

(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

特開昭63 - 285600

(43)公開日 昭和63年(1988)11月22日

(51)Int. Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 1 0 L	9/18			
H 0 4 L	23/00			
H 0 4 N	7/13			

審査請求 未請求 請求項の数 1

(全4頁)

(21)出願番号 特願昭62-120833

(22)出願日 昭和62年(1987)5月18日

(71)出願人 999999999  
日本電信電話株式会社  
東京

(72)発明者 守谷 健弘  
\*

(72)発明者 誉田 雅彰  
\*

(54)【発明の名称】ベクトル量子化器

## (57)【要約】

【産業上の利用分野】音声や画像の信号系列値をベクトルとし、予め貯えておいた符号帳の中から最も類似したベクトルを高速に探索することで符号化や照合を行う装置に関する

【目的】符号帳内のベクトルが要素毎に独立な乱数から構成される場合のベクトル量子化器において、歪計算における演算の回数を大幅に削減することにある

【効果】符号ベクトルの要素を並べかえて重複して利用することで符号帳の記憶容量を  $1/K$  ですますことができる

【特許請求の範囲】

【請求項1】入力ベクトルと符号帳中の符号ベクトルの歪計算と最小値探索を、入力ベクトルを高速フーリエ変換する手段と、符号レベルを高速逆フーリエ変換する手段または予め高速逆フーリエ変換された符号ベクトルを蓄積する手段と、変換された入力、符号両ベクトルの積のベクトルを作りそれを高速フーリエ逆変換する手段と、符号ベクトルの2乗和から逆変換されたベクトル中の最大要素の値の2倍を減算することで最小値を求め、逆変換されたベクトル中で最大要素をもつ要素の番号を歪最小を与える番号とする手段と、を使って実行することを特徴とするベクトル量子化器。

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

昭63-285600

⑬ Int.Cl.<sup>4</sup>

G 10 L 9/13  
H 04 L 23/00  
H 04 N 7/13

識別記号

庁内整理番号

E-8622-5D  
A-7240-5K  
Z-7060-5C

⑭ 公開 昭和63年(1988)11月22日

審査請求 未請求 発明の数 1 (全4頁)

⑮ 発明の名称 ベクトル量子化器

⑯ 特 願 昭62-120833

⑰ 出 願 昭62(1987)5月18日

⑱ 発 明 者 守 谷 健 弘 東京都武蔵野市緑町3丁目9番11号 日本電信電話株式会社基礎研究所内

⑲ 発 明 者 菅 田 雅 彰 東京都武蔵野市緑町3丁目9番11号 日本電信電話株式会社基礎研究所内

⑳ 出 願 人 日本電信電話株式会社 東京都千代田区内幸町1丁目1番6号

㉑ 代 理 人 弁理士 星野 恒司 外1名

明 細 書

1. 発明の名称 ベクトル量子化器

2. 特許請求の範囲

入力ベクトルと符号帳中の符号ベクトルの歪計算と最小値探索を、

入力ベクトルを高速フーリエ交換する手段と、

符号レベルを高速逆フーリエ変換する手段または予め高速逆フーリエ変換された符号ベクトルを蓄積する手段と、

変換された入力、符号両ベクトルの積のベクトルを作りそれを高速フーリエ逆変換する手段と、

符号ベクトルの2乗和から逆変換されたベクトル中の最大要素の値の2倍を減算することで最小値を求め、逆変換されたベクトル中で最大要素をもつ要素の番号を至最小を与える番号とする手段と、を使って実行することを特徴とするベクトル量子化器。

3. 発明の詳細な説明

(発明の属する技術分野)

本発明は、音声や画像の信号系列値をベクトルとし、予め貯えておいた符号帳の中から最も類似したベクトルを高速に探索することで符号化や照合を行う装置に関するものである。

(従来の技術)

通常、類似度を測るために2乗距離による歪、 $D_j$ の計算が実行され、その最小値を与えるベクトルの番号jを出力する。

$$D_j = \sum_{i=1}^k (X_i - C_{ij})^2 \quad (j = 0 \dots n-1) \quad (1)$$

ここで、 $X_i$ は入力系列の第i成分の値、 $C_{ij}$ は符号帳中のj番目のベクトルの第i成分の値である。kはベクトルの次元数(要素数)、nは符号帳中のベクトルの個数である。

符号化の場合( $\log_2 n$ )ビットが情報量となる。

減算して2乗することを1つの演算単位とすると、通常、1つの入力ベクトルに対してk・n回の演算を要する。演算量を大幅に削減するために、符号帳の探索を樹木状に行う方法が知られている。しかし、この方法では、量子化歪がかなり増加し

## 特開昭63-285600(2)

てしまうという欠点がある。

またこれとは別に、入力ベクトルおよび符号ベクトルのそれぞれの特徴を表わすパラメータを作り、まずそのパラメータを使って予備選択をして、その後、至最小のベクトルを選択する方法も知られている。この方法では、相関の強い系列に対する演算量が減少するが、一般的には必ずしも減少しないという欠点がある。

(発明の目的)

本発明の目的は、符号帳内のベクトルが要素毎に独立な乱数から構成される場合のベクトル量子化器において、至計算における演算の回数を大幅に削減することにある。

(発明の構成)

(発明の特徴と従来技術との差異)

本発明は、入力ベクトルと符号帳中のベクトルの至計算を、ベクトル間のたたみ込みとパワ計算に分離し、たたみ込み演算を高速フーリエ変換によって少ない演算量で実行することを最も主要な特徴とする。

(4)、(5)の場合(3)式の右辺第3項は入力 $X_i$ と系列 $q$ ：のたたみ込み、あるいは巡回たたみ込みと等価になることを利用すれば、演算量を減少させることができる。ただし、「 $\lceil$ 」は小数以下切り捨て、 $[ ] \bmod k$ は $k$ で割算した余りである。

(実施例)

第4回は高速に至計算を第3回の形式の符号帳を使って行うときの処理手順を示すものである。この場合、 $k$ 次元の符号帳ベクトル $n$ 個の中から入力に近いものを探すか、簡単のため $k$ は2のべき乗であるとし、 $n$ は $k$ でわりきれれるとする。

( $L = n/k$ ) $n$ 個の要素からなる $q$ の系列を $L$ 個の $k$ 次元部分符号ベクトル $Q_i$ に分割する。まず、1番目の部分符号ベクトル $Q_1(q_1 \dots q_k)$ を使って $k$ 回の至計算 $d_0 \sim d_k$ を行う。

$$d_0 = \sum_{i=0}^{k-1} (X_i - q_i)^2$$

$$= \sum_{i=0}^{k-1} X_i^2 + \sum_{i=0}^{k-1} q_i^2 - 2 \sum_{i=0}^{k-1} X_i q_i$$

$$d_1 = \sum_{i=0}^{k-1} (X_i - q_{[i+1]_{n \dots k}})^2$$

ここで至 $D$ を次のように置き換える。

$$D_j = \sum_{i=0}^{k-1} (X_i - C_{ij})^2 \quad (j=0 \dots n-1) \quad (2)$$

$$= \sum_{i=0}^{k-1} X_i^2 + \sum_{j=0}^{n-1} C_{ij}^2 - 2 \sum_{j=0}^{n-1} X_i C_{ij} \quad (j=0 \dots n-1) \quad (3)$$

通常のベクトル量子化では、第1図のように符号帳中のベクトル $c_{ij}$ は $j$ 毎に計 $n \cdot k$ 個の要素からなる。

ところが、 $C_{ij}$ が独立で記憶のない乱数で十分である時には第2図、第3図のように一次元乱数系列 $q_i$ を重複して利用することができる。

第2図は1要素ずつらせることで次の新たな符号ベクトルとするものである。すなわち、 $C_{ij}$ を次のようにおきかえる。

$$C_{ij} = q_{[i+j]_{n \dots k}} \quad (4)$$

第3図は $q_i$ の系列を入力ベクトルと同じ要素数をもつ $n/k$ 個の部分ベクトルに分割し、各部分ベクトル内の要素を巡回的に符号ベクトルとして利用するものである。この時

$$C_{ij} = q_{[i/k] \cdot k + [i+j] \bmod k} \quad (5)$$

$$= \sum_{i=0}^{k-1} X_i^2 + \sum_{i=0}^{k-1} q_i^2 - 2 \sum_{i=0}^{k-1} X_i q_{[i+1]_{n \dots k}}$$

$$= \sum_{i=0}^{k-1} X_i^2 + \sum_{i=0}^{k-1} q_i^2 - 2 \{ (X_0 \cdot q_1) + (X_1 \cdot q_2) + \dots + (X_{k-1} \cdot q_0) \}$$

$$d_{k-1} = \sum_{i=0}^{k-1} (X_i - q_{[i+k-1]_{n \dots k}})^2$$

$$= (X_0 - q_{k-1})^2 + (X_1 - q_0)^2 + \dots + (X_{k-1} - q_{k-2})^2$$

$$= \sum_{i=0}^{k-1} X_i^2 + \sum_{i=0}^{k-1} q_i^2 - 2 \{ (X_0 \cdot q_{k-1}) + (X_1 \cdot q_0) + \dots + (X_{k-1} \cdot q_{k-2}) \}$$

そして $d_0 \sim d_{k-1}$ の中から最小値を与える $d^*$ と

その番号をそれぞれ $d_1^*$ 、 $I_1$ と記憶する。

同様に $Q_2 \dots Q_L$ に対して $d_2^*$ 、 $I_2 \sim d_1^*$ 、 $I_L$ を求める。最終的に $d_1^*$ から $d_2^*$ の中の最小値とその番号を求めれば計 $n$ 種類の符号の中から $X_i$ に最も近いものが求まったことになる。

このうち各 $Q_i$ での至計算と最小値探索を高速に行う手順を示したのが第5図である。

$X$ のフーリエ変換を $u$ 、 $q$ の逆フーリエ変換を $v$ とする。 $u$ 、 $v$ の要素ごとの積をとり $g$ とし、これを逆フーリエに変換して $f$ を求める。ここでフーリエを変換と巡回たたみこみ性質から $f$ は次のように書ける。

特開昭63-285600(3)

$$f_0 = k \sum_{i=0}^{k-1} X_i q_i$$

$$f_1 = k \sum_{i=0}^{k-1} X_i q_{[i+1]_{\text{mod } k}}$$

⋮

$$f_{k-1} = k \sum_{i=0}^{k-1} X_i q_{[i+k-1]_{\text{mod } k}}$$

従って、

$$d_0 = \sigma_x^2 + \sigma_q^2 - 2(f_0/k)$$

$$d_1 = \sigma_x^2 + \sigma_q^2 - 2(f_1/k)$$

⋮

$$d_{k-1} = \sigma_x^2 + \sigma_q^2 - 2(f_{k-1}/k)$$

ここで、 $\sigma_x^2 = \sum_{i=0}^{k-1} X_i^2$

$$\sigma_q^2 = \sum_{i=0}^{k-1} q_i^2$$

すなわち、 $d_i (i = 0 \dots k-1)$ の最小値を与える  $i$  は  $f_i$ の最大値を与える  $i$ であり、最小値は上式で求まる。また  $\sigma_x^2$ の計算は省略できる。ここで、加減算と2乗、加減算と乗算の組み合わせを1演算単位として演算量を評価する。

をベースとし、かつ複素数の積が実数の積の4倍の演算量を必要とする場合でも十分有効である。

第6図は第2図の形式の符号幅を用いる場合の第2の実施例である。

ここでも第1の実施例と同様に歪計算をパワの計算とたたみこみの計算に分離して、たたみこみ計算を高速に行う。第1の実施例のたたみこみは巡回たたみこみであったのに対し、第2の実施例では通常のたたみこみになっている。

すなわち、入力ベクトル  $X$ 、符号ベクトル  $C$ の双方に  $k$ 次元の0をつめたベクトルをつけて、 $2k$ 点の巡回たたみこみを行い、結果のベクトルを半分ずつ重ね合わせることで通常のたたみこみを実現している。

第6図中の太い線は実施例1の巡回たたみこみをフーリエ変換を介して行う処理を示す。

これらの手順は系列  $q$ に  $X$ という計数のFIRフィルタをかける手順と等価である。演算量は第1の実施例の場合の約2倍となる。

また、歪計算に加重が加わる場合、すなわち歪

	符号ベクトル比	全体
$\sigma_x^2$	$k$	$n$
変換後の積	$k$	$n$
高速フーリエ変換 2回	$2k \log_2 k$	$2n \log_2 k$
低速フーリエ変換 1回	—	$k \log_2 k$

全体で  $(k \log_2 k) + 2n(1 + \log_2 k)$

と評価できる。ここで

$\sigma_x^2$ や  $q$ の逆フーリエ変換は予め演算して記憶しておいてもよいことを考慮するとさらに、

$$n + (k + n)(\log_2 k)$$

これを通常の歪計算の場合の  $kn$ と比較すると、 $k$ が大きいときに大幅に演算量が減少することがわかる。例えば、10ビットで  $n = 1024$ 、 $k = 128$ 次元とすると約1/14の演算量ですむ。

高速フーリエ変換は、複素数をベースとしてもよいし、整数を使った有限体をベースとしてもよい。また、フーリエ変換の代わりにウオルシュワグマル変換を用いて同様の演算が可能である。ただしこの場合、符号ベクトル中の要素の巡回の方法(並べかえ方法)が異なってくる。また複素数

$D_j$ が

$$D_j = \sum_{i=0}^{k-1} w_i (X_i - C_{j,i})^2$$

と定義される場合でも同様に応用できる。

$$D_j = \sum_{i=0}^{k-1} w_i X_i^2 + \sum_{i=0}^{k-1} w_i C_{j,i}^2 - 2 \sum_{i=0}^{k-1} w_i X_i C_{j,i}$$

で第1項は  $j$ によらず一定、第2項は  $C_{j,i}^2$ と  $w_i$ のベクトルのたたみこみ、第3項は  $w_i \cdot X_i$ (成分毎の積)ベクトルと  $C_{j,i}$ のたたみこみである。

従って、第2項、第3項の演算にそれぞれ実施例1の演算を適用すれば高速化できる。この場合の演算量は実施例1の約2倍となるが、もともと加重つきの歪計算は通常の2倍の乗算を要するため、改善効果は実施例1の場合と同等である。

(発明の効果)

以上説明したように、 $k$ 次元のベクトルと  $n$ 種類の符号ベクトルの歪計算を  $\{n + (k + n)(\log_2 k)\}$ 回の演算で実行できる。

特に  $k$ が32程度以上で大きくなるほど通常  $kn$ 回と比較して大幅に演算量の削減ができるという利点がある。

特開昭63-285600(4)

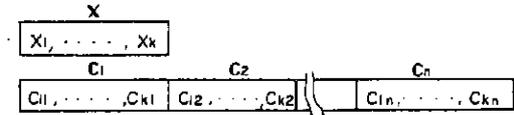
ただし、符号ベクトルの内容は独立な(相関のない)系列である場合に限定される。実際の音声や画像の符号化においては、予測残差の量子化や、量子化誤差を再び量子化する場合は乱数を符号帳とすることで十分よい性質が得られることが知られている。このため、本発明が有効に利用できる。

また、付随的ではあるが、符号ベクトルの要素を並べかえて重畳して利用することで符号帳の記憶容量を  $1/k$  で済ますことができるという利点もある。

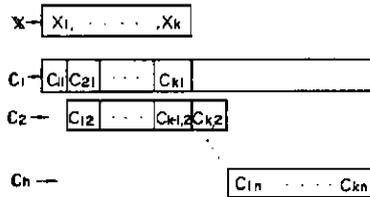
4. 図面の簡単な説明

- 第1図は通常の入力ベクトルと符号ベクトルを示す図、
- 第2図は符号ベクトルをずらせながら利用する場合を示す図、
- 第3図は符号ベクトルを部分ベクトルに分割し、その中で要素を巡回させて利用する場合を示す図、
- 第4図は歪計算と最小値探索の手順を示す図、
- 第5図は部分ベクトル内での高速歪計算の手順を示す図、
- 第6図は符号帳の要素を1個ずつずらせて用いる場合の実施例を示す図である。

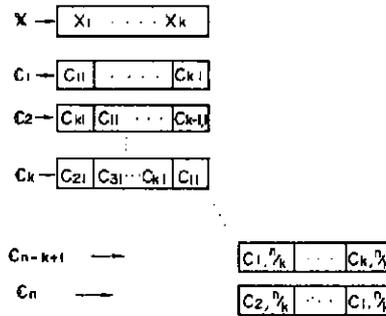
第 1 図



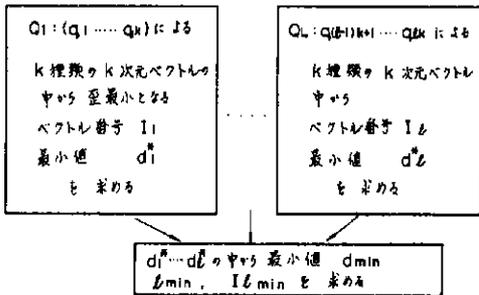
第 2 図



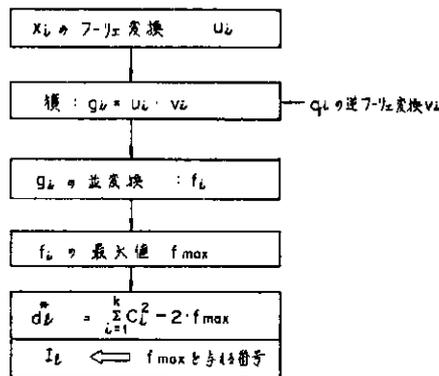
第 3 図



第 4 図



第 5 図



第 6 図

