A 符号化 / 復号化のためのコンテキスト

ゼロ % 指標	R [n-1]	コンテキスト
0	= 0	0
0	> 0	4
1	= 0	1
1	> 0	5
2	= 0	2
2	> 0	6
3	= 0	3
3	> 0	7

10

【 0 1 5 5 】

例えば、レベルの 9 1 % がゼロである ( 0 というゼロ・パーセンテージ指標をもたらす ) 入力シーケンスでは、ゼロの前のランの長さが 1 5 であった場合、コンテキストは、4 である。図 1 4 A ~ 1 4 H の確率分布は、入力シーケンス中のゼロのパーセンテージがより高い場合、より長いラン・レングスの可能性が高いことを示している。また、その確率分布は、所与のゼロ・パーセンテージ指標の範囲内で、ゼロのラン・レングスに続くラン・レングスは、ゼロより大きいラン・レングスに続くラン・レングスより短い可能性が高い、という傾向を示している。

20

【 0 1 5 6 】

ラン・レベル・データ中のレベルを符号化する際、コンテキストを計算するために符号器によって使用されるファクタには、現在のランの長さ ( R [ n ] )、前のランの長さ ( R [ n - 1 ] )、および前の 2 つのレベル ( L [ n - 1 ] および L [ n - 2 ] ) の値が含まれる。このコンテキスト計算は、現在のレベルは、レベル間の間隔 ( すなわち、ランレングス ) が大き過ぎない限り、2 つの前のレベルに依存するという所見に基づいている。また、前のレベルが低い場合、および前のランが短い場合、現在のレベルは、低である可能性がより高い。前のランが長い場合、前のレベルは、現在のレベルに対してそれほど影響を有さない。

30

【 0 1 5 7 】

現在のラン・レングスの値、前のラン・レングスの値、および前の 2 つのレベルの値に関連付けられたコンテキストを以下に表 5 で示している。図 1 5 A ~ 1 5 H は、それらのコンテキストに関連するレベルに関する確率分布を示している。

【 0 1 5 8 】

【表 5】

表 5 : ラン・レベル符号化におけるレベルの C B A 符号化 / 復号化に関するコンテキスト

R [n]	R [n-1]	L [n-1]	L [n-2]	コンテキスト
$\geq 2$	不定	不定	不定	0
< 2	$\geq 2$	= 1	不定	1
< 2	$\geq 2$	= 2	不定	2
< 2	$\geq 2$	> 2	不定	3
< 2	< 2	= 1	= 1	4
< 2	< 2	= 1	> 1	5
< 2	< 2	= 2	不定	6
< 2	< 2	> 2	不定	7

40

50

## 【 0 1 5 9 】

例えば、ゼロの現在のランの長さが1であり、ゼロの前のランの長さが2であり、前のレベルが1である入力シーケンスでは、コンテキストは、1である。図15A～15Hの確率分布は、前のレベルが低い場合、および現在のラン・レンジおよび前のラン・レンジが短い場合、現在のレベルは、ゼロであるか、またはゼロに近い可能性がより高い。

## 【 0 1 6 0 】

## B．適応切替え

図16は、符号器がラン・レンジおよび信号レベルの符号化に切り換えるための切替えポイントを、適応して決定する態様で、直接信号レベルの係数のCBA符号化のための一実装形態によるテクニック(1600)を示している。適応切替えポイントは、連続するゼロ値係数のカウントに依存する。入力データは、最低周波数係数から最高周波数係数まで進む量子化された変換係数に関する信号レベルである。代替として、入力データは、別の形態および/または編成のオーディオ・データである。

10

## 【 0 1 6 1 】

最初に、符号器は、いくつかの変数を初期設定する。具体的には、符号器は、ラン・カウント変数を0に設定し(1610)、符号化状態変数を直接コンテキスト・ベースの算術(direct context-based arithmetic)(DCBA)に設定する(1612)。

## 【 0 1 6 2 】

符号器は、次の係数QCを入力として受け取る(1620)。次に、符号器は、その係数がゼロであるかどうかを調べる(1630)。係数QCがゼロでない場合、符号器は、ラン・カウントをリセットし(1638)、DCBA符号化を使用してその係数を符号化する(1640)。

20

## 【 0 1 6 3 】

それ以外の場合(すなわち、係数QCがゼロである場合)、符号器は、ラン・カウント変数を増分し(1632)、現在のラン・カウントがラン・カウントしきい値を超えているかどうかを調べる(1634)。ラン・カウントしきい値は、静的であること、または係数のブロックのサイズ(例えば、256個の係数の入力シーケンス中の4つのゼロ)などのファクタに依存すること、または何らかの別の形で適応することが可能である。例えば、しきい値は、入力シーケンス中の係数の数に関連して、または関連せずに上げること、または下げることができる。ラン・カウントがしきい値を超えた場合、符号器は、符号化状態をラン・レベル符号化[「RLE」]に切り換える(1636)。次に、符号器は、DCBA符号化を使用してその係数を符号化する(1640)。

30

## 【 0 1 6 4 】

係数を符号化した後、符号器は、符号化状態を調べる(1650)。符号化状態がもはやDCBAでない場合(例えば、ゼロ係数のしきい値を超えた結果、符号器が、符号化状態をRLEに切り換えた場合)、符号器は、ラン・レンジおよびレベルとして係数を符号化することを始める(1660)。ラン・レベル符号化は、例えば、ハフマン符号化、ベクトル・ハフマン符号化、またはCBA符号化(場合により、前述した以前のCBA符号化とは異なるコンテキストを伴う)を含むいくつかの仕方で行うことができる。一部の実施形態では、ラン・レベル符号化は、1つのテーブルが、より短いランの可能性が高いデータを符号化するために使用され、もう1つのテーブルが、より長いランの可能性が高いデータを符号化するために使用される2つのハフマン符号テーブルを有するハフマン符号化を使用して行われる。符号器は、各テーブルを試みて、どちらのテーブルを符号器が使用したかを示すシグナル・ビットと共に、一つのテーブルからの符号を選択する。

40

## 【 0 1 6 5 】

符号化状態が切り換わっていない場合、符号器は、符号化されるべきさらなる係数が存在するかどうかを判定する(1670)。存在する場合、符号器は、次の係数を入力し(1620)、符号化プロセスを続ける。

## 【 0 1 6 6 】

## C．コンテキスト・ベースの算術復号化

50

図17は、復号器がラン・レングスおよび信号レベルの復号化に切り換えるための切替えポイントを、適応して決定する態様で、CBA符号化を使用して符号化された係数を復号化するため一実装形態によるテクニック(1700)を示している。適応切替えポイントは、データ中の連続するゼロ値係数のカウントに依存し、この連続するゼロ値係数は、最低周波数係数から最高周波数係数まで進む量子化された変換係数に関する信号レベルである。代替として、データは、別の形態および/または編成のオーディオ・データである。

#### 【0167】

最初に、符号器は、いくつかの変数を初期設定する。具体的には、復号器は、ラン・カウントを0に設定し(1710)、復号化状態を直接コンテキスト・ベースの算術(DCBA)に設定する(1712)。

10

#### 【0168】

復号器は、算術符号化において係数を表わすのに符号器が使用した数を参照し、その数から係数の値を抽出することにより、次の量子化された係数をDCBAを使用して復号化する(1720)。次に、復号器は、復号化された係数に基づいてラン・カウントを更新する(1730)(具体的には、復号化された係数がゼロ値であるかどうかに基づき、ラン・カウントをリセットするか、または増分する)。

#### 【0169】

復号器は、ラン・カウントがしきい値を超えているかどうかを調べる(1740)。ラン・カウントしきい値は、静的であること、または係数のブロックのサイズ(例えば、256個の係数の入力シーケンス中の4つのゼロ)などのファクタに依存すること、または何らかの別の形で適応することが可能である。ラン・カウントがしきい値を超えた場合、復号器は、ラン・レベル復号化を使用して符号化された係数を復号化することを開始する(1750)。ラン・レベル復号化は、例えば、ハフマン復号化、ベクトル・ハフマン復号化、またはCBA復号化(場合により、前述した以前のCBA復号化とは異なるコンテキストを伴う)を含むいくつかの仕方で行われることが可能である。

20

#### 【0170】

一の実施形態では、ラン・レベル復号化は、1つのテーブルが、より短いランの可能性が高いデータを復号化するために使用され、もう1つのテーブルが、より長いランの可能性が高いデータを復号化するために使用される、2つの可能なハフマン符号テーブルを有するハフマン符号化を使用して行われる。復号器が符号を受け取った際、その符号中のシグナル・ビットは、どちらのテーブルを符号器が使用したかを示し、復号器は、適切なテーブルの中でその符号をルックアップする。

30

ラン・カウントがしきい値を超えていない場合、復号器は、復号化が終わるまで係数を処理することを続ける(1760)。

#### 【0171】

##### VI. テーブルのない符号化(table-less coding)

ハフマン符号化を使用する一の実施形態では、図2の符号器(200)のような符号器が、ハフマン符号テーブルに関するエスケープ符号を使用して、特定のシンボル(またはシンボルの組合せ)がテーブルの中で関連する符号を有さないことを示す。ときとして、エスケープ符号は、特定のシンボル(例えば、レベルに関するスカラー・ハフマン符号テーブルの中で表わされていないレベル、ラン・レングスに関するスカラー・ハフマン符号テーブルの中で表わされていないラン・レングス等に関するスカラー値)が、ハフマン・テーブルからの符号を使用せずに符号化されるべきことを示すのに使用される。言い換えれば、そのシンボルは、「テーブルのない(table-less)」符号化テクニックを使用して符号化されることになる。

40

#### 【0172】

算術符号化を使用する一の実施形態では、エスケープ符号は、ときとして、特定のシンボルが算術符号化されるべきでないことを示すのに使用される。そのシンボルは、ハフマン・テーブルからの符号を使用して符号化されることが可能であり、または「テーブル

50

のない」符号化テクニックを使用して符号化されることも可能である。

【 0 1 7 3 】

一部のテーブルのない符号化技術は、固定長の符号を使用してシンボルを表わす。ただし、固定長の符号を使用することは、不必要に長い符号をもたらす可能性がある。

【 0 1 7 4 】

したがって、一部の実施形態では、テーブルのない符号化テクニックにおいて、量子化された変換係数などのシンボルが、それ以外で符号化されない場合、可変長の符号を使用して表わされる。図 3 の復号器 ( 3 0 0 ) のような復号器が、対応するテーブルのない復号化テクニックを実施する。

【 0 1 7 5 】

10

例えば、テーブル 6 が、そのようなテーブルのない符号化テクニックの一実装形態に関する擬似コードを示している。

【 0 1 7 6 】

テーブル 6 : 一実装形態におけるテーブルのない符号化テクニックのための擬似コード

```

If (value < 28) {
    Send "0";
    Send value using 8 bits;
}

else if (value < 216) {
    Send "10";
    Send value using 16 bits
}

else if (value < 224) {
    Send "110";
    Send value using 24 bits;
}

else if (value < 231) {
    Send "111";
    Send value using 31 bits;
}

```

20

30

【 0 1 7 7 】

係数を符号化するのに符号器が使用するビットの数は、その係数の値に依存する。符号器は、1 ビットの値、2 ビットの値、または 3 ビットの値を送って値を符号化するのに使用されるビットの数を示し、次に、8 ビット、16 ビット、24 ビット、または 31 ビットを使用して符号化された値自体を送る。係数を符号化するのに符号器が使用するビットの総数は、 $2^8$  より小さい値に関する 9 ビットから、 $2^{24}$  に等しいか、それより大きい、 $2^{31}$  より小さい値に対する 34 ビットまでの範囲である。

【 0 1 7 8 】

一続きの係数に関して、送られる平均のビットは、

40

$$P(0 \leq C < 2^8) * 9 + P(2^8 \leq C < 2^{16}) * 18 + P(2^{16} \leq C < 2^{24}) * 27 + P(2^{24} \leq C < 2^{31}) * 34$$

であり、ただし、 $P(m \leq C < n)$  は、入力シーケンスにおける示された範囲内の係数  $C$  の生起確率である。したがって、係数の大きいパーセンテージが小さい (例えば、 $2^{16}$  より小さい) 場合、相当なビットの節約が可能である。

【 0 1 7 9 】

代替として、符号器および復号器は、別のテーブルのない符号化テクニック / 復号化テクニックを使用する。

【 0 1 8 0 】

様々な前述の実施形態に関連して本発明の原理を説明し、例示してきたが、そのような

50

原理を逸脱することなく、前述の実施形態の構成および詳細を変更できることが認められよう。本明細書で説明したプログラム、プロセス、または方法は、特に明記しない限り、いずれの特定のタイプのコンピューティング環境にも関連する、または限定されることはないことを理解されたい。様々なタイプの汎用コンピューティング環境または特殊化されたコンピューティング環境が、本明細書で説明した教示による動作とともに使用すること、またはその動作を行うことが可能である。ソフトウェアで示した前述の実施形態の要素は、ハードウェアで実装することができ、またその逆も該当する。

#### 【 0 1 8 1 】

本発明の原理を適用することができる多くの可能な実施形態に鑑みて、頭記の特許請求の範囲の範囲および趣旨に含まれる可能性があるようなすべての実施形態、ならびに等価形態を本発明として主張する。

#### 【図面の簡単な説明】

#### 【 0 1 8 2 】

【図 1】説明する実施形態を実装することが可能な適切なコンピューティング環境を示すブロック図である。

【図 2】説明する実施形態を実装することが可能なオーディオ符号器を示すブロック図である。

【図 3】説明する実施形態を実装することが可能なオーディオ復号器を示すブロック図である。

【図 4】一般化されたマルチ・モード・オーディオ符号化テクニックを示す流れ図である。

【図 5】適応切替えポイント計算を伴うマルチ・モード・オーディオ符号化テクニックを示す流れ図である。

【図 6】一般化されたマルチ・モード・オーディオ復号化テクニックを示す流れ図である。

【図 7】一般化された可変次元ベクトル・ハフマン符号化テクニックを示す流れ図である。

【図 8】可変次元ベクトル・ハフマン符号化を使用してオーディオ・データを符号化するための詳細なテクニックを示す流れ図である。

【図 9】符号器がラン・レンジおよび信号レベルの符号化に変更するための切替えポイントを、適応して決定する態様の、直接信号レベルの可変次元ベクトル・ハフマン符号化のためのテクニックを示す流れ図である。

【図 10】一般化された可変次元ベクトル・ハフマン復号化テクニックを示す流れ図である。

【図 11】可変次元ベクトル・ハフマン符号化を使用して符号化されたベクトルを復号化するための詳細なテクニックを示す流れ図である。

【図 12】復号器がラン・レンジおよび信号レベルの復号化に変更するための切替えポイントを、適応して決定する態様の、直接信号レベルの可変次元ベクトル・ハフマン復号化のためのテクニックを示す流れ図である。

【図 13A】コンテキスト・ベースの算術符号化スキームにおける非ラン・レンジレベルに関する確率分布を示す図である。

【図 13B】コンテキスト・ベースの算術符号化スキームにおける非ラン・レンジレベルに関する確率分布を示す図である。

【図 13C】コンテキスト・ベースの算術符号化スキームにおける非ラン・レンジレベルに関する確率分布を示す図である。

【図 13D】コンテキスト・ベースの算術符号化スキームにおける非ラン・レンジレベルに関する確率分布を示す図である。

【図 14A】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジに関する確率分布を示す図である。

【図 14B】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジ

10

20

30

40

50



に関する確率分布を示す図である。

【図 1 4 C】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジスに関する確率分布を示す図である。

【図 1 4 D】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジスに関する確率分布を示す図である。

【図 1 4 E】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジスに関する確率分布を示す図である。

【図 1 4 F】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジスに関する確率分布を示す図である。

【図 1 4 G】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジスに関する確率分布を示す図である。

10

【図 1 4 H】コンテキスト・ベースの算術符号化スキームにおける異なるラン・レンジスに関する確率分布を示す図である。

【図 1 5 A】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 5 B】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 5 C】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 5 D】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

20

【図 1 5 E】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 5 F】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 5 G】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 5 H】コンテキスト・ベースの算術符号化スキームにおけるラン・レンジス符号化されたレベルに関する確率分布を示す図である。

【図 1 6】ラン・レンジスおよびラン・レベルの符号化に変更するための切替えポイントが符号器によって適応して決められる態様の、係数の直接コンテキスト・ベースの算術符号化のためのテクニックを示す流れ図である。

30

【図 1 7】復号器がラン・レンジスおよび信号レベルの復号化に切り換えるための切替えポイントを、適応して決定する態様の、コンテキスト・ベースの算術復号化のテクニックを示す流れ図である。

【符号の説明】

【 0 1 8 3 】

1 0 0 コンピューティング環境

1 1 0 処理ユニット

1 2 0 メモリ

1 3 0 基本的構成

1 4 0 ストレージ

1 5 0 入力デバイス

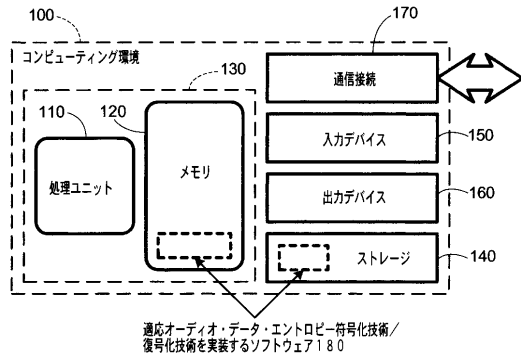
1 6 0 出力デバイス

1 7 0 通信接続

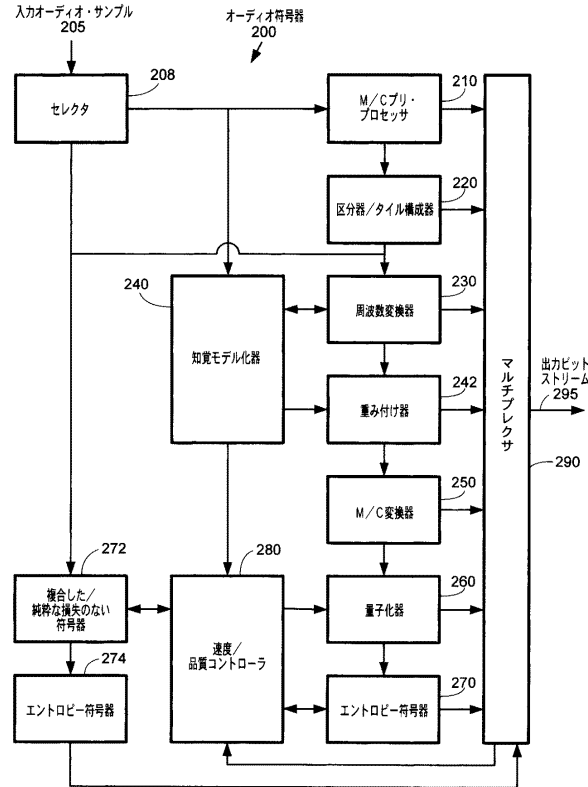
1 8 0 ソフトウェア

40

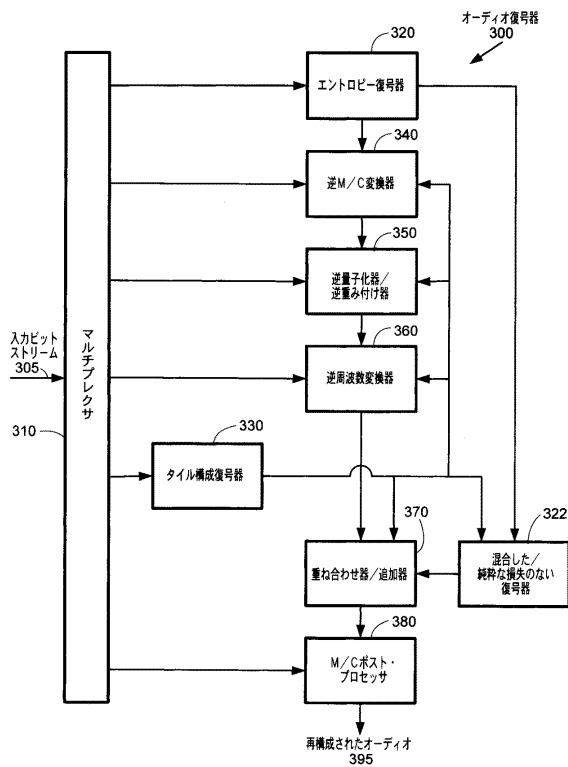
【図 1】



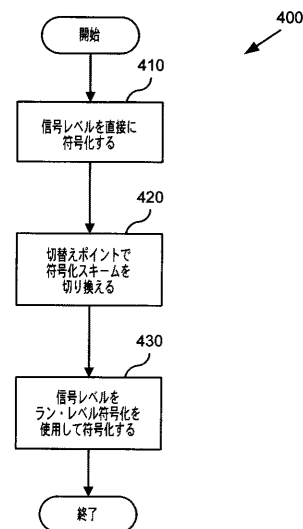
【図 2】



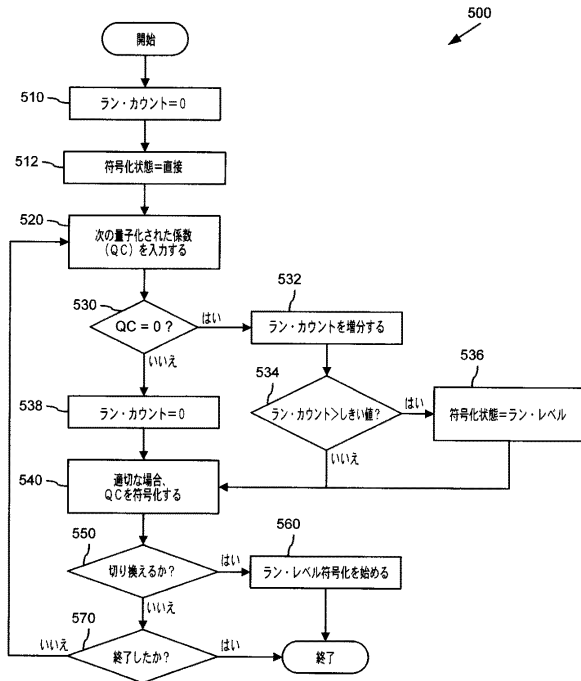
【図 3】



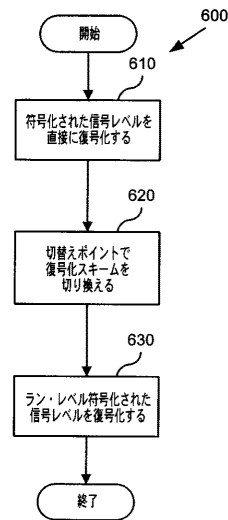
【図 4】



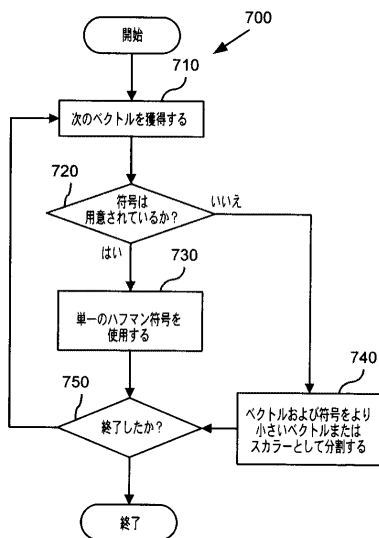
【図 5】



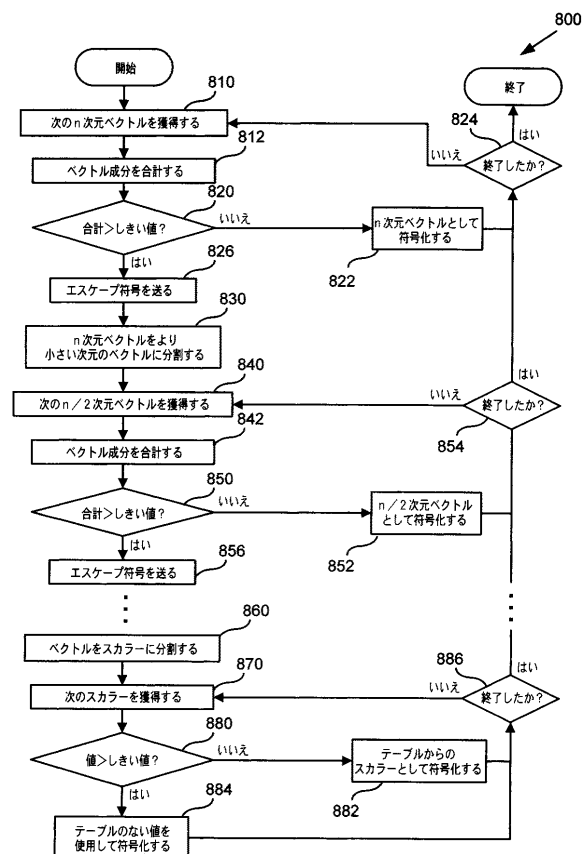
【図 6】



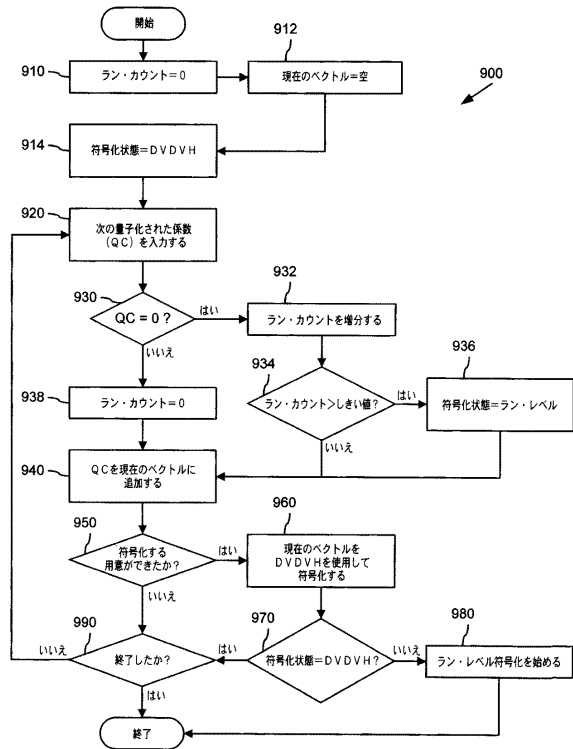
【図 7】



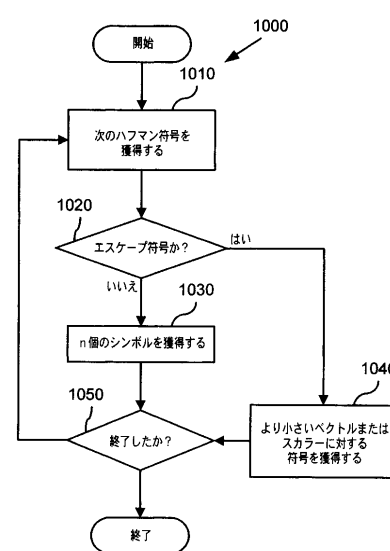
【図 8】



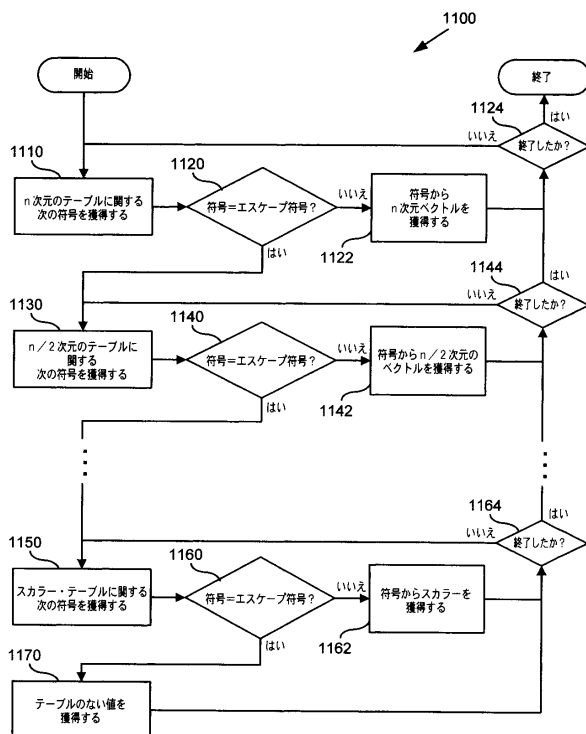
【図 9】



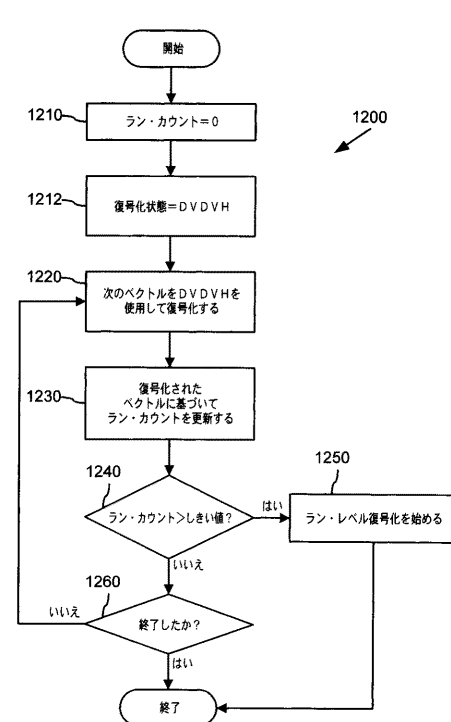
【図 10】



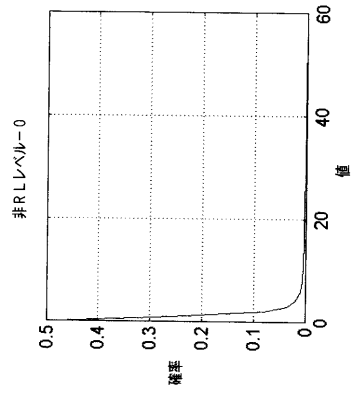
【図 11】



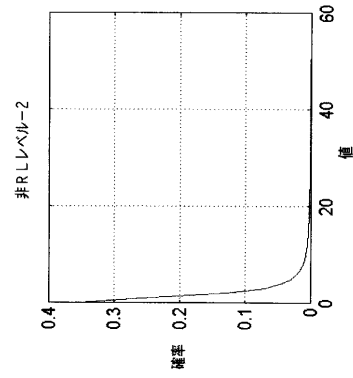
【図 12】



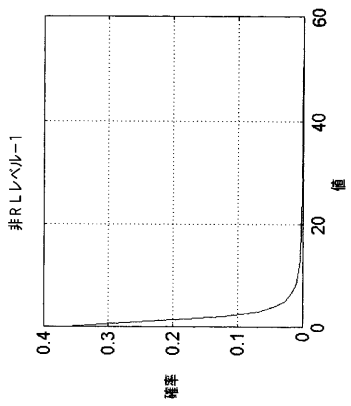
【図 13 A】



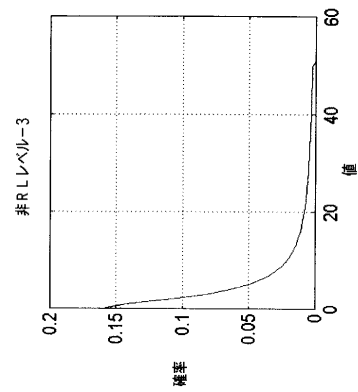
【図 13 C】



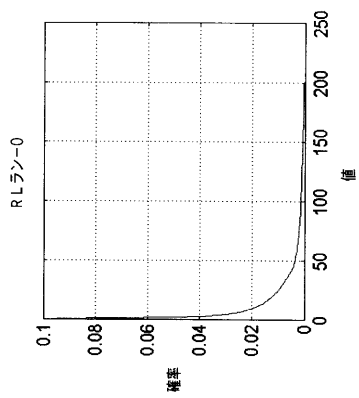
【図 13 B】



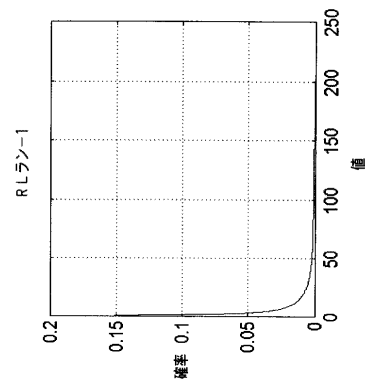
【図 13 D】



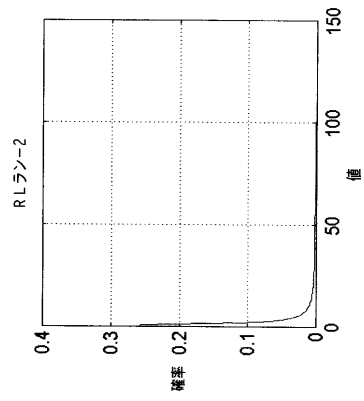
【図 14 A】



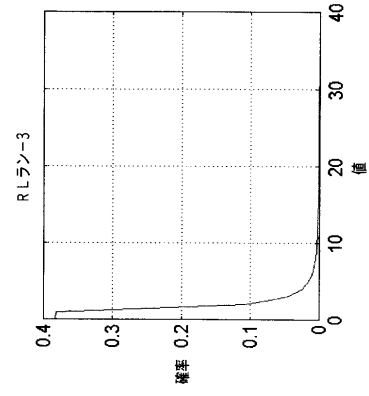
【図 14 B】



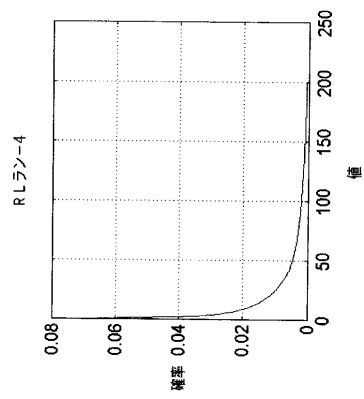
【図 1 4 C】



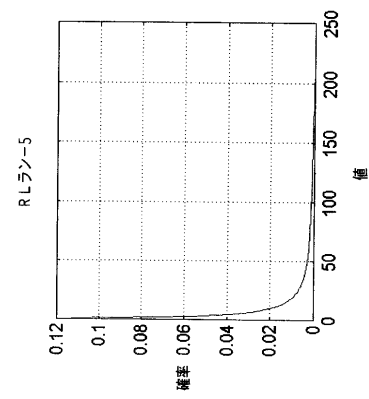
【図 1 4 D】



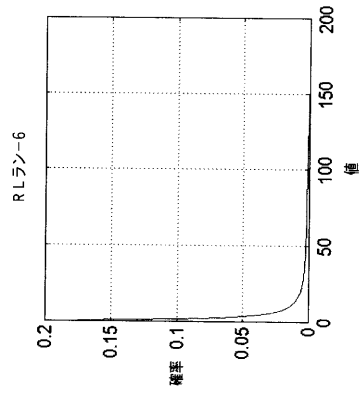
【図 1 4 E】



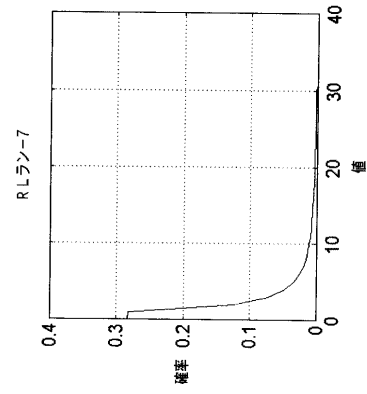
【図 1 4 F】



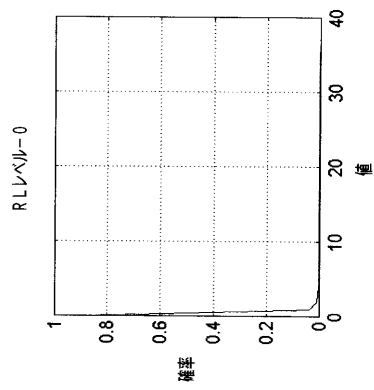
【図 1 4 G】



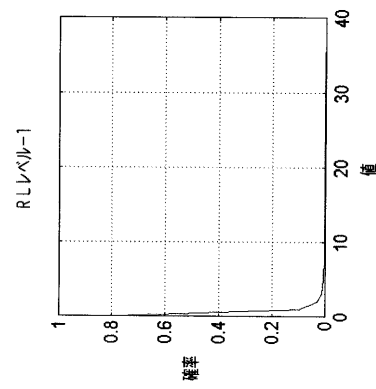
【図 1 4 H】



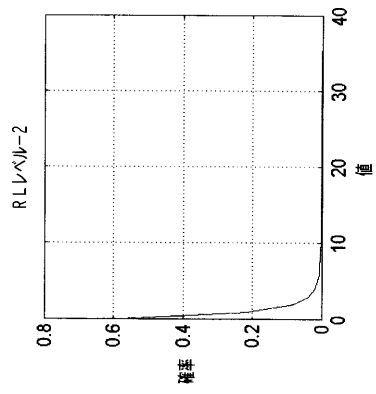
【図 1 5 A】



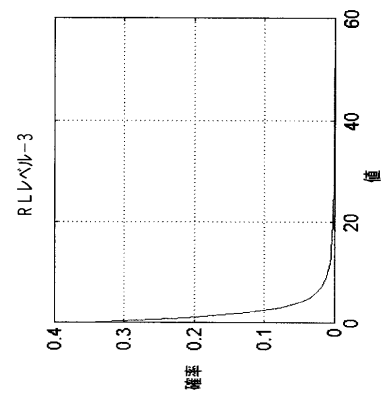
【図 1 5 B】



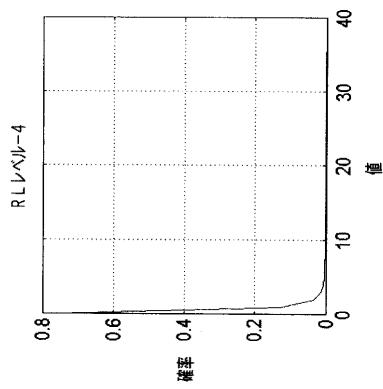
【図 15 C】



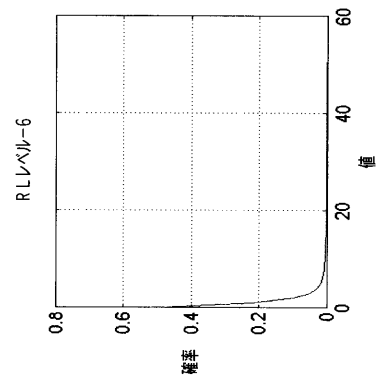
【図 15 D】



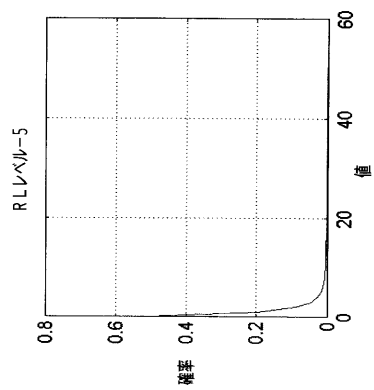
【図 15 E】



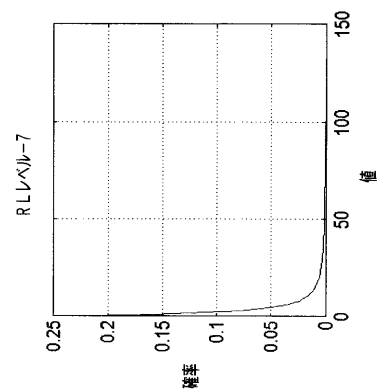
【図 15 G】



【図 15 F】

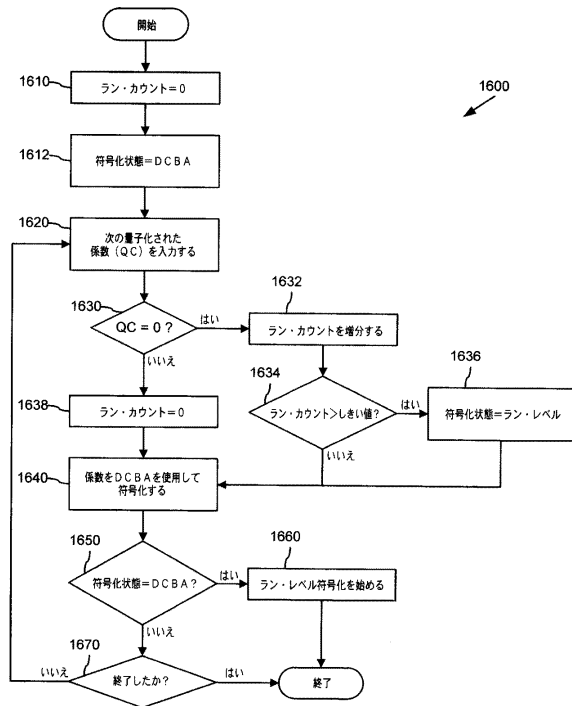


【図 15 H】

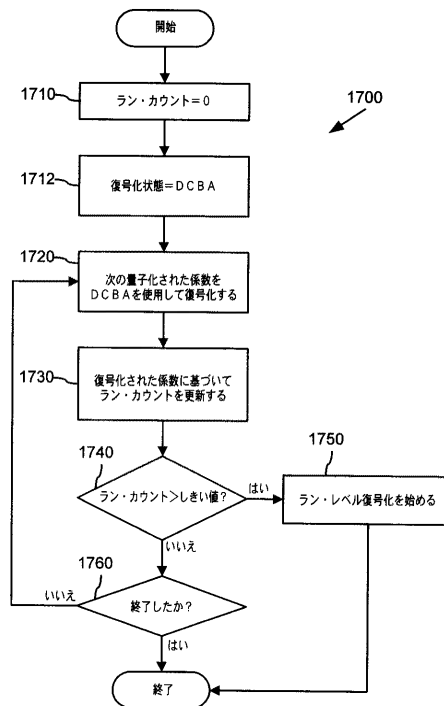




【図 16】



【図 17】



---

フロントページの続き

(72)発明者 ウェイ - ゲ チェン

アメリカ合衆国 98029 ワシントン州 イサコア サウスイースト 37 ストリート 2  
4635

審査官 山下 剛史

(56)参考文献 特開2002 - 158589 (JP, A)

特開平3 - 108824 (JP, A)

特開2000 - 338998 (JP, A)

国際公開第00 / 36754 (WO, A1)

(58)調査した分野(Int.Cl., DB名)

G10L 19 / 00 - 19 / 14

H03M 7 / 42