

(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2004-266587
(P2004-266587A)

(43) 公開日 平成16年9月24日(2004.9.24)

(51) Int. Cl. ⁷	F I	テーマコード (参考)
H03M 7/36	H03M 7/36	5D044
G10L 19/00	G11B 20/10 311	5D045
G10L 19/04	G11B 20/12 102	5J064
G11B 20/10	H03M 7/40	
G11B 20/12	G10L 9/14 J	
	審査請求 未請求 請求項の数 11 O L (全 18 頁) 最終頁に続く	

(21) 出願番号 特願2003-55114 (P2003-55114)
(22) 出願日 平成15年3月3日(2003.3.3)

(71) 出願人 000002897
大日本印刷株式会社
東京都新宿区市谷加賀町一丁目1番1号
(74) 代理人 100111659
弁理士 金山 聡
(72) 発明者 茂出木 敏雄
東京都新宿区市谷加賀町一丁目1番1号
大日本印刷株式会社内
Fターム(参考) 5D044 AB05 BC02 CC06 DE43 GK08
5D045 CC07 DA20
5J064 AA01 BA01 BA09 BB01 BB04
BC01 BC23 BC27 BD03

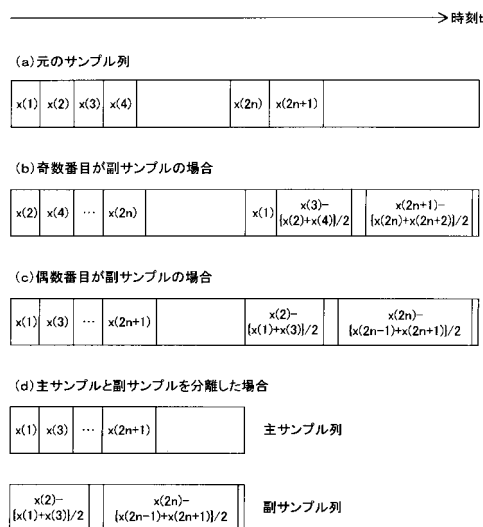
(54) 【発明の名称】 時系列信号の符号化装置および記録媒体

(57) 【要約】

【課題】 デジタル編集された高精細オーディオのワークデータを効率的に可逆圧縮することが可能な時系列信号の符号化装置および記録媒体を提供する。

【解決手段】 時系列のサンプル列を奇数番目と偶数番目に分けて、奇数番目のサンプルと隣接する両偶数番目のサンプルの平均との差分値、偶数番目のサンプルと隣接する両奇数番目のサンプルの平均との差分値を算出し、差分値が小さいものが続く方を主サンプル列、他方を副サンプル列として再配置する。再配置した主サンプル列、副サンプル列それぞれに対して線形予測誤差に基づく予測誤差値を求め、可変長符号化を行う。

【選択図】 図2



【特許請求の範囲】

【請求項 1】

時系列のサンプル列で構成される時系列信号に対して、前記全てのサンプル列を再現できるように情報量を圧縮する符号化装置であって、
前記サンプル列に対して、録音信号をサンプリングすることにより得られた主サンプル列と、主サンプル列を補間処理することにより得られた副サンプル列を分離すると共に、前記副サンプル列中の各副サンプルの値を、近傍の主サンプルの平均値と当該副サンプルの値との差分値に変換するサンプル列再配置手段と、
前記主サンプル列、前記変換された副サンプル列それぞれに対して、線形予測誤差を算出し、前記主サンプル列および前記副サンプル列の値をそれぞれ予測誤差値に変換する予測誤差変換手段と、
前記予測誤差値に変換された主サンプル列、副サンプル列を可変長で符号化する可変長符号化手段と、
を有することを特徴とする時系列信号の符号化装置。

10

【請求項 2】

請求項 1 において、
前記サンプル列再配置手段は、前記サンプル列の偶数番目に位置するサンプルの値が前後の奇数番目に位置するサンプル列の平均値に近い場合に、奇数番目に位置するサンプルを主サンプル列、偶数番目に位置するサンプルを副サンプル列に分離するものであり、
前記サンプル列の奇数番目に位置するサンプルの値が前後の偶数番目に位置するサンプル列の平均値に近い場合に、偶数番目に位置するサンプルを主サンプル列、奇数番目に位置するサンプルを副サンプル列に分離するものであることを特徴とする時系列信号の符号化装置。

20

【請求項 3】

請求項 1 において、
前記主サンプル列および前記副サンプル列変換手段による処理後の副サンプル列に対して、下位の所定数のビット成分を削除する下位固定ビット削除手段を有し、当該下位固定ビット削除手段による処理後のサンプル列に対して前記予測誤差変換手段が処理を行うものであることを特徴とする時系列信号の符号化装置。

30

【請求項 4】

請求項 1 において、
前記サンプル列が同一時刻に複数の値をもつ複数のチャンネルで構成されている場合、チャンネル間の同一時刻におけるサンプル同士の差分を算出し、いずれかのチャンネルのサンプル列を前記算出された差分値で更新するようにしたチャンネル間演算手段をさらに有することを特徴とする時系列信号の符号化装置。

【請求項 5】

請求項 1 において、
前記サンプル列の中で、サンプルの値が連続して同一値になっているサンプルを抽出し、前記サンプル列から削除すると共に、削除したサンプルの先頭時間位置と、サンプル個数と、サンプル値の 3 つの値を符号化する信号平坦部処理手段を、前記予測誤差変換手段の前段に有することを特徴とする時系列信号の符号化装置。

40

【請求項 6】

請求項 1 において、
前記サンプル列の中から所定の個数のサンプル列で構成されるフレームと同一内容のフレームが時間的に過去に存在する場合に、未来に位置するフレームを削除し、両フレームの先頭時間位置と、フレームを構成するサンプル個数を符号化する相関フレーム検出手段を、前記予測誤差変換手段の前段に有することを特徴とする時系列信号の符号化装置。

【請求項 7】

請求項 1 において、
前記予測誤差変換手段は、前記主サンプル列および副サンプル列に対して、時間的に過去

50

のサンプル列から、複数の予測計算式に基づいて、複数の予測誤差値の候補を算出し、その中から符号化対象の予測誤差値を選別するものであることを特徴とする時系列信号の符号化装置。

【請求項 8】

請求項 7 において、

前記複数の予測計算式の線形係数を、所定のサンプル数ごとに更新することを特徴とする時系列信号の符号化装置。

【請求項 9】

請求項 1 において、

前記予測誤差に変換されたサンプル値の絶対値に対して、全体を 1 ビット上位にずらし、正負符号を最下位 1 ビットに挿入する極性処理手段を、前記可変長符号化手段の前段に有することを特徴とする時系列信号の符号化装置。

10

【請求項 10】

請求項 1 において、

前記可変長符号化手段は、前記予測誤差値に変換された各サンプルのビット成分のうち、下位のビット成分をそのままのビット成分で符号化し、残りの上位ビット成分に対してビット成分を変更して符号化を行うものであることを特徴とする時系列信号の符号化装置。

【請求項 11】

与えられた時系列信号に対して、請求項 1 から請求項 10 のいずれかの時系列信号の符号化装置により出力された符号データを記録した記録媒体。

20

【発明の詳細な説明】

【0001】

【産業上の利用分野】

本発明は、音楽制作、音響データの素材保管、ロケ素材の中継など音楽制作分野、CD・DVD等のデジタル記録媒体を用いたオーディオ記録再生、遠隔医療における生体信号の解析・診断等の分野において好適なデータの圧縮符号化技術に関する。

【0002】

【従来の技術】

従来より、音響信号の圧縮には様々な手法が用いられている。音響信号を圧縮して符号化する手法として、MP3(MPEG-1/Layer 3)、AAC(MPEG-2/Layer 3)などが実用化されている。このような圧縮符号化方式により、音響信号を小さいデータとして扱うことが可能となり、データの記録・伝送の効率化に貢献している。

30

【0003】

最近では、上述のようなMP3、AAC等のロッシー符号化方式だけでなく、完全に復元することが可能なロスレス符号化方式も開発されており、音響素材の管理に用いられている(例えば、特許文献1参照)。

【0004】

【特許文献1】

特表2000-821199号公報

【0005】

【発明が解決しようとする課題】

高精細オーディオを扱う場合は、通常のオーディオ(音楽CD品質: サンプル周波数44.1kHz、量子化ビット数16ビット)に比べ、サンプル周波数または量子化ビット数を高く設定してサンプルしている。音楽編集分野では、さまざまな音をミックスして編集する作業がしばしば行われるが、この際、部分的に通常のオーディオがミックスされる場合があり、この通常のオーディオについては、高精細オーディオに合わせるべく、サンプルの補間、量子化ビット数の引き伸ばし等が行われる。このような補間処理を施された音響信号は、原理的に原音もつ情報量程度に圧縮を行うことが可能であるが、従来のロスレス符号化方式で符号化を行っても、その程度の圧縮効果が得られない。逆に、そのまま線形予測符号化を中心とした圧縮を行うと、補間された箇所の線形予測誤差

40

50

が増大して圧縮率が劣化する場合もあり、冗長箇所を圧縮するのとは逆効果にもなり得る。例えば、サンプリング周波数を2倍に拡大したオーディオデータは、理論上は50%以下に圧縮可能なはずであるが、現実には50%を超える圧縮率になってしまう。

【0006】

そこで、このような問題を解決するため、本発明は、デジタル編集途上の高精細オーディオのワークデータを効率的に可逆圧縮することが可能な時系列信号の符号化装置および記録媒体を提供することを課題とする。

【0007】

【課題を解決するための手段】

上記課題を解決するため、本発明では、時系列のサンプル列で構成される時系列信号に対して、前記全てのサンプル列を再現できるように情報量を圧縮する符号化装置として、前記サンプル列に対して、録音により作成された主サンプル列と、主サンプル列を補間処理することにより得られた副サンプル列を分離すると共に、前記副サンプル列中の各副サンプルの値を、近傍の主サンプルの平均値と当該副サンプルの値との差分値に変換するサンプル列再配置手段と、前記主サンプル列、前記変換された副サンプル列それぞれに対して、線形予測誤差を算出し、前記主サンプル列および前記副サンプル列の値をそれぞれ予測誤差値に変換する予測誤差変換手段と、前記予測誤差値に変換された主サンプル列、副サンプル列を可変長で符号化する可変長符号化手段を有する構成としたことを特徴とする。

10

【0008】

本発明によれば、時系列信号を構成するサンプル列を、録音により得られた主サンプル列と、主サンプル列を補間することにより得られた副サンプル列に再配置し、主サンプル列と副サンプル列それぞれに対して予測符号化し、可変長符号化するようにしたので、デジタル編集途上の高精細オーディオのワークデータを効率的に可逆圧縮することが可能となる。

20

【0009】

【発明の実施の形態】

以下、本発明の実施形態について図面を参照して詳細に説明する。

(第1の実施形態)

図1は、本発明に係る時系列信号の符号化装置の第1の実施形態を示す構成図である。図1において、10は時系列信号入力手段、20はサンプル列再配置手段、30は下位固定ビット削除手段、40は予測誤差変換手段、50はチャンネル間演算手段、60は極性処理手段、70は可変長符号化手段である。

30

【0010】

図1において、時系列信号入力手段10はデジタル音響信号等のデジタル化された音響信号を入力する機能を有している。サンプル列再配置手段20は、入力された時系列信号であるサンプル列を録音を基に得られたサンプル列である主サンプル列と主サンプル列を補間することにより得られた副サンプル列とに分離する機能を有している。下位固定ビット削除手段30は、量子化雑音成分である下位の所定数のビットを削除する機能を有している。予測誤差変換手段40は、線形予測誤差の手法を用いて、各サンプルの値を予測誤差値に変換する機能を有する。チャンネル間演算手段50は、複数のチャンネルからなるサンプル列の各チャンネル間の差分演算を行う機能を有する。極性処理手段60は、正負の値を補数表現により表した各サンプルのビット列を、正負の極性を表す1ビットと他のビット列に分ける処理を行う機能を有する。可変長符号化手段70は、各サンプルの値を可変ビット長で符号化する機能を有している。図1に示した装置は、実際には、コンピュータおよびコンピュータにインストールされた専用のソフトウェアプログラムにより実現される。

40

【0011】

次に、図1に示した時系列信号の符号化装置の処理動作について説明する。本発明では、時系列信号として複数の音響信号をミックスしたワークデータを扱う場合を例にとって説明する。このような音響信号は、上述のように、サンプリング周波数48kHz、量子化

50

ビット数 16 ビットの通常の音響信号や、サンプリング周波数 96 kHz、量子化ビット数 24 ビットの高精細の音響信号が混在している。このようにサンプリング周波数の異なる音響信号を混在させることにより得られる音響信号は、高精細の音響信号にサンプリング周波数を統一させて扱うことになる。この場合、サンプリング周波数 48 kHz の音響信号は、サンプリング周波数 96 kHz の音響信号にサンプル数を合わせるべく隣接するサンプルの平均値で間を補間していく。

【0012】

このような音響信号を模式的に示すと図 2 (a) のようになる。図 2 (a) において括弧内の数字は、1 から昇順に付されたサンプル番号であり、x は、そのサンプルの値を示している。このようなサンプル列である時系列信号が時系列信号入力手段 10 から入力されると、サンプル列再配置手段 20 は、奇数番目のサンプルについて、その両隣の偶数番目のサンプルの平均値との差分を演算すると共に、偶数番目のサンプルについて、その両隣の奇数番目のサンプルの平均値との差分を演算する。このときのサンプル列を模式的に示すと、それぞれ図 2 (b) (c) に示すようになる。なお、図 2 (b) の例では、演算を行わない偶数番目のサンプルを、図 2 (c) の例では、奇数番目のサンプルを、それぞれ時間的に過去に移動させた状態で示している。

10

【0013】

この差分演算の結果、差分値が小さいものが多い方を副サンプル列とし、少ない方を主サンプル列とする。図 2 の例では、図 2 (b) の配列の後半分の各値と、図 2 (c) の配列の後半分の各値とを比較することになる。例えば、奇数番目のサンプルが補間によって得られたものである場合、図 2 (b) に示した配列の後半の値が 0 になる。一方、偶数番目のサンプルが補間によって得られたものである場合、図 2 (c) に示した配列の後半の値が 0 に近くなる。例えば、図 2 (b) に示す配列の後半に 0 近辺の値が多い場合、偶数番目のサンプルの集合を主サンプル列、奇数番目のサンプルの集合を副サンプル列として分離する。また、サンプル列再配置手段 20 の処理においては、図 2 (b) (c) に示したように主サンプルを時間的に過去に移動し、副サンプルを時間的に未来に移動させるようにしても良いが、主サンプルと副サンプルを分離して扱うようにしても良い。例えば、奇数番目が副サンプルの場合には、図 2 (d) に示すように主サンプルと副サンプルを分離する。本発明においては、本来のサンプルを利用して補間することにより得られたサンプルを含んだサンプル列に対して線形予測を行うことにより、逆にデータ

20

30

【0014】

次に、下位固定ビット削除手段 30 が、主サンプル列、副サンプル列の各サンプルの下位の所定数のビットを分離する。これは、量子化ビット数が 16 ビットのデータを高精細の音響信号と合わせるために 24 ビットに変換している場合に、冗長な下位ビット成分を削除するために行い、この処理を行わないと、符号化された情報量は 3 / 2 倍に増大することになる。一方、ミックスする基になった素材の音響信号が全て高精細の 24 ビットで量子化されている場合は、下位固定ビット削除手段 30 による処理を行う必要はないが、同様に削除を行い、削除された下位ビットデータ配列を出力符号データの一部として別途記録する手段もとれ、その方が後段の予測誤差変換手段以降の処理負荷が軽減する。下位固定ビット削除手段 30 については、動作させるかどうかをあらかじめ設定しておくことができる。

40

【0015】

続いて、主サンプル列、副サンプル列の各サンプルの値を、予測誤差変換手段 40 が予測誤差値に変換する。あるサンプルにおける予測誤差値の算出は、時間的に過去に位置する直前の 1 つもしくは複数のサンプルの値を利用して行われる。本実施形態では、利用する直前のサンプル数を動的に変化させる手法を用いている。以下に、このような適応型線形

50

予測符号化について説明する。予測誤差変換手段40により行われる適応型線形予測符号化の処理概要を図3のフローチャートに示す。まず、あらかじめ準備された複数の予測計算式を用いて、各予測計算式に対応した線形予測誤差を算出する(ステップS1)。具体的には、サンプル番号tの予測誤差を算出する予測計算式として、以下の〔数式1〕～〔数式6〕を用意している。

【0016】

〔数式1〕

$$e_0(t) = x(t) - e_0(t-1) / 2$$

【0017】

〔数式2〕

$$e_1(t) = x(t) - a_{11} \cdot x(t-1) - e_1(t-1) / 2$$

【0018】

〔数式3〕

$$e_2(t) = x(t) - a_{21} \cdot x(t-1) - a_{22} \cdot x(t-2) - e_2(t-1) / 2$$

【0019】

〔数式4〕

$$e_3(t) = x(t) - a_{31} \cdot x(t-1) - a_{32} \cdot x(t-2) - a_{33} \cdot x(t-3) - e_3(t-1) / 2$$

【0020】

〔数式5〕

$$e_4(t) = x(t) - a_{41} \cdot x(t-1) - a_{42} \cdot x(t-2) - a_{43} \cdot x(t-3) - a_{44} \cdot x(t-4) - e_4(t-1) / 2$$

【0021】

〔数式6〕

$$e_5(t) = x(t) - a_{51} \cdot x(t-1) - a_{52} \cdot x(t-2) - a_{53} \cdot x(t-3) - a_{54} \cdot x(t-4) - a_{55} \cdot x(t-5) - e_5(t-1) / 2$$

上記〔数式1〕～〔数式6〕において、 $e_0(t) \sim e_5(t)$ は各予測計算式による時刻tのサンプルにおける予測誤差であり、 $x(t) \sim x(t-5)$ は時刻t～t-5におけるサンプル値である。

【0023】

上記〔数式3〕における「 $a_{21} \cdot x(t-1) + a_{22} \cdot x(t-2)$ 」、上記〔数式4〕における「 $a_{31} \cdot x(t-1) + a_{32} \cdot x(t-2) + a_{33} \cdot x(t-3)$ 」、上記〔数式5〕における「 $a_{41} \cdot x(t-1) + a_{42} \cdot x(t-2) + a_{43} \cdot x(t-3) + a_{44} \cdot x(t-4)$ 」、上記〔数式6〕における「 $a_{51} \cdot x(t-1) + a_{52} \cdot x(t-2) + a_{53} \cdot x(t-3) + a_{54} \cdot x(t-4) + a_{55} \cdot x(t-5)$ 」は過去の2～5個のサンプルに基づく線形予測成分である。この線形予測成分、および、直前のサンプルにおいて算出された予測誤差「 $e_1(t-1) / 2$ 」～「 $e_5(t-1) / 2$ 」(誤差フィードバック成分)を用いて時刻tにおける予測誤差 $e_0(t) \sim e_5(t)$ を算出する。

【0024】

上記の係数 $a_{11} \sim a_{55}$ には初期値として、 $a_{11} = 1$ 、 $a_{21} = 2$ 、 $a_{22} = -1$ 、 $a_{31} = 3$ 、 $a_{32} = -3$ 、 $a_{33} = 1$ 、 $a_{41} = 4$ 、 $a_{42} = -6$ 、 $a_{43} = 4$ 、 $a_{44} = -1$ 、 $a_{51} = 5$ 、 $a_{52} = -10$ 、 $a_{53} = 10$ 、 $a_{54} = -5$ 、 $a_{55} = 1$ なる値が各々設定されているが、本実施形態では、これらの係数を動的に変化させる。具体的には、Levinson-Durbinのアルゴリズムを利用した以下の〔数式7〕を用いて係数 $a_{11} \sim a_{55}$ を決定する。

【0025】

〔数式7〕

10

20

30

40

50

$$\begin{aligned}
 (k) &= 1 / (N - K) \cdot \sum_{j=1, N-K}^{N-K} x(j) \cdot x(j+k) \\
 k_i &= - \{ (i) + \sum_{j=1, i-1}^{i-1} a_j (i-1) \cdot (i-j) \} / E(i-1) \\
 a_i (i) &= k_i \\
 a_j (i) &= a_j (i-1) + k_i \cdot a_{i-j} (i-1) \quad \text{ただし、} 1 \leq j \leq i-1 \\
 E(i) &= (1 - k_i^2) E(i-1)
 \end{aligned}$$

【0026】

上記〔数式7〕において、 (k) は、 N 個のサンプル $x(j)$ ($j = 1, \dots, N$)において、最大値 K (上記例では5)の範囲で k サンプルシフトさせたサンプル列との自己相関値である。なお、 N は K に対して十分大きな数値をとっている (例えば $K = 5$ の場合、 $N = 32768$)。〔数式7〕は、 $i = 1$ から $i = K$ まで再帰的に繰り返し、最終的に得られた $a_j(K)$ が過去 K 個のサンプルに対応する係数になるとともに、各フェーズにおいて得られた中間結果である $a_j(i)$ が係数 a_{ij} となる。ステップS1においては、上記〔数式7〕により決定した係数を用いて、〔数式1〕～〔数式6〕の各計算式で計算を行うことになる。〔数式7〕による計算は、実際には後述するステップS7において行われるものである。また、係数を決定するには、過去の数サンプル分の値を必要とするので、初めの $N - 1$ サンプルについては、前述した初期係数で〔数式1〕～〔数式6〕の計算を行うことになる。

10

【0027】

続いて、上記各予測計算式別の予測誤差値の絶対値の累積である累積誤差が最小となる線形予測誤差をそのサンプルの予測誤差として選出する (ステップS2)。ここでは、累積誤差という考え方を用いている。具体的には、各予測計算式〔数式1〕～〔数式6〕により算出された予測誤差の過去のサンプルについての累積値を $A_0 \sim A_5$ として設定する。そして、この累積誤差 $A_0 \sim A_5$ のうち、最小となるものに対応する予測誤差を選出する。例えば、 $A_0 \sim A_5$ のうち、 A_2 が最小であったとする。この場合、〔数式3〕で算出された予測誤差 $e_2(t)$ を符号化対象とする予測誤差 $e(t)$ として選出することになる。選出された予測誤差 $e(t)$ はサンプルの元の値 $x(t)$ と置き換えられて以降処理が行われることになる。

20

【0028】

続いて、累積誤差 $A_0 \sim A_5$ に各予測誤差 $e_0(t) \sim e_5(t)$ の絶対値を加算する (ステップS3)。具体的には、以下の〔数式8〕に示すように、累積誤差値となる変数 $A_0 \sim A_5$ を更新していく。同時に、各サンプルの処理を行う度に、カウンタ C_1 、 C_2 を1つずつ加算していく処理を行う。

30

【0029】

〔数式8〕

$$\begin{aligned}
 A_0 &= A_0 + |e_0(t)| \\
 A_1 &= A_1 + |e_1(t)| \\
 A_2 &= A_2 + |e_2(t)| \\
 A_3 &= A_3 + |e_3(t)| \\
 A_4 &= A_4 + |e_4(t)| \\
 A_5 &= A_5 + |e_5(t)|
 \end{aligned}$$

40

【0030】

続いて、カウンタ C_1 が所定回数を超えたかどうかの判定を行う (ステップS4)。本実施形態では、この所定回数を100回として設定している。すなわち、カウンタ C_1 が100を超えたかどうかの判定を行う。

【0031】

この結果、カウンタが100を超えていたら、累積誤差を半分にする (ステップS5)。具体的には、以下の〔数式9〕に示すように、累積誤差となる変数 $A_0 \sim A_5$ を2で除算する。同時に、カウンタ C_1 を0にリセットする。すなわち、ここでの $A_0 \sim A_5$ は純粋な意味での累積誤差ではなく、累積誤差の移動平均となっている。本実施形態では、直前の最大100サンプルまでは累積されるが、それ以前のものは半分になるように処理する

50

。これにより、時間的に離れたサンプルの影響が小さくなるようにしている。

【0032】

〔数式9〕

$$A_0 = (A_0) / 2$$

$$A_1 = (A_1) / 2$$

$$A_2 = (A_2) / 2$$

$$A_3 = (A_3) / 2$$

$$A_4 = (A_4) / 2$$

$$A_5 = (A_5) / 2$$

【0033】

続いて、カウンタC2が所定回数を超えたかどうかの判定を行う(ステップS6)。本実施形態では、この所定回数を32768回として設定している。すなわち、カウンタC2が32768を超えたかどうかの判定を行う。

【0034】

この結果、カウンタC2が32768を超えていたら、係数 $a_{11} \sim a_{55}$ の再計算を行う(ステップS7)。具体的には、上記〔数式7〕を用いて、係数 $a_{11} \sim a_{55}$ を計算し直すことになる。同時に、カウンタC2を0にリセットする。

【0035】

上記ステップS1～ステップS7の処理を主サンプル列および副サンプル列のサンプルに渡って実行することにより、全サンプルの値が元の振幅値 $x(t)$ から対象誤差 $e(t)$ に置き換えられることになる。本実施形態では、特に、複数の予測式の係数を動的に変化させることにより、より精度の高い予測誤差を算出することが可能になる。

【0036】

次に、チャンネル間演算手段50が、予測誤差値が記録された各チャンネルの主サンプル、副サンプルに対して、チャンネル間の差分演算を行う。このチャンネル間差分演算の処理概要を図4のフローチャートに示す。まず、主サンプルを読み込む(ステップS11)。読み込んだ主サンプルがLチャンネルのものであれば、主サンプルの値をメモリに格納すると共に、次の極性処理手段60に出力する(ステップS12)。一方、読み込んだ主サンプルがRチャンネルのものであれば、ステップS13以降の処理を行う。なお、主サンプルは、各チャンネルのものが交互に読み込まれるので、判断を行う必要はない。例えば、サンプル番号 $t=1$ のLチャンネルのサンプルを読み込んだら、次はサンプル番号 $t=1$ のRチャンネルのサンプル、その次は、サンプル番号 $t=2$ のLチャンネルのサンプルというように順番が決まっているので、交互にステップS12の処理とステップS13以降の処理とを切替えるようにすれば良い。

【0037】

Rチャンネルのサンプルの場合は、変数 A_0 と A_d の比較を行う(ステップS13)。ここで、 A_0 はRチャンネルのサンプル値の絶対値の累積であり、 A_d はRチャンネルとLチャンネルのサンプル値の差分の絶対値の累積である。変数 A_0 、 A_d 共に初期値は0である。ステップS13において、 $A_d < A_0$ であれば、Rチャンネルのサンプルをそのままの値で、次の極性処理手段60に出力する(ステップS14)。さらに、累積値 A_0 を以下の〔数式10〕の第1式に示すように更新する。具体的には、Rチャンネルのサンプル値 e_R の絶対値を累積値 A_0 に加えることになる。

【0038】

ステップS13において、 $A_d < A_0$ であれば、上記ステップS12においてメモリに格納したLチャンネルのサンプルとの差分を算出し、差分値を次の極性処理手段60に出力する(ステップS15)。さらに、累積値 A_d を以下の〔数式10〕の第2式に示すように更新する。具体的には、RチャンネルとLチャンネルのサンプルの差分値 $e_R - e_L$ の絶対値を累積値 A_d に加えることになる。

【0039】

〔数式10〕

10

20

30

40

50

$$A_o = A_o + |e_R|$$

$$A_d = A_d + |e_R - e_L|$$

【0040】

続いて、L、Rの1対のサンプルを処理したことを示すカウンタC3を1つ加算する（ステップS16）。

【0041】

続いて、カウンタが所定回数を超えたかどうかの判定を行う（ステップS17）。本実施形態では、この所定回数を100回として設定している。すなわち、カウンタC3が100を超えたかどうかの判定を行う。

【0042】

この結果、カウンタC3が100を超えていたら、累積値A_o、A_dを半分にする（ステップS18）。具体的には、以下の〔数式11〕に示すように、累積値となる変数A_o、A_dを2で除算する。同時に、カウンタCも半分にリセットする。すなわち、ここでの変数A_o、A_dも上記〔数式8〕におけるA₀～A₅と同様、純粋な意味での累積値ではなく、累積値の移動平均となっている。本実施形態では、直前の最大100サンプルまでは累積されるが、それ以前のは半分になるように処理する。これにより、時間的に離れたサンプルの影響が小さくなるようにしている。

10

【0043】

〔数式11〕

$$A_o = A_o / 2$$

$$A_d = A_d / 2$$

$$C3 = C3 / 2$$

20

【0044】

上記ステップS11～ステップS18の処理を主サンプル列中の全主サンプルに渡って実行することにより、Rチャンネルの全サンプルの値が、Lチャンネルとの差分値に置き換えられることになる。ただし、上述の処理から明らかなように、累積値A_o、A_dの大小関係によっては、Rチャンネルのサンプル値がそのまま記録されるサンプルも存在する。なお、チャンネル間演算手段50では、Lチャンネルのサンプルは、全てそのままの値で出力されることになる。主サンプル列中の各サンプルに対して処理を終えたら、副サンプルに対しても同様に処理を行う。なお、チャンネルが1つだけのモノラルの音響信号に対しては、チャンネル間演算手段50による処理は省略される。

30

【0045】

続いて、極性処理手段60が、各サンプルの正負極性処理を行う。上記予測誤差変換手段40およびチャンネル間演算手段50の処理により各サンプルの値は、振幅値から予測誤差に置き換えられると共に、Rチャンネルの値は、Lチャンネルとの差分に置き換えられたが、各サンプルのビット形式は、当初のままである。通常、コンピュータ等の計算機で演算される場合は、各データは32ビット単位で処理され、2の補数表現を用いて表現されている。これを、正負の符号付き絶対値表現に変換し、なおかつ、その絶対値部分を上位に1ビット移動させ、正負の符号ビットをLSB（最下位ビット）に移動させる。極性処理手段60によるビット構成の変換の様子を模式的に示すと図5のようになる。図5（a）は処理前のビット構成であり、図5（b）は処理後のビット構成である。このように正負の符号ビットをLSBに移動させるのは、後の可変長符号化手段70の処理で、各サンプルのビット長を検出し易くするためである。

40

【0046】

次に、可変長符号化手段70が、各サンプルを可変長に変換する処理を行っていく。本実施形態における可変長符号化は、一般にゴロム符号化と呼ばれる方式を採用している。具体的には、1サンプルを構成するビット成分を上位ビット成分と下位ビット成分に分け、下位ビット成分は変更を加えずそのままとし、上位ビット成分は、上位ビットだけを十進数変換した数値分のビット「0」を並べ、最後にセパレータビット「1」を加えた配列とする。例えば、8ビットのビット成分「00101000」を考えてみる。このとき、下

50

位ビット成分を4ビットとすると、下位ビット成分は「1000」となる。上位ビットは「0010」であるため、これを十進数変換した「2」個分の「0」を配列して最後に「1」を加えた「001」に変換される。この結果、8ビットのビット列「00101000」は、7ビットのビット列「0011000」に変換されることになる。本実施形態では、変換の前後でビット成分を不変とする下位ビット成分のビット長を各サンプルで可変とするようにしている。

【0047】

以下、可変長符号化手段70が行う処理を具体的に説明していく。図6は可変長符号化の概要を示すフローチャートである。まず、過去のサンプルのビット長の移動平均である平均ビット長 B_f を算出する(ステップS21)。平均ビット長 B_f は、過去のビット長の累積値である累積ビット長 R_B を、過去のサンプル数を基にしたカウンタ C_4 で除算することにより求められる。すなわち、 $B_f = R_B / C_4$ で算出される。累積ビット長 R_B は、初期状態では0であるので、 $t = 1$ のサンプルを処理する場合には、 $t = 1$ のサンプルのビット長 $B_d(t)$ を初期値として設定しておく。また、初期のカウンタ $C_4 = 1$ と設定する。

10

【0048】

続いて、時刻 t におけるサンプルのビット長 $B_d(t)$ を算出する(ステップS22)。 $t = 2$ 以降のサンプルについては、平均ビット長 B_f の算出後、サンプルのビット長 $B_d(t)$ を算出する。このビット長 $B_d(t)$ は、上記極性処理手段60によりビット構成の変換を行ったことにより算出し易くなっている。図5(b)に示したようなビット構成に変換したことにより、各サンプルのビット構成において先頭にビット「1」が出現したところからがビット長となる。次に、変更部のビット長 B_v を算出する(ステップS23)。これは、上記サンプルのビット長 $B_d(t)$ から平均ビット長 B_f を減じることにより算出される。続いて、データの符号出力を行う(ステップS24)。具体的には、上位 B_v ビットを十進数変換した数値分だけ「0」を出力した後、セパレータビット「1」を出力し、下位 B_f ビットを不変部として出力する。符号出力は、ハードディスク、CD-R等の外部記憶装置への記録として行われることになる。次に、累積ビット長 R_B にビット長 $B_d(t)$ を加算する(ステップS25)。同時に、各サンプルの処理を行う度に、カウンタ C_4 を1ずつ加算していく処理を行う。続いて、カウンタ C_4 が所定の数を越えたかどうかを判定する(ステップS26)。所定の数としては、ここでも100程度を設定している。そのため、カウンタ4が100を超えたかどうかを判断することになる。この結果、カウンタが100を超えていたら、累積ビット長 R_B を半分にする(ステップS27)。具体的には、累積ビット長となる変数 R_B を2で除算する。同時に、カウンタ C_4 を1/2にする。

20

30

【0049】

上記のようにして、各サンプルについて可変ビット長での符号化が行われて行く。符号化により得られた可変長サンプルは、符号データとして出力される。なお、可変長符号化手段70には、上記のような処理を行うに先立ち量子化雑音成分を分離する機能を持たせておいても良い。具体的には、極性処理手段60による処理後の各サンプルの下位の所定数のビットを量子化雑音成分とみなして分離する。例えば各サンプルが16ビットで表現されている場合、本実施形態では、上位ビット12ビットと、下位ビット4ビットに分離する。この分離は、基本的に、A/D変換機等、音響信号をデジタル化する際に用いる回路の熱雑音を分離するために行う。そのため、熱雑音であると考えられる下位ビットを分離するのである。下位ビットとして、どの程度分離するかは、音源や利用した回路の特性によっても変化するが、通常量子化ビット数の1/4程度とすることが望ましい。したがって、ここでは、16ビットの1/4にあたる4ビットを下位ビットとして分離しているのである。

40

【0050】

ここで、上位ビットと下位ビットのデータ分離の様子を図7に模式的に示す。図7において、Hは上位ビットもしくは上位サンプルデータを示し、Lは下位ビットもしくは下位サ

50

ンプルデータを示す。図7(a)は分離前のサンプルデータである。可変長符号化手段70により、サンプルデータは、図7(b)に示す上位サンプルデータと図7(c)に示す下位サンプルデータに分離された後処理されることになる。なお、上位ビットに含まれる符号ビットは、そのまま上位サンプルデータに含まれて分離される。このように、量子化雑音成分の分離を行った場合には、残りの上位ビットに対して、上記図6に示したフローチャートに従った可変長符号化が行われ、下位ビットについては、そのまま固定長で符号化が行われる。

【0051】

以上のようにして得られた符号データは、コンピュータに接続されたハードディスク等の記憶装置等に随時記憶され、その後、必要な記憶媒体に対応するフォーマットで記憶される。

10

【0052】

(第2の実施形態)

続いて、本発明第2の実施形態に係る符号化装置について説明する。図8は、本発明第2の実施形態に係る符号化装置の機能ブロック図である。図8において、図1に示した構成と同様の機能を有するものについては、同一符号を付して説明を省略する。第1の実施形態と異なる点は、信号平坦部処理手段80と相関フレーム検出手段90が加わったことである。図2において、信号平坦部処理手段80は、各チャンネルごとのサンプル列に対して、信号の値が一定である平坦部を検出し、効率的に符号化する機能を有する。相関フレーム検出手段90は、各サンプル列に対して、所定の区間をフレームとして設定した後、フレーム間で対応する全てのサンプル値が同一になっている相関フレームを検出し、時間的に後方(未来)に位置する相関フレームを削除する機能を有する。図8に示した装置は、実際には、コンピュータおよびコンピュータにインストールされた専用のソフトウェアプログラムにより実現される。

20

【0053】

続いて、図8に示した符号化装置の処理動作について説明する。まず、時系列信号入力手段10より上記のようなミックスされた音響信号を入力する。すると、チャンネル間演算手段50が上記図に示した手順に従って、チャンネル間の差分演算処理を行う。続いて、サンプル列再配置手段20が、上記図2に示したような処理で主サンプル列と副サンプル列に再配置する。その後、下位固定ビット削除手段30が各サンプルの下位ビットを削除する。

30

【0054】

次に、信号平坦部処理手段80が、信号平坦部の処理を行う。信号平坦部とは、同一の信号レベルが連続する部分のことをいう。特に信号レベルが「0」の無音部、および信号レベルの絶対値が最大の飽和部に現れることが多い。無音部は実際に無音であるか、音が非常に小さく記録されなかった場合に生じるが、飽和部は、信号の録音およびA/D変換の過程において生じる。無音部、飽和部またはそれ以外の同一信号レベルが連続する場合のいずれであっても、信号平坦部は、同一の信号レベルが所定の時間(所定のサンプル数)連続して記録される。このため、この部分は圧縮し易いデータになっている。具体的には、信号平坦部の先頭時刻位置と、同一信号レベルが続くサンプルの個数と、信号レベル(サンプル値)の3つの値を信号平坦部データとして各チャンネルのサンプル列と分離して記録する。各チャンネルのサンプル列からは、信号平坦部が削除される。これを模式的に示すと図9(a)(b)に示すようになる。図9(a)は、信号平坦部処理前のサンプル列である。図9(a)において、網掛けで示した部分は信号平坦部を示す。信号平坦部処理手段80の処理により、信号平坦部は元のサンプル列からは削除され、図9(b)に示すようになる。ただし、復号時に元通りに復元するために、分離された信号平坦部は、信号平坦部データとして図9(c)に示すような形式で記録しておく。

40

【0055】

信号平坦部データは、上述のように、信号平坦部ごとに、その先頭時刻(サンプル番号)、サンプル数、サンプル値の3属性で記録する。ここで、先頭時刻とは、信号の開始位置

50

からの時刻であり、図9(c)の例では、先頭からのサンプル番号で記録している。このサンプル番号をサンプリング周波数で除算すれば、時刻に変換されることになる。サンプル数は、そのサンプル値がどの程度連続して続くかを示す情報である。なお、サンプル数の代わりに信号平坦部の終了時刻を記録するようにしても良い。サンプル値は、デジタル化された信号レベルを示している。符号付き16ビットで表現した場合は、最大値は「32767」、最小値は「-32768」となる。すなわち、「0」は無音部、「32767」および「-32768」は飽和部を示している。ただし、信号平坦部処理手段80は、信号平坦部を無条件には処理しない。本発明は、データの圧縮を目的としているため、サンプル列の削減分よりも信号平坦部データが大きくなると意味がないからである。したがって、信号平坦部となるサンプルが所定数以上連続する場合に限り信号平坦部データを作成して各チャンネルのサンプル列から分離するのである。

10

【0056】

続いて、各チャンネルのサンプル列に対して、相関フレーム検出手段90が、所定の区間長をもつフレームを設定して、設定されたフレーム間の比較を行う。本実施形態では、フレーム長をサンプル列の開始時刻から終了時刻までの全区間に渡って固定長としている。具体的には、1フレームを512サンプルとしている。相関フレーム検出手段90は、各チャンネルのサンプル列の先頭から512サンプルずつを1フレームとして設定し、フレーム間で全サンプルが一致する相関フレームを求めていくことになる。具体的な手順を図10のフローチャートに従って説明する。

【0057】

まず、相関フレーム検出手段90は、所定のサンプル数単位でフレーム化を行う(ステップS31)。本実施形態では、上述のようにフレーム長をサンプル列の開始時刻から終了時刻までの全区間に渡って固定長512サンプルとしている。相関フレーム検出手段90は、図11(a)に示すように、サンプル列の先頭から512サンプルずつを1フレームとして設定していくことになる。

20

【0058】

次に、各フレームに対して構成するサンプル値が全て一致するフレームを探索する。具体的には、図11(b)に示すように、まず、設定されたフレームのうち、時間的に最後尾のフレームを、相関フレームを探すための対象フレームとする。次に、所定の探索範囲内において、対象フレームの先頭サンプルの値と同一の値をもつサンプルを、時間的に遡りながら探索していく(ステップS32)。例えば、図12(a)に示すように、対象フレームが $mT \sim mT + 511$ の512個のサンプルで構成されているとする。この場合、まず、対象フレームの先頭サンプル mT のサンプル値 $e(mT)$ と同一となるサンプルを探索していく。サンプル $mT - 1$ 、サンプル $mT - 2$ と順に探索していく。なお、図12において、 m は先頭から m 番目のフレームであることを示し、 T はフレーム長(本実施形態では512サンプル)を示している。

30

【0059】

一致するサンプル t が見つかったら(ステップS33)、次に、そのサンプル t の次のサンプル $t + 1$ と対象フレームの2番目のサンプル $mT + 1$ が一致するかどうかを比較する。このようにしてサンプルの値が一致する限り後続するサンプル同士の比較を行っていく(ステップS34)。ステップS34においては、 $e(t + p)$ と $e(mT + p)$ の値が一致する限り、処理を繰り返していく。例えば、図12(b)に示す例では、 $e(t) \sim e(t + 8)$ が $e(mT) \sim e(mT + 8)$ と一致しているので、さらに $p = 9$ として、ステップS34の処理が続けられることになる。 $p = 0 \sim p = 511$ までの全ての $e(t + p)$ と $e(mT + p)$ が一致した場合(ステップS35)、そのサンプル列を対象フレームに対する相関フレームとし、相関フレームの先頭のサンプル番号と対象フレームの先頭のサンプル番号とを対応付けてフレーム相関データとして記録し、対象フレームを元のサンプル列から削除する(ステップS36)。対象フレームの全サンプルと一致しなければ、さらに対象フレームの先頭サンプルと値が一致するサンプルが存在するかどうかを時間的に遡りながら探索していく。所定のサンプル数分遡っても一致する相関フレームが存

40

50

在しない場合は、その対象フレームに関する相関フレームの探索を中止し、対象フレームの直前のフレームを新たな対象フレームとして相関フレームの探索を行う。1つの対象フレームに対しての処理が終わったら、ステップS32に戻って、1つ直前のフレームを新たな対象フレームとして処理を続けていく(ステップS37)。このようにして、時系列信号の先頭時刻近辺に位置するフレームを除く全フレームを対象フレームとして相関フレームの検出処理を行う。

【0060】

サンプル列全体でみると、図11(c)に示すように対象フレームに対応する相関フレームが検出されたとすると、図11(d)に示すように対象フレームが削除されることになる。このとき、復号時に完全に復元できるように図11(e)に示すようなフレーム相関データが記録される。図11(e)に示すように、フレーム相関データには対象フレームの先頭のサンプル番号と相関フレームの先頭のサンプル番号が対応づけて記録される。

10

【0061】

以上、本発明の好適な実施形態について説明したが、本発明は上記実施形態に限定されず、種々の変形が可能である。例えば、上記実施形態では、主サンプルと副サンプルの出現比率が1対1のものについて説明したが、これ以外の比率のものでも良い。上記実施形態では、最初に48kHzでサンプリングしたものを96kHzに補間して作成した音響信号を扱ったので、主サンプルと副サンプルの出現比率が1対1となったが、例えば、最初に24kHzでサンプリングしたものを96kHzに補間すると、主サンプルと副サンプルの出現比率は1対3となる。このような音響信号を扱う場合には、サンプル列再配置手段20が、各サンプルを順番に1つの主サンプル列と3つの副サンプル列に順に振り分けるようにすれば良い。

20

【0062】

【発明の効果】

以上、説明したように本発明によれば、時系列信号を構成するサンプル列に対して、録音により作成された主サンプル列と、主サンプル列を補間処理することにより得られた副サンプル列を分離すると共に、前記副サンプル列中の各副サンプルの値を、近傍の主サンプルの平均値と当該副サンプルの値との差分値に変換するサンプル再配置を行い、主サンプル列、変換された副サンプル列それぞれに対して、線形予測誤差を算出し、主サンプル列および副サンプル列の値をそれぞれ予測誤差値に変換し、予測誤差値に変換された主サン

30

【図面の簡単な説明】

【図1】本発明第1の実施形態に係る時系列信号の符号化装置を示す機能ブロック図である。

【図2】サンプル列再配置手段20によるサンプルの再配置の様子を示す図である。

【図3】予測誤差変換手段40による処理を示すフローチャートである。

【図4】チャンネル間演算手段50による処理を示すフローチャートである。

【図5】極性処理手段60によるビット構成の変換の様子を示す図である。

40

【図6】可変長符号化手段70による処理を示すフローチャートである。

【図7】上位ビットと下位ビットのデータ分離の様子を示す図である。

【図8】本発明第2の実施形態に係る時系列信号の符号化装置を示す機能ブロック図である。

【図9】信号平坦部処理手段80による処理の様子を示す図である。

【図10】相関フレーム検出手段90による処理を示すフローチャートである。

【図11】相関フレーム検出手段90の処理による時系列信号全体の様子を示す図である。

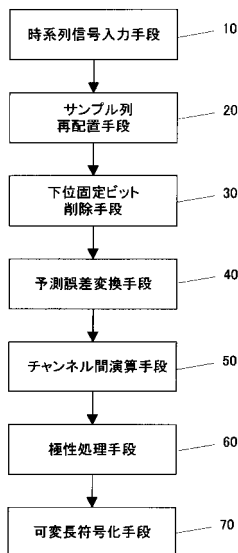
【図12】相関フレーム検出手段90の処理により比較されるサンプルの様子を示す図である。

50

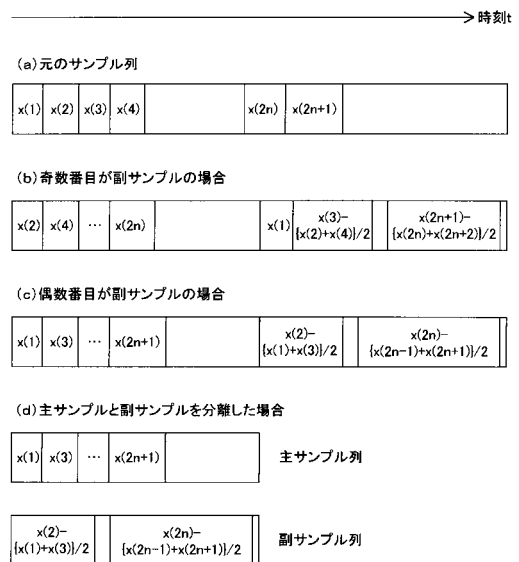
【符号の説明】

- 10・・・時系列信号入力手段
- 20・・・サンプル列再配置手段
- 30・・・下位固定ビット削除手段
- 40・・・予測誤差変換手段
- 50・・・チャンネル間演算手段
- 60・・・極性処理手段
- 70・・・可変長符号化手段
- 80・・・信号平坦部処理手段
- 90・・・相関フレーム検出手段

【図1】

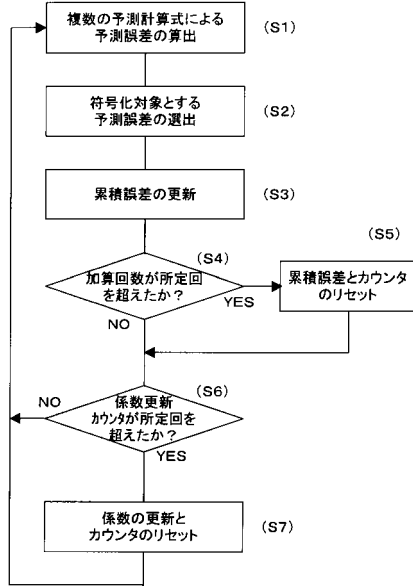


【図2】



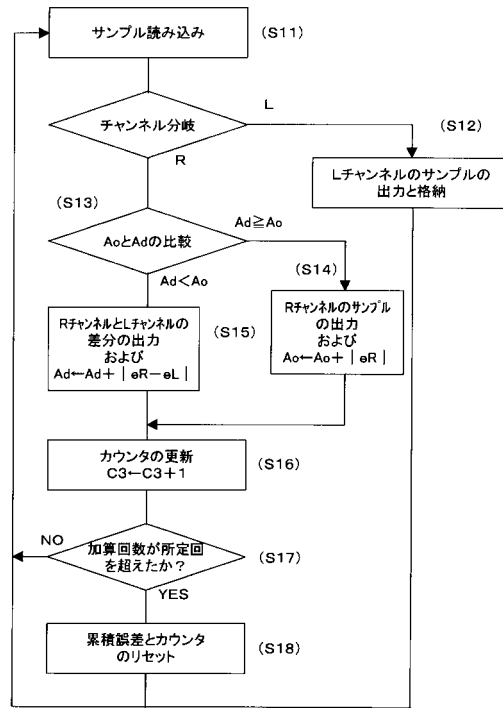
【 図 3 】

予測誤差変換

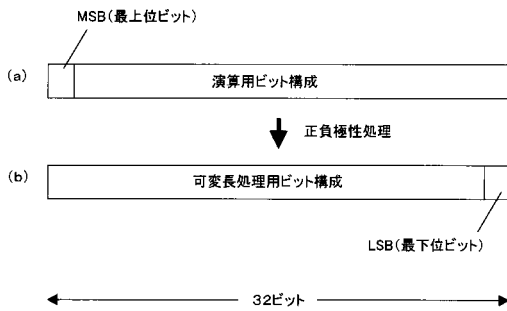


【 図 4 】

チャンネル間演算

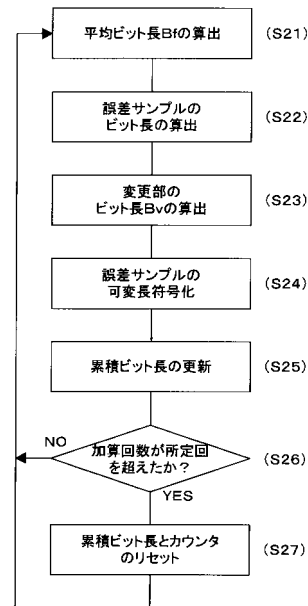


【 図 5 】



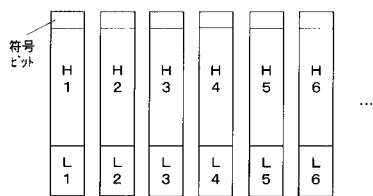
【 図 6 】

可変長符号化

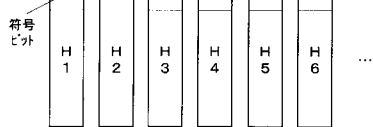


【 図 7 】

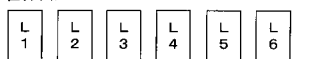
(a) サンプルデータ



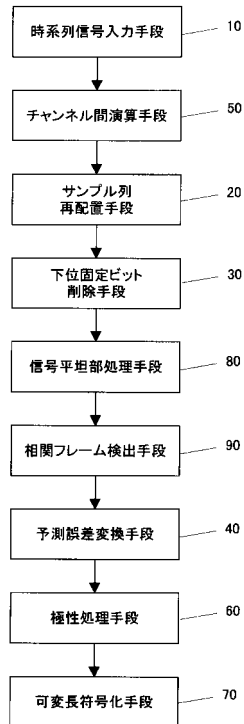
(b) 上位サンプル



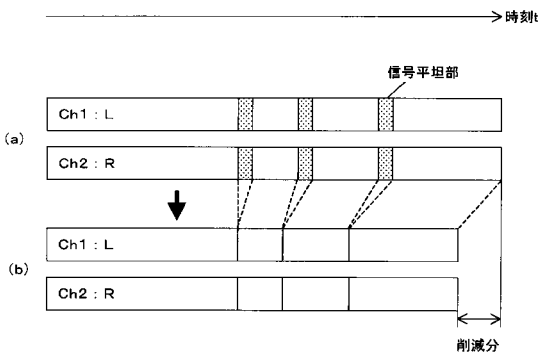
(c) 下位サンプル



【 図 8 】



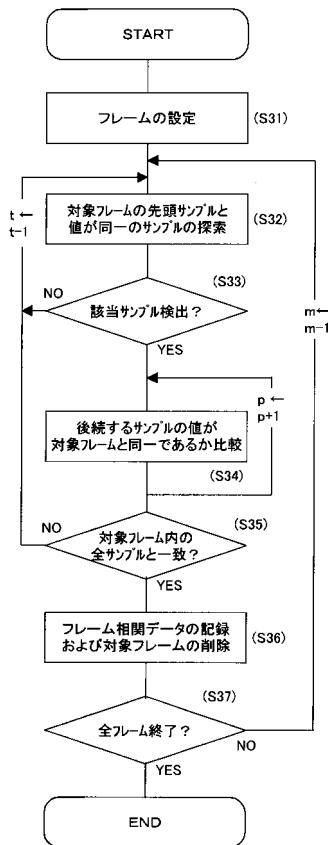
【 図 9 】



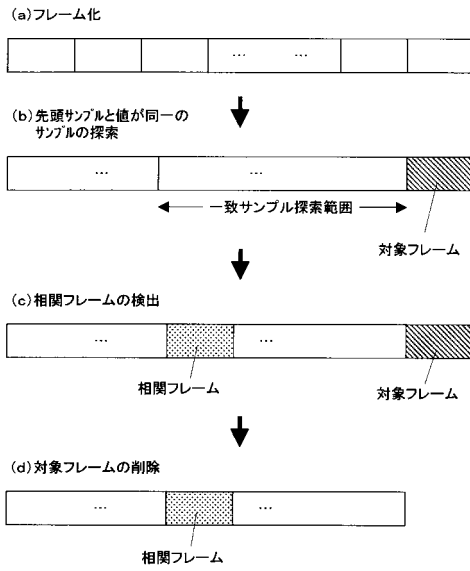
(c) 信号平坦部データ

	先頭時刻	サンプル数	サンプル値
平坦部①	291	18	0
平坦部②	385	20	32767
平坦部③	511	21	-32768
⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮

【 図 10 】



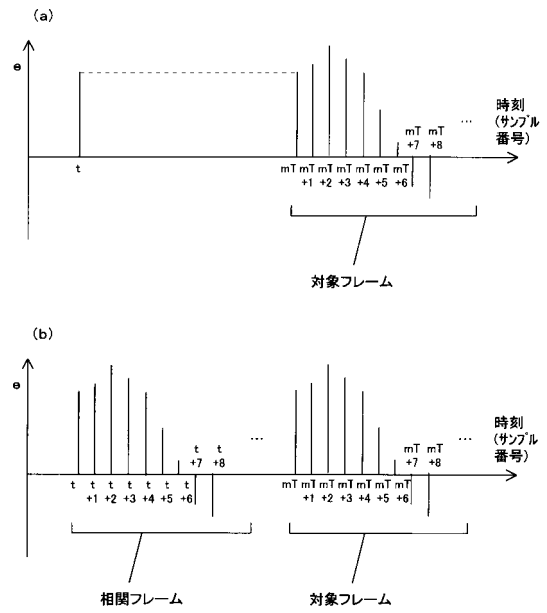
【 図 1 1 】



(e) フレーム相関データ

対象フレーム	相関フレーム
...	...
102401	68215
...	...
...	...

【 図 1 2 】



フロントページの続き

(51)Int.Cl.⁷

H 0 3 M 7/40

F I

G 1 0 L 9/18

M

テーマコード(参考)