

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4987086号
(P4987086)

(45) 発行日 平成24年7月25日(2012.7.25)

(24) 登録日 平成24年5月11日(2012.5.11)

(51) Int.Cl. F I
HO4N 7/32 (2006.01) HO4N 7/137 Z

請求項の数 12 (全 20 頁)

(21) 出願番号	特願2009-539030 (P2009-539030)	(73) 特許権者	000004226
(86) (22) 出願日	平成20年10月23日(2008.10.23)		日本電信電話株式会社
(86) 国際出願番号	PCT/JP2008/069257		東京都千代田区大手町二丁目3番1号
(87) 国際公開番号	W02009/057506	(74) 代理人	100064908
(87) 国際公開日	平成21年5月7日(2009.5.7)		弁理士 志賀 正武
審査請求日	平成22年1月12日(2010.1.12)	(74) 代理人	100108453
(31) 優先権主張番号	特願2007-281556 (P2007-281556)		弁理士 村山 靖彦
(32) 優先日	平成19年10月30日(2007.10.30)	(72) 発明者	高村 誠之
(33) 優先権主張国	日本国(JP)		東京都千代田区大手町二丁目3番1号 日 本電信電話株式会社内
		(72) 発明者	八島 由幸
			東京都千代田区大手町二丁目3番1号 日 本電信電話株式会社内
		審査官	坂本 聡生

最終頁に続く

(54) 【発明の名称】 画像符号化方法及び復号方法、それらの装置、それらのプログラム並びにプログラムを記録した記録媒体

(57) 【特許請求の範囲】

【請求項1】

符号化対象の画素の値を、復号済み画像から空間予測または時間予測により生成される予測値を用いて符号化する画像信号の符号化方法であって、

符号化対象の画素の値の予測を行い、予測値を得るステップと、

前記得られた予測値に対し、原画素値がどのような値をとっているのかという確率分布のデータを、予め蓄積されている予測符号化における原画素値と予測値との差分の分布データを予測値に応じてシフトさせることにより算出するステップと、

前記得られた確率分布のデータを原画素値が取り得る下限値から上限値までの範囲にクリッピングするステップと、

前記クリッピングされた原画素値の下限値から上限値までの確率分布のデータを用いて、符号化対象の画素の値を符号化するステップとを有する

ことを特徴とする画像符号化方法。

【請求項2】

請求項1に記載の画像符号化方法であって、

前記符号化対象の画素の値が所定の画素ブロックに対応づけられるように、前記予測を画素ブロック単位で行い、

前記画素値を符号化するステップでは、ブロック予測による画素ブロック値に対して求められた前記確率分布のデータをもとに、量子化代表ベクトルを定め、前記画素ブロック値をベクトル量子化することにより符号化する

ことを特徴とする画像符号化方法。

【請求項3】

復号対象の画素の値を、復号済み画像から空間予測または時間予測により生成される予測値を用いて復号する画像信号の復号方法であって、

復号対象の画素の値の予測を行い、予測値を得るステップと、

前記得られた予測値に対し、原画素値がどのような値をとっているのかという確率分布のデータを、予め蓄積されている予測符号化における原画素値と予測値との差分の分布データを予測値に応じてシフトさせることにより算出するステップと、

前記得られた確率分布のデータを原画素値が取り得る下限値から上限値までの範囲にクリッピングするステップと、

前記クリッピングされた原画素値の下限値から上限値までの確率分布のデータを用いて、復号対象の画素の値を復号するステップとを有する

ことを特徴とする画像復号方法。

【請求項4】

請求項3に記載の画像復号方法であって、

前記復号対象の画素の値が所定の画素ブロックに対応づけられるように、前記予測を画素ブロック単位で行い、

前記画素値を復号するステップでは、ブロック予測による画素ブロック値に対して求められた前記確率分布のデータをもとに量子化代表ベクトルを定め、復号対象のベクトル量子化された画素ブロック値を、前記定められた量子化代表ベクトルに従って復号する

ことを特徴とする画像復号方法。

【請求項5】

符号化対象の画素の値を、復号済み画像から空間予測または時間予測により生成される予測値を用いて符号化する画像信号の符号化装置であって、

符号化対象の画素の値の画素値予測を行い、予測値を得る予測手段と、

前記得られた予測値に対し、原画素値がどのような値をとっているのかという確率分布のデータを、予め蓄積されている予測符号化における原画素値と予測値との差分の分布データを予測値に応じてシフトさせることにより算出する手段と、

前記得られた確率分布のデータを原画素値が取り得る下限値から上限値までの範囲にクリッピングする手段と、

前記クリッピングされた原画素値の下限値から上限値までの確率分布のデータを用いて、符号化対象の画素の値を符号化する符号化手段とを備える

ことを特徴とする画像符号化装置。

【請求項6】

請求項5に記載の画像符号化装置であって、

前記予測手段は、前記符号化対象の画素の値が所定の画素ブロックに対応づけられるように、前記予測を画素ブロック単位で行い、

前記符号化手段は、ブロック予測による画素ブロック値に対して求められた前記確率分布のデータをもとに、量子化代表ベクトルを定め、前記画素ブロック値をベクトル量子化することにより符号化する

ことを特徴とする画像符号化装置。

【請求項7】

復号対象の画素の値を、復号済み画像から空間予測または時間予測により生成される予測値を用いて復号する画像信号の復号装置であって、

復号対象の画素の値の画素値予測を行い、予測値を得る予測手段と、

前記得られた予測値に対し、原画素値がどのような値をとっているのかという確率分布のデータを、予め蓄積されている予測符号化における原画素値と予測値との差分の分布データを予測値に応じてシフトさせることにより算出する手段と、

前記得られた確率分布のデータを原画素値が取り得る下限値から上限値までの範囲にクリッピングする手段と、

10

20

30

40

50

前記クリッピングされた原画素値の下限値から上限値までの確率分布のデータを用いて、復号対象の画素の値を復号する復号手段とを備えることを特徴とする画像復号装置。

【請求項 8】

請求項 7 に記載の画像復号装置であって、

前記予測手段は、前記復号対象の画素の値が所定の画素ブロックに対応づけられるように、前記予測を画素ブロック単位で行い、

前記復号手段では、ブロック予測による画素ブロック値に対して求められた前記確率分布のデータをもとに量子化代表ベクトルを定め、復号対象のベクトル量子化された画素ブロック値を、前記定められた量子化代表ベクトルに従って復号する

ことを特徴とする画像復号装置。

10

【請求項 9】

請求項 1 に記載された画像符号化方法を、コンピュータに実行させるための画像符号化プログラム。

【請求項 10】

請求項 3 に記載された画像復号方法を、コンピュータに実行させるための画像復号プログラム。

【請求項 11】

請求項 1 に記載された画像符号化方法を、コンピュータに実行させるための画像符号化プログラムを記録したコンピュータ読み取り可能な記録媒体。

20

【請求項 12】

請求項 3 に記載され画像復号方法を、コンピュータに実行させるための画像復号プログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像（静止画像もしくは映像（動画像））を効率よく符号化・復号する高能率画像符号化方式の技術に関する。

本願は、2007年10月30日に出願された特願2007-281556号に基づき優先権を主張し、その内容をここに援用する。

30

【背景技術】

【0002】

画像（静止画像もしくは映像）の符号化においては、既に復号された画素を用いて、符号化対象の画素値を、空間方向の予測や時間方向の予測方法により予測する予測符号化方式が主流である。

【0003】

例えば H.264 / AVC では、4 × 4 ブロックのイントラ水平予測において、符号化対象の A から P まで（以下、A...P と表記する。他も同様）の 4 × 4 ブロックを、既に復号済みの左隣の画素 a...d を使い、

40

a		A	B	C	D
b		E	F	G	H
c		I	J	K	L
d		M	N	O	P

すなわち、

A = B = C = D = a

E = F = G = H = b

I = J = K = L = c

M = N = O = P = d

のように水平方向に予測し、次いで予測残差を、

50

A - a B - a C - a D - a
 E - b F - b G - b H - b
 I - c J - c K - c L - c
 M - d N - d O - d P - d

のように計算し、その後直交変換・量子化・エントロピ符号化を施して圧縮符号化することが行われている。

【0004】

動き補償予測においても同様に、 4×4 ブロックの動き補償では、 $A \dots P$ を別フレームにより予測した 4×4 ブロック $A \dots P$ を

A B C D
 E F G H
 I J K L
 M N O P

10

のように作成し、次いで予測残差を、

A - A B - B C - C D - D
 E - E F - F G - G H - H
 I - I J - J K - K L - L
 M - M N - N O - O P - P

と計算し、その後直交変換・量子化・エントロピ符号化を施して圧縮符号化することが行われている。

20

【0005】

復号器は、復号済みの画素値から、例えばブロック左上の位置については、予測値 A および予測残差の復号値($A - A$)を得て、それらの和として原画素値 A を得る。これは可逆な場合であるが、非可逆の場合にも、復号器は予測残差復号値($A - A +$) (は符号化雑音)を得て、予測値 A との和として($A +$)を得る。

【0006】

以上は画素値が 4×4 の16個の場合であったが、以下、概念的に単純化し次元で考える。また以下、一般的な8bit画素値の場合を考える。画素値は0から255までの256通りの整数値をとる。なお、8bit以外の画素値についても、以降の議論は同様に適用できる。

30

【0007】

符号化対象の画素値を x とし、予測値を x とする。 x は x の値に近い値のため、予測残差($x - x$)は-255...255の値を取り得るが、0に近い値に集中し、絶対値の大きな値は相対的に少ない。この関係をグラフにより図1に示す。

【0008】

偏りのある分布の情報量は、一様な分布の情報量より小さいため、符号化後圧縮が期待できる。従来は、この分布の偏りを利用し、高効率圧縮を実現していた。

【0009】

なお、非特許文献1には、本発明の実施例の説明で用いるベクトル符号化において、代表ベクトルが空間内に規則的に位置するピラミッドベクトル量子化の技術が記載されている。

40

また、非特許文献2には、ベクトル量子化において代表ベクトルを学習により最適化し、結果としてそれらを空間的に不規則に配置するLBGアルゴリズムに基づくベクトル量子化の技術が記載されている。

【非特許文献1】T.R.Fischer, "A pyramid vector quantizer", IEEE Trans. Inform. Theory, vol.IT-32, no.4, pp.568-583, July, 1986.

【非特許文献2】Y.Linde, A.Buzo and R.M.Gray, "An algorithm for vector quantizer design," IEEE Trans. on Communications, vol.com-28, no.1, pp.84-95, Jan., 1980

【発明の開示】

【発明が解決しようとする課題】

50

【 0 0 1 0 】

従来の方法において、例えば予測値が $x = 255$ であったとする。画素値 x は、 $0 \dots 255$ の値であるので、予測残差は $x - x = -255 \dots 0$ と、 0 以下の値しか取り得ない。

【 0 0 1 1 】

したがって、予測残差分布において、正方向のほぼ右半分の分布は使われることがない。分布の端（生起確率が極めて低い）を無視して定性的に述べれば、分布は左右対称であるので「右か左のどちらか」という情報に 1 bit （例えば 0 が右、 1 が左）を要する。右半分の分布が使われない（とり得る値の範囲からはみだす部分がある）場合、その 1 bit は元来不要である。予測値が $x = 0$ の場合も、予測誤差分布の負方向のほぼ左半分は使われることがなく、同様に 1 bit が元来不要となる。

10

【 0 0 1 2 】

これらの関係を図 2 に示す。なお、以下の図において、画素値または予測誤差などの値が取り得る範囲を図中に斜線を付けて表す。

【 0 0 1 3 】

定量的に述べれば、まず左右に広がっている確率分布を $p_w(d)$ とする。

【 0 0 1 4 】

【数 1】

$$\sum_{d=-255}^{255} p_w(d) = 1 \quad (1)$$

20

【 0 0 1 5 】

である。実際には右半分の値は取り得ないので、誤差 d の正しい分布は p_w の二倍となる。

【 0 0 1 6 】

$$\begin{aligned} p_c(d) &= 2 p_w(d) && (d \leq 0 \text{ のとき}) \\ p_c(d) &= 0 && (d > 0 \text{ のとき}) \end{aligned} \quad (2)$$

【 0 0 1 7 】

【数 2】

$$\sum_{d=-255}^{255} p_c(d) = \sum_{d=-255}^0 p_c(d) = 1 \quad (3)$$

30

【 0 0 1 8 】

生起確率を p_w としたときの平均エントロピ H_w を見積もると、次のようになる。

【 0 0 1 9 】

【数 3】

$$H_w = - \sum_{d=-255}^0 p_c(d) \log_2 p_w(d) \quad (4)$$

40

【 0 0 2 0 】

正しい生起確率を用いた場合の平均エントロピは、以下のとおりである。

【 0 0 2 1 】

【数4】

$$-\sum_{d=-255}^0 p_c(d) \log_2 p_c(d) = -\sum_{d=-255}^0 p_c(d) \log_2 (2p_w(d)) \quad (5)$$

$$= -\sum_{d=-255}^0 p_c(d) (\log_2 p_w(d) + 1) \quad (6)$$

$$= -\sum_{d=-255}^0 p_c(d) \log_2 p_w(d) - \sum_{d=-255}^0 p_c(d) \quad (7)$$

$$= H_w - 1 \quad (8)$$

10

【0022】

しかしながら、従来は差分値 (x - x') のみを対象として符号化を行っていたので、この不要な 1 b i t の削減が不可能であった。

【0023】

この理由として、

- ・ (x - x') からは予測値 (x') の情報が失われていること、
 - ・ (多次元の場合であるが) 差分値 (x - x') に直交変換を施すため、変換後の空間における x の取り得る範囲も同時に変換したものを考慮しなければならず、多次元分布における「(取り得る値の範囲からの) はみだし」の判定が極めて困難となること、
- が挙げられる。

20

【0024】

以上のような問題に鑑み、本発明は、従来と同様の時間・空間的予測は行うが、原画素値と予測値との差分を求めずに、上述のような分布に関する「はみだし」を考慮した原画素値に対応した分布により原画素値を符号化することにより、予測符号化における符号化効率を改善することを目的とする。

【課題を解決するための手段】

【0025】

本発明は、上記課題を解決するため、符号化対象の画素の値 (画素ブロックに対応づけられる値の場合もある) を、復号済み画像から空間予測や時間予測 (動き補償) により生成される予測値を用いて符号化する予測符号化において、画素値の取り得る上下限値を考慮した、画素値の該予測値における条件付き分布を用いて、符号化対象画素 (または画素ブロック) の値を、符号化することを主要な特徴とする。

30

【0026】

画素値の取り得る上下限とは、デジタル画像における画素の取り得る値の上限と下限である。もっとも広範に用いられている 8 b i t 画像では、それぞれ 255 と 0 であり、10 b i t 画像では、1023 と 0 となる。

この上限より大きな値 (2000 など) や下限より小さい値 (-1) などを取るような画素は、原画像に存在していないと仮定して問題ない。上下限値の考慮とはこのことを意味しており、本発明はこのような事実を用いて符号化を効率化している。

40

【0027】

画素値の該予測値における条件付き分布とは、予測値が x' であるような画素において、原画素値 x が実際にはどのような値をとっているのかという確率分布である。

条件付きとは、ここでは予測値が x' であるということと等価である。

数学では、 $P_r(x | x')$ と書かれる。通常、この形状は x' をピークとする釣鐘状となる。

【0028】

予測値が x' であるという条件の下での x の分布も、(当然ながら) そのような条件がない場合の x の分布も、常に画素の上下限内の値 (8 b i t 画像では 0 から 255 の整数

50

値)しか取らない。

【0029】

また、本発明は、上記発明において予測を画素ブロック単位で行う場合に、ブロック予測による画素ブロック値の条件付き分布の符号化にベクトル量子化を用いることが可能である。

【発明の効果】

【0030】

本発明によれば、従来の、予測値と原画素値の差分を扱う場合の「予測値という重要な情報の欠落」を発生させることなく、余すことなく符号化に利用し、結果としてより少ない符号量で画像(静止画像もしくは映像)を符号化することができるようになる。

10

【図面の簡単な説明】

【0031】

【図1】画素値差分の分布を示す図である。

【図2】予測値が255の場合の画素値差分の分布を示す図である。

【図3】単純なベクトル量子化の場合(差分あり)を示す図である。

【図4】単純なベクトル量子化の場合(差分なし)を示す図である。

【図5】Lノルム=4に対応する代表ベクトルを示す図である。

【図6】Lノルム=4に対応する代表ベクトル(異なる予測値の例)を示す図である。

【図7】ピラミッドベクトル量子化の場合(差分あり)を示す図である。

【図8】ピラミッドベクトル量子化の代表ベクトルの計数を示す図である。

20

【図9】LBGアルゴリズムによる差分ベクトル量子化の模式図である。

【図10】特定の予測値(x_1 , x_2)における差分ベクトル量子化の模式図である。

【図11】特定の予測値(x_1 , x_2)における、本発明による手法によるベクトル量子化の模式図である。

【図12】本願発明の実施形態における符号化時の処理の流れを示す図である。

【図13A】原画素値の生起確率が一樣である概念を示す図である。

【図13B】原画素値と予測値の差分分布を示す概念図である。

【図13C】差分分布を原画素値に対応するものにシフトさせた分布を示す概念図である。

30

【図13D】原画素値のとり得る値にクリッピングした分布を示す概念図である。

【図14】本実施形態における符号化装置のブロック図である。

【図15】本実施形態における復号時の処理の流れを示す図である。

【図16】代表ベクトルのはみだし個数が正確に求まらない例を示す図である。

【符号の説明】

【0032】

300 信号端子

301 画素値予測器

302 予測値蓄積メモリ

303 差分分布蓄積メモリ

40

304 加算・クリッピング器

305 代表ベクトル設計器

306 原画素値蓄積メモリ

307 ベクトル量子化器

308 量子化インデックス符号化器

309 出力端子

【発明を実施するための最良の形態】

【0033】

まず、本発明の概要を具体例かつ簡単に説明する。

【0034】

50

例えば、信号 d が $\{-2, -1, 1, 2\}$ の 4 通りのいずれをとるかわからない場合 (25% で等確率を仮定)、この信号を符号化するには 2 [bit] が必要である。これを確率分布で表したのが図 1 (二次元の例は図 9) である。

【0035】

もし「信号 d が正である」とわかっているとすると、 $\{1, 2\}$ の二通りしか可能性がないので 1 [bit] で符号化ができてしまう。これを確率分布で表したのが図 2 (二次元の例は図 11) である。

【0036】

同じことが、静止画像もしくは映像の予測符号化でも起きる。

画像信号 x ($0 \leq x$) の予測値が x_0 であったとすると、予測誤差 $d = x - x_0$ の分布は予測値 x_0 により変化する。

例えば、もし $x_0 = 0$ であれば、 $0 \leq d$ 。つまり d は負の値をとらない。一方、 $x_0 = 2.55$ であれば、 $d \leq 0$ 。つまり d は正の値をとらない (概念は同じく図 2 を参照)。

【0037】

このように、予測値 x_0 の値を手がかりに、符号化・復号の前に d の存在範囲を絞り込んでおくことができる。これはとりもなおさず符号化効率の改善につながる。

d の存在範囲を絞り込む処理は、 $x_0 + d$ の範囲を (8 bit 画像の場合) $0 \dots 255$ に規格化することと等価である。

これは符号化方法では、図 12 のフローチャートのクリッピングステップ 105 に、復号方法では、図 15 のフローチャートのクリッピングステップ 405 に相当する。

【0038】

さらに、予測単位のブロック毎に代表ベクトルを設計することで、適応的に最適な代表ベクトルを設計できる (符号化方法では図 12 のフローチャートの代表ベクトル設計ステップ 106 に、復号方法では図 15 のフローチャートの代表ベクトル設計ステップ 406 に相当する)。

【0039】

こうして符号化対象信号を狭めた後、通常の符号化を経て従来よりも短い (符号化効率の高い) 符号が出力される (図 12 のフローチャートの量子化インデックス符号化ステップ 109)。

【0040】

画素値の予測は、既存の符号化技術 MPEG-1、MPEG-2、MPEG-4、H.264/AVC において「動き補償」あるいは「イントラ予測」として用いられており (符号化方法では図 12 のフローチャートのステップ 101、復号方法では図 15 のフローチャートのステップ 401)、本発明はこのような予測が用いられる場面に広範に適用でき、符号化効率の改善に寄与するものである。

【0041】

なお、既存の予測符号化技術においては、予測誤差は常に正負いずれも取り得るものとして符号化している (概念図は図 9)。

【0042】

次に、本発明の実施形態における原理的な仕組みについて詳細に説明する。

予測を画素ブロック単位で行い、ブロック予測による画素ブロック値の条件付き分布の符号化にベクトル量子化を用いる例を説明するが、予測値が x_0 であるような画素において、原画素値 x が実際にはどのような値をとっているのかという確率分布を用いて符号化を行うという基本的な考え方は、画素単位で符号化する場合も同様である。

【0043】

[距離尺度が L_1 ノルムの場合]

図 3 を用い、二次元での量子化・符号化を例にとり説明する。これは二次元の最も単純な格子量子化の模式図である。原点からの距離尺度は、 L_1 ノルム (座標絶対値の最大値) に基づき計算される。

例えば点 $(0, 3)$ と点 $(-2, -3)$ は、同一の L_1 ノルムを持つ。

10

20

30

40

50

図中、点線でつながれた点（ベクトル量子化後、離散データが属する代表ベクトルである）同士が同一のL ノルムを持ち、同一ノルムの点同士は同確率で生起することを仮定する。

【0044】

この図では、隣り合う二画素の原画素値がそれぞれ $x_1 = 253$ 、 $x_2 = 102$ であり、予測値が $x_1 = 255$ 、 $x_2 = 100$ であったとしている。差分値は $x_1 - x_1 = -2$ 、 $x_2 - x_2 = 2$ となる。

【0045】

原画素値に対応する差分ベクトル $(-2, 2)$ が属するL ノルム $= 2$ の生起確率が、 0.3 であったとする。

L ノルムが2となる代表ベクトルは全部で16個あるので、原画素値を符号化するのに必要な情報量は、

$$- \log_2 0.3 + \log_2 16 = 5.737 \text{ [bit]} \quad (9)$$

となる。

【0046】

次に、本発明により、予測誤差を計算しない場合の符号量を見積もる。概念図を図4に示す。

図中に斜線を付けた部分に属する白丸で示される代表ベクトル()は、原画素値の可能性があり、斜線の外側に属する黒丸で示される代表ベクトル()は、原画素値の可能性がない。

中心は予測値 $(x_1, x_2) = (255, 100)$ であり、先と同様に、中心からのL ノルムが2となる確率は、 0.3 であるとする。

これに属する代表ベクトルは9個であるので、原画素値を符号化するのに必要な情報量は、

$$- \log_2 0.3 + \log_2 9 = 4.907 \text{ [bit]} \quad (10)$$

と、差分を計算した場合(第9式)よりも 0.83 bit 低い値で収まっている。

【0047】

[ノルム一定面上の代表ベクトル点数]

図5に、二次元、L ノルム $= 4$ に対応する代表ベクトルを示す。

次元をL、ノルムをKとすると、代表ベクトルの個数 $N(L, K)$ は次式で与えられる

$$N(L, K) = (2K + 1)^L - (2K - 1)^L \quad (11)$$

図5に示す例では、

$$N(2, 4) = 9^2 - 7^2 = 81 - 49 = 32$$

となる。

【0048】

符号化において、ノルムが特定された後に、さらに代表ベクトルを特定するために、以下の情報量が必要となる。

【0049】

$$\log_2 N(L, K) \text{ [bit]} \quad (12)$$

さらに、図5では、黒丸()で示される「はみだし部分」(原画素値の可能性がない部分)が存在する。この個数を差し引いた白丸()の個数を求めるには、 (x_1, x_2, \dots, x_L) の各次元において、上限を $(pK_1, pK_2, \dots, pK_L)$ 、下限を $(-nK_1, -nK_2, \dots, -nK_L)$ とすると、

$$0 \leq nK_i \leq pK_i \leq K \quad (i = 1 \dots L)$$

である。はみだしがない場合、 K_i (上限、下限) $= K$ である。

【0050】

の個数を $N(L, K, nK_1, \dots, nK_L, pK_1, \dots, pK_L)$ とすると、

【0051】

10

20

30

40

50

【数5】

$$N'(L, K, {}_nK_1, \dots, {}_pK_1, \dots) = \prod_{i=1}^L ({}_nK_i + {}_pK_i + 1) - \prod_{i=1}^L (f(K, {}_pK_i) + f(K, {}_nK_i) + 1) \quad (13)$$

【0052】

として求められる。ただし、

$$\begin{aligned} f(K, K) &= K - 1 && (K = K \text{ のとき}) \\ f(K, K) &= K && (K < K \text{ のとき}) \end{aligned} \quad (14)$$

となる。

図5の例では、 $L = 2$ 、 $K = 4$ 、 ${}_nK_1 = 4$ 、 ${}_nK_2 = 4$ 、 ${}_pK_1 = 2$ 、 ${}_pK_2 = 4$ なので、

$$\begin{aligned} N(2, 4, 4, 4, 2, 4) \\ &= (4 + 2 + 1)(4 + 4 + 1) - (4 - 1 + 2 + 1)(4 - 1 + 4 - 1 + 1) \\ &= 63 - 42 = 21 \end{aligned}$$

となる。

【0053】

本発明の手法によるエントロピ削減の度合いを見積もると、

$$\log_2 32 - \log_2 21 = 0.608 [\text{bit}]$$

となる。

【0054】

次に、異なる予測値の例として、図6の場合を考える。

$L = 2$ 、 $K = 4$ 、 ${}_nK_1 = 4$ 、 ${}_nK_2 = 3$ 、 ${}_pK_1 = 2$ 、 ${}_pK_2 = 3$ なので、

$$\begin{aligned} N(2, 4, 4, 3, 2, 3) \\ &= (2 + 4 + 1)(3 + 3 + 1) - (2 + 4 - 1 + 1)(3 + 3 + 1) \\ &= 49 - 42 = 7 \end{aligned}$$

となり、はみだし部分が大幅に削除されている。

【0055】

〔距離尺度が L^1 ノルムの場合〕

次に、原点からの距離が L^1 ノルムに従う、いわゆるピラミッドベクトル量子化の場合を、図7を用いて説明する。先と同様に、点線は等確率面、斜線部は画素値が取り得る範囲(0...255)を示している。

大きな円で示された (x_1, x_2) が原画素であり、このノルム(図では4)に属する代表ベクトルは、範囲外も含めると16個、範囲内は10個である。

この場合、原画素値と予測値との差分をとらないとき(本発明)のエントロピは、とったとき(従来法)に比べ、

$$\log_2 16 - \log_2 10 = 0.678 [\text{bit}] \quad (15)$$

低くなる。

【0056】

〔ノルム一定面上の代表ベクトル点数〕

図7の L^1 ノルム=4の部分を抜き出したものが図8である。

【0057】

L 次元のピラミッドベクトル量子化の、 L^1 ノルム=Kである代表ベクトルの点の個数を、先と同様に $N(L, K)$ とする。これは次のように漸化的に求められる(非特許文献1参照)。

【0058】

・ $K = 1$ のとき

$$N(L, K) = 2L$$

・ $L = 1$ のとき

$$N(L, K) = 2$$

10

20

30

40

50

・それ以外

$$N(L, K) = N(L, K-1) + N(L-1, K-1) + N(L-1, K) \quad \dots (16)$$

従って、

$$\begin{aligned} N(2, 4) &= N(2, 3) + N(1, 3) + N(1, 4) \\ &= N(2, 2) + N(1, 2) + N(1, 3) + 2 + 2 \\ &= N(2, 2) + 2 + 2 + 4 \\ &= N(2, 1) + N(1, 1) + N(1, 2) + 8 \\ &= 4 + 2 + 2 + 8 \\ &= 16 \end{aligned}$$

10

となる。

次に、「はみだし部分」の個数を考える。例えば ${}_p K_1 = 1$ により 5 個、 ${}_p K_2 = 3$ により 1 個の「はみだし」代表ベクトル () が生じている。

【0059】

まず、簡単のため、 ${}_p K_1$ のみが ${}_p K_1 < K$ であり、残りは ${}_n K_i, K_i = K (i = 2, \dots, L)$ である場合を考える。

【0060】

そのような場合のはみ出し代表ベクトル () の個数 $M(L, K)$ は、第 16 式の N を用いて、

20

$$K = 0 \text{ のとき: } M(L, K) = 1$$

$$L = 1 \text{ のとき: } M(L, K) = 1$$

$$\text{それ以外: } M(L, K) = (N(L, K) + N(L-1, K)) / 2 \quad \dots (17)$$

において、

$$M(L, K - {}_p K_1 - 1)$$

から求められる。

【0061】

図 8 のような $L = 2, K = 4, {}_p K_1 = 1$ の場合、

$$\begin{aligned} M(2, 4 - 1 - 1) &= M(2, 2) \\ &= (N(2, 2) + N(1, 2)) / 2 \\ &= (8 + 2) / 2 = 5 \end{aligned}$$

30

となる。

【0062】

同様の手法は他の次元にも適用でき、例えば ${}_p K_2 = 3$ の場合、

$$M(2, 4 - 3 - 1) = M(2, 0) = 1$$

のように求められる。

【0063】

こうして、代表ベクトル総数 N とはみ出し個数 M から、 の個数が $N - M$ として求められる。

40

【0064】

なお、上記のようにはみ出し量が正確に求まるには、以下の条件を満たしている必要がある：

- ・ i 番の座標軸に垂直な $x_i = K_i$ 面の、全端点の L^1 ノルムが常に K 以上であること、
- ・ すなわち、 $|K_j|, |255 - K_j| (j = i)$ のうち、最小値を与える座標軸 j を残し、それ以外の座標値が全て 0 であるような点の L^1 ノルム (これが全端点の中で最小の L^1 ノルムを与える) が K 以上であること、

・すなわち、

【0065】

50

【数6】

$$\min \left(\min_{j \neq i} |K_j|, \min_{j \neq i} |255 - K_j| \right) + |K_i| \geq K \quad (18)$$

【0066】

具体例として、図8における斜線を施した四角部分の4つの角が「全端点」となり、この場合は全ての短点が L^1 ノルム=4のライン上もしくはその外側にあり、はみ出し量が正確に求まる。

仮に、この斜線部分が図16のように設定されていると、 L^1 ノルム=4のラインの内側に存在する端点があり($(x_1, x_2) = (1, 1)$ に存在する端点)、重複カウントされる「はみだし」代表ベクトルが存在してしまう($(x_1, x_2) = (2, 2)$ に存在する代表ベクトル)。

この場合、はみ出し量が正確に求まらない。

【0067】

[一般のベクトル量子化の場合]

よく知られたLBGアルゴリズム(非特許文献2)などによる、代表ベクトルの配置が規則的でない、一般のベクトル量子化においては、本発明は次のように適用される。

【0068】

図9が、通常の差分信号(二次元)に対するベクトル量子化の模式図である。差分信号は、各次元-255...255の値をとるため、代表ベクトルの設計においてもこの範囲を網羅するようになされる。画像予測差分信号のように原点(0,0)付近に信号が生起する確率が高い場合、この図のように代表ベクトルも原点付近において密になり、周辺において疎となる。

【0069】

図10は、従来の符号化方法に対応するもので、予測値(x_1, x_2)における原信号の符号化を行っている。原信号は0...255の値しか取り得ないため、代表ベクトルはその外にも存在してしまう。したがって、一次元の符号化議論のときと同様に、符号量の無駄が発生する。

【0070】

図11は、本発明の符号化方法に対応するもので、予測値(x_1, x_2)に応じて代表ベクトルを設計する。

当然ながら各次元0...255の範囲内で設計するため(即ち、 $0 \leq x_1, x_2 \leq 255$)、従来方法(図10)で生じていたような代表ベクトルの「はみだし」は起きず、符号量の無駄のない符号化が行える。

【0071】

以下では、本発明の実施形態の一例として、予測を画素ブロック単位で行い、ブロック予測による画素ブロック値の条件付き分布の符号化にベクトル量子化を用いる場合の例を説明する。

本実施形態では、事前の学習用データをもとに、ベクトル量子化の代表ベクトルを作成する。

符号化対象ブロックの予測値 x に対応した学習データのみを用いてもよいが、次元が高い場合にはデータ数が少なくなるので、例えば原画素値との差分値($x - x'$)を記憶しておき、学習には差分値に予測値を加えた値を用いる、という構成も可能である。

【0072】

図12に、本実施形態のフローチャートを示す。実線矢印は処理の流れ、波線矢印はデータの流れを示す。

【0073】

画素値予測ステップ101において、ブロック単位に動き補償あるいは画面内予測などにより、符号化対象ブロックの画素値予測を行い、予測値102(ベクトル量)を得る。

シフトステップ104において、別途蓄積しておいた差分分布データ103（ベクトル量）における差分値に予測値を加算（シフト）し、次いでクリッピングステップ105において、ベクトルの各要素を0...255にクリッピングする。これが学習の元データとなる。

【0074】

代表ベクトル設計ステップ106において、例えばLBGアルゴリズムにより、学習の元データを用いて代表ベクトルを設計する（例えば図11のような結果を得る）。

【0075】

次に、符号化対象ブロックの原画素値108（ベクトル量）を、ベクトル量子化ステップ107により、原画素値最近傍の代表ベクトルへ対応づける。

10

【0076】

得られた代表ベクトルのインデックス情報を、その生起確率に基づき、量子化インデックス符号化ステップ109において、算術符号などのエントロピ符号化により符号化し、符号を出力し、終了する。

【0077】

図13A～13Dを用いて、図12に示す符号化の処理による作用を簡単な例で説明する。わかりやすく図示するため、図13A～13Dでは一次元の画素値で示す。原画素値 x は0から255までの値を取るものとする。

もし、これらの値の生起確率が一様であるとする、図13Aに示すように、各値は1/256の確率で生じることになる。このような等確率分布で生じる値の符号化コストは大きい。

20

しかし、もしこの原画素値 x の予測値 x' が得られた場合、原画素値 x の取り得る確率分布を、既知の予測誤差値の分布から等確率分布ではないものとすることができる。本方式は、この点に着目して符号化効率を改善するものである。

【0078】

原画素値 x と予測値 x' の差分である予測誤差 $x - x'$ の値（差分値）は、-255から255までの値を取り得るが、この差分分布は、多くのサンプル画像の予測符号化実験などにより求めることができる。この差分分布のデータを予め蓄積し、保持しておく。

差分分布は各予測誤差の値の頻度値もしくは確率値の分布であり、例えば図13Bのような分布となっている。図12の差分分布データ103が、図13Bのような分布を示すデータ（ただしベクトル量）である。

30

【0079】

原画素値 x を符号化するとき、画素値予測ステップ101では、まず予測値 x' を求める。

シフトステップ104では、差分分布データ103の各差分値、すなわち図13Bに示す差分分布における横軸の各予測誤差 $x - x'$ の値に予測値 x' を加算（シフト）し、この差分分布を、図13Cに示すように原画素値 x に対応するものにシフトさせる。

これが、予測値 x' が既知の場合の原画素値 x の取り得る確率分布に相当するものになる。

【0080】

40

図13Cの分布では、原画素値 x が $-255 + x'$ から $255 + x'$ の範囲まで分布するものになっている。しかし、原画素値 x は、ここでは0から255までの値しか取らないという前提であるので、続くクリッピングステップ105では、図13Dに示すように、原画素値 x の分布における0未満と255の値を超える部分を切り捨て、これにより得られた分布を必要に応じて正規化し、確率分布とする。

この図13Dに示す分布に基づいて原画素値 x を符号化すれば、図13Aに示すような等確率分布の下で符号化するよりも高い符号化効率を得られることはもちろん、図13Bに示すような裾の広い確率分布の下で符号化する（従来法に対応）よりも効率は高くなる。

【0081】

50

このような確率分布の下で効率よく符号化する一例としてはベクトル量子化があり、さらに本実施形態では、図 1 3 D に示すような確率分布に基づいて、図 1 2 に示すステップ 1 0 6 ~ 1 0 9 により、量子化代表ベクトルの配置を決め、ベクトル量子化を行っている。

【 0 0 8 2 】

図 1 4 に、本実施形態の符号化装置のブロック図を示す。

【 0 0 8 3 】

信号端子 3 0 0 より、映像原信号および復号済み映像信号が入力される。

符号化対象ブロックの原画素値は、原画素値蓄積メモリ 3 0 6 に蓄積される。

画素値予測器 3 0 1 において、ブロック単位に動き補償あるいは画面内予測などにより、符号化対象ブロックの画素値予測を行い、予測値（ベクトル量）を得、予測値蓄積メモリ 3 0 2 に記憶する。

10

加算・クリッピング器 3 0 4 において、差分分布蓄積メモリ 3 0 3 に別途蓄積しておいた差分分布データベクトルと予測値を加算し、ベクトルの各要素を 0 . . . 2 5 5 にクリッピングする。これが学習の元データとなる。

【 0 0 8 4 】

代表ベクトル設計器 3 0 5 において、例えば L B G アルゴリズムにより、学習の元データを用いて代表ベクトルを設計する。

【 0 0 8 5 】

次に、メモリ 3 0 6 に蓄えられた符号化対象ブロックの原画素値（ベクトル量）を、ベクトル量子化器 3 0 7 により、原画素値最近傍の代表ベクトルへ対応づける。

20

【 0 0 8 6 】

得られた代表ベクトルのインデックス情報を、その生起確率に基づき、量子化インデックス符号化器 3 0 8 において、算術符号などのエントロピ符号化により符号化し、符号を出力端子 3 0 9 へ出力し、終了する。

【 0 0 8 7 】

図 1 5 に、本実施形態における復号時のフローチャートを示す。実線矢印は処理の流れ、波線矢印はデータの流れを示す。

【 0 0 8 8 】

画素値予測ステップ 4 0 1 において、ブロック単位に動き補償あるいは画面内予測などにより、符号化対象ブロックの画素値予測を行い、予測値 4 0 2 （ベクトル量）を得る。

30

加算ステップ 4 0 4 において、別途蓄積しておいた差分分布データベクトルと予測値を加算し、次いでクリッピングステップ 4 0 5 においてベクトルの各要素を 0 . . . 2 5 5 にクリッピングする。これが学習の元データとなる。

【 0 0 8 9 】

代表ベクトル設計ステップ 4 0 6 において、例えば L B G アルゴリズムにより、学習の元データを用いて代表ベクトルを設計する。

【 0 0 9 0 】

得られた代表ベクトルのインデックス情報の生起確率に基づき、量子化インデックス復号ステップ 4 0 7 において、インデックスを復号する。

40

【 0 0 9 1 】

次に、ベクトル逆量子化ステップ 4 0 8 において、インデックスに対応する代表ベクトル値を求めてこれを出力し終了する。

【 0 0 9 2 】

なお、本実施形態の復号装置のブロック構成については、図 1 4 に示す符号化装置のブロック図の説明と、図 1 5 に示す復号時のフローチャートの説明から容易に類推できるので、図面を用いた説明は省略する。

復号装置の構成も、基本的には図 1 4 に示す符号化装置のブロック図と同様であり、図 1 4 における原画素値蓄積メモリ 3 0 6 が、復号装置では量子化インデックス符号化データ蓄積メモリになり、ベクトル量子化器 3 0 7 が、復号装置では量子化インデックス復号

50

器になり、量子化インデックス符号化器 3 0 8 が、復号装置ではベクトル逆量子化器になる。

【 0 0 9 3 】

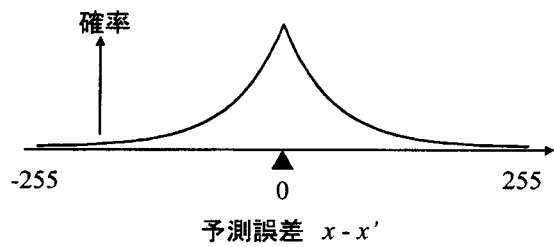
以上の画像・映像符号化および復号の処理は、コンピュータとソフトウェアプログラムとによっても実現することができ、そのプログラムをコンピュータ読み取り可能な記録媒体に記録して提供することも、ネットワークを通して提供することも可能である。

【産業上の利用可能性】

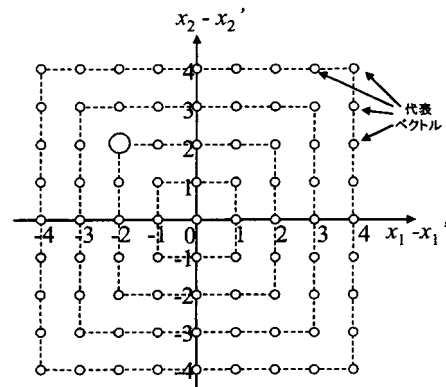
【 0 0 9 4 】

本発明によれば、従来の、予測値と原画素値の差分を扱う場合の「予測値という重要な情報の欠落」を発生させることなく、余すことなく符号化に利用し、結果としてより少ない符号量で画像（静止画像もしくは映像）を符号化することができるようになる。

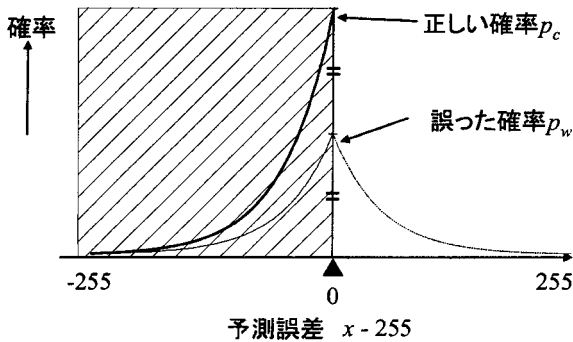
【 図 1 】



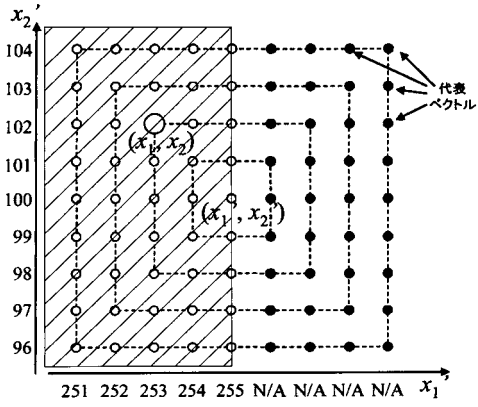
【 図 3 】



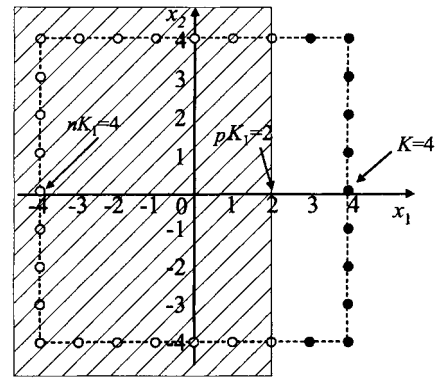
【 図 2 】



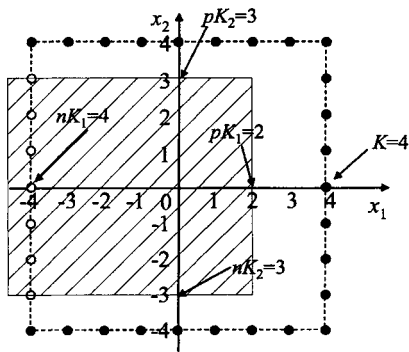
【 図 4 】



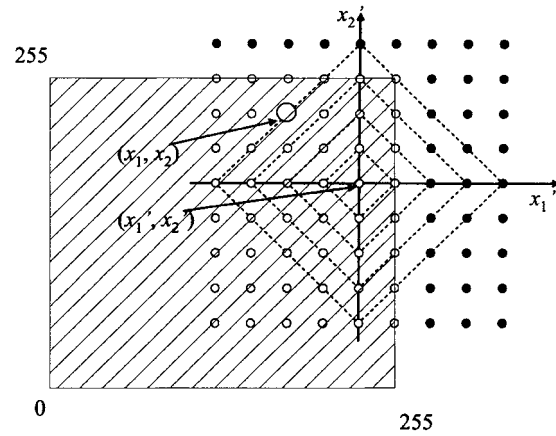
【 図 5 】



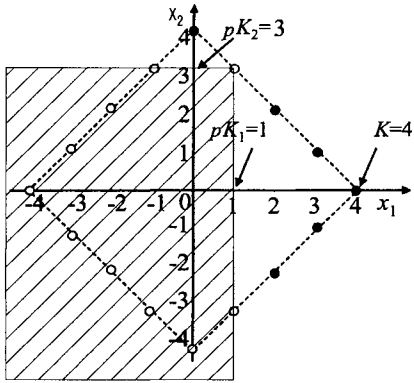
【 図 6 】



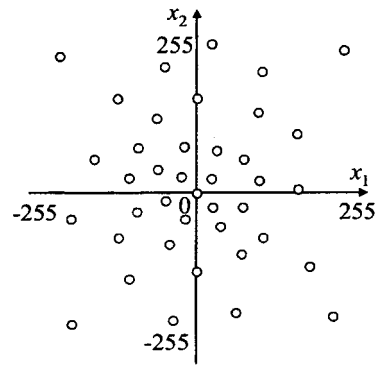
【 図 7 】



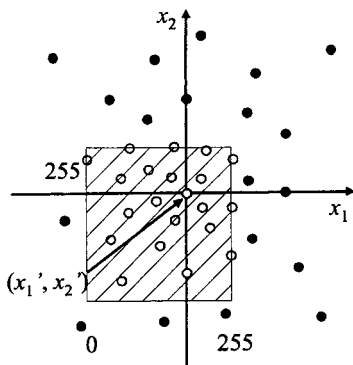
【 図 8 】



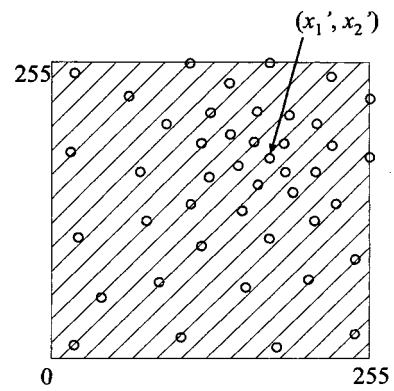
【 図 9 】



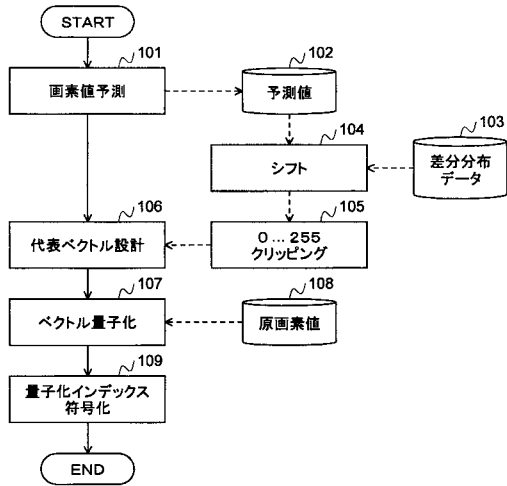
【 図 10 】



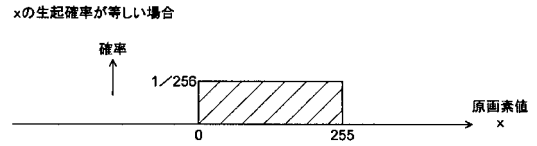
【 図 11 】



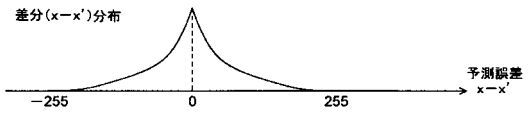
【図12】



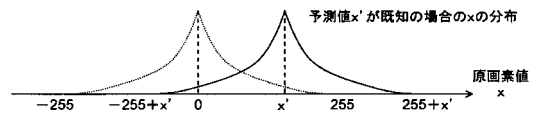
【図13A】



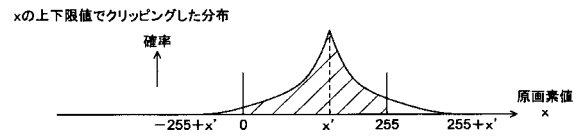
【図13B】



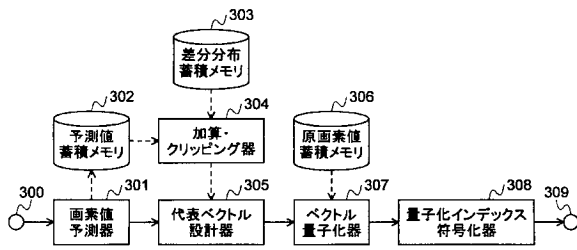
【図13C】



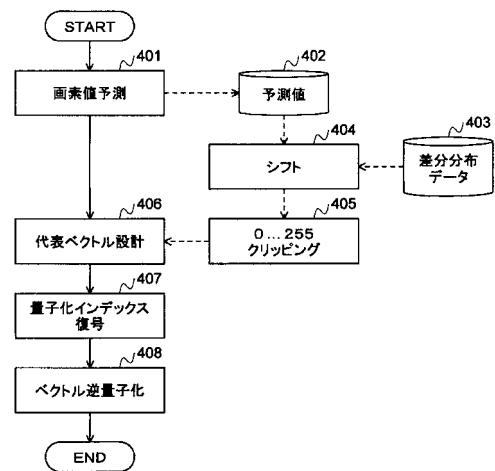
【図13D】



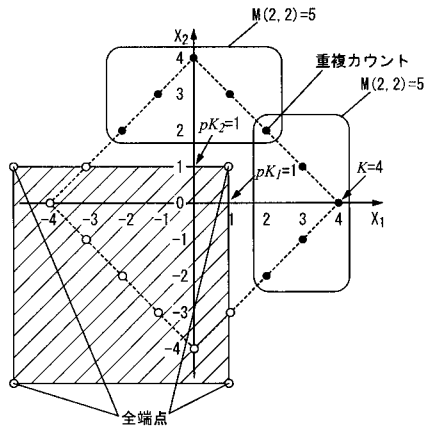
【図14】



【図15】



【図16】



フロントページの続き

(56)参考文献 特開平3 - 145887 (JP, A)
特開2006 - 229623 (JP, A)

(58)調査した分野(Int.Cl., DB名)
H04N 7/24 - 7/68