ベクトル量子化による画像情報圧縮に関する研究

波　　通隆

Study on Image Compression using Vector Quantization

Michitaka NAMI

抄　録

ディジタル画像情報を処理するシステムにおいては，情報量が非常に多くなるため，その情報の転送に多大の時間を要し，保存のために大容量のメモリーが必要となる。この問題を解決することを目的に，ベクトル量子化（VQ：Vector quantization）を用いた画像情報圧縮法であるTSVQ(tree structured vector quantization)とPTSVQ(pruned tree structured vector quantization）について実験・検討し，低符号化レートの圧縮などに有効であることを確認した。

1．はじめに

最近のコンピュータの分野では，パソコン，ワークステーションのマルチメディア化の研究が盛んであり，動画や静止画の高効率圧縮についての重要度が高まってきている。また，通信の分野では，ディジタル技術により伝送，交換，サービスを統合するサービスである総合ディジタル通信網(ISDN)が実用化され，家庭にも光ファイバを結ぶ広帯域ISDNの導入が現実のものとなってきている。これらディジタル情報・通信分野の情報伝達媒体として最も重要なのは，人間が得る情報の70％以上を占めるといわれる画像である。しかし，画像情報をディジタル化すると，情報量がきわめて多くなる。例えば，ISDNの音声の伝送レートの基準は64kbit/秒であり，これに比較して，画像のディジタル化では，その1000倍以上の情報量になってしまい[1)]，これでは，当然，速度的にも，コスト的にも，実用にはならない。このような背景から，ここ数年の情報処理分野における画像情報圧縮技術の進歩にはめざましいものがある。

一方，画像など大量のデータを扱うシステム開発では，それらデータの転送に多大の時間を要し，保存のために大容量のメモリーが必要となる。したがって，コンパクトで，高速で，安価なシステム開発が難しくなる。このため，圧縮伸長のオーバーヘッドによるシステム低下が生じない高速CPUを用いることによって，この問題の解決がなされるようになってきた。また，従来，圧縮はCPU性能を劣化させると敬遠されていたが，CPU性能の向上にともなってテキスト・ファイルやプログラム・ファイルを圧縮して外部記憶装置に格納し，システム性能を引き上げることが可能になってきた。これらの点から，情報圧縮技術の必要性が画像に限らず多くの種類のデータについて要求されている。

本研究では，情報圧縮技術の一つであるベクトル量子化（Vector Quantization：以下，VQと呼ぶ）による圧縮手法の中で，現在，スタンフォード大学において医用画像などへの応用が進められているTSVQ（tree structured vector quantization)とPTSVQ (pruned tree structured vector quantization）を用い，これらの他のVQとの優位性などについてテスト画像を用いて実験・検討したので報告する。

2．情報圧縮手法

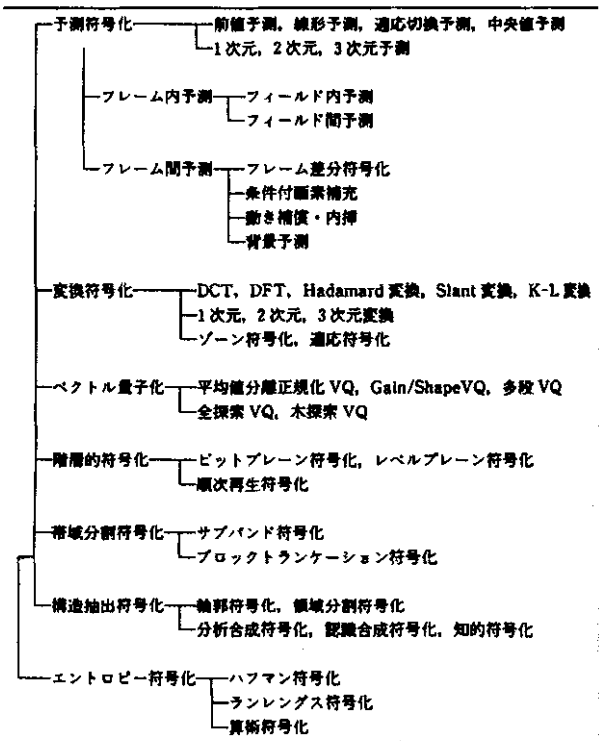
情報圧縮手法をアルゴリズム別に整理したものを表1に示す[1]。アルゴリズム的に代表的なのは，予測符号化と変換符号化（特に2次元DCT）であるが，レートが低い場合などにはベクトル量子化が有力な方式の一つになる。表2に，画像情報圧縮技術の適用例を示す[1]。この表でも示されているが，ベクトル量子化は通信用動画，記録用動画，濃淡またはカラー静止画に適している。

表1の各手法の優位性を論じる場合，最終的には復元画像のひずみが少ないほど良いことになる。しかし，圧縮対象となる画像の種類にも影響されるので一概にどの手法が最も優れているとは言えない。

表1　画像情報圧縮技術の適用例

画像信号の種類		ビットレート	符号化方式
放送用動画	HDTV画像	120~140 Mb/s	フレーム内/間適応予測
	標準方式テレビ画像	30~45 Mb/s	直接符号化 フレーム内/間 複合差分または適応予測
		15 Mb/s	分離符号化 フレーム・フィールド間 適応予測
	CATV/CCTV	32 Mb/s	直接符号化 フレーム内1次元高次予測 または2次元予測
通信用動画	テレビ会議画像	6.3 Mb/s	分離符号化 フレーム間条件付画素補充 複合差分符号化
		1.5~2.0 Mb/s	動き補償予測，背景予測 駒落し
		384 kb/s	動き補償予測，DCT 駒落し
	テレビ電話画像	64~128 kb/s	動き補償予測 DCTまたはVQ 駒落し
記録用動画	CD-ROM記録用動画	約1 Mb/s 5~10 Mb/s	周期的ブロック化 動き補償予測・内挿 DCTまたはVQ サブバンド符号化
静止画	自然画	0.5~2 b/画素	ブロックトランケーション符号化 順次再生符号化（PCS） 適応DCT符号化，VQ 予測符号化+算術符号化
	2値（FAX）		MH符号化，MR符号化 MMR符号化 算術符号化

表2　各種の画像符号化アルゴリズム



3．ベクトル量子化による画像情報圧縮

音声情報などの圧縮法では，圧縮対象となるデータのサンプル値が個々独立なものとして一個づつ処理するスカラー量子化（Scalar Quantization）がよく知られている。これは，データの統計的性質としての分布と相関を考えて，統計的符号化を行うものである。一方，ベクトル量子化は，サンプル値の系列（つながり）を適当な大きさに区切って，それを一まとまりのものとしてその性質を利用する符号化である。n個のサンプル値系列を一個のデータと考えると，これはn次空間の一点または原点からその点までのベクトルと考えられる。[2]

VQを用いた画像の圧縮，復元の考え方を図1に示す[3]。送信側符号器（ENCODER）と受信側復号器（DECODER）は，入力画像からサンプル（標本化）される複数要素からなる画像データをまとめたベクトル（image data：以下，画像ベクトルと呼ぶ）を圧縮，復元するための参照データとなる量子化代表ベクトルの集合であるコードブック（CODE BOOK）をそれぞれ持っている。コードブックは，学習系列（training sequence）と呼ばれる何枚（何十枚）かの標準画像からロイド・ア

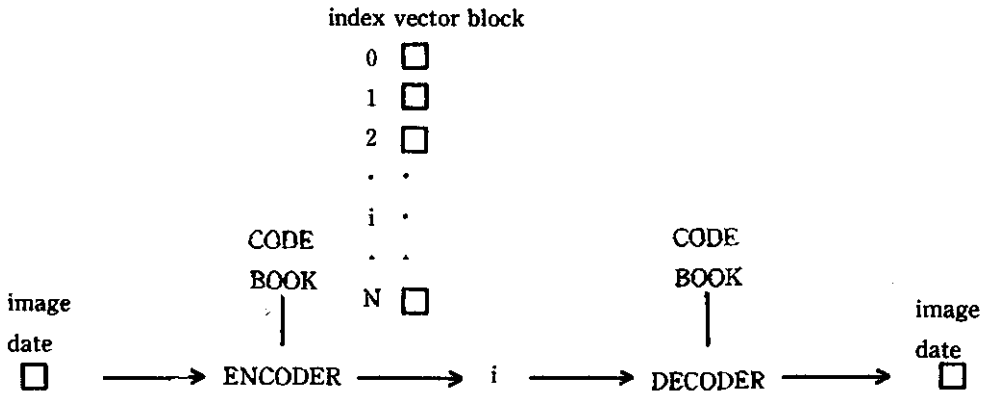


図1　VQのブロック図

ルゴリズム（Lloyd Algorithm）を適用して設計される。標準的なVQでは，コードブックは，入力画像からの画像ベクトルと同じ次元数の量子化代表ベクトル（Vector block）を添字番号すなわちインデックス（index）とともに持っている。これらベクトルの次元数は，2×2または，4×4がよく用いられる。例えば，4つの標本値を要素とする画像ベクトル2×2で，コードブック設計の際には，学習系列の画像の一番左上の2×2マトリックスの4画素の四次元配列が最初のベクトルになり，以下，ラスタースキャン順に順次2×2の4画素が取り出され，処理される。

圧縮は，入力されたベクトルと最も似ているベクトルをコードブック中から探索し，そのベクトルの持つインデックスを出力する。復元は，それらインデックスのベクトルを同じコードブックに基づいて順に出力する。

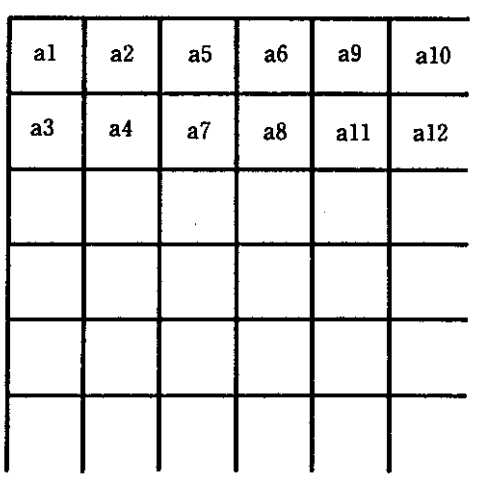


図2　image data

図2に一枚の画像からサンプルされる画像ベクトルの例を示す。この例では，2×2のベクトル（a 1，a 2，a 3，a 4）から始まり，（a 5，a 6，a 7，a 8），（a 9，a 10，a 11，a 12）とサンプルされて行く。ここで，a 1 ～ a 12は画像の濃度値に相当する。画像ベクトルとコードブックの内容とが比較され，最も近い値を持つコードブック中のベクトルのインデックスが出力され，その画像ベクトルについての符号化が終了する。この操作が全画像ベクトルについて行われ，一枚の画像の圧縮が終了する。

なお，以下，ベクトルを記号□により表す。

2×2の圧縮復元精度は，1bit/pixelで，4×4の精度は，1/2　bit/pixelと云われている。

VQの動作順序を以下に示す[4]。

①入力画像である圧縮対象画像の画像ベクトルを入力する。

②この画像ベクトルと最も似ているベクトルをコードブックの中から探す。

③そのベクトルのインデックスを出力する。

④この①～③の手順を入力画像の全ての画像ベクトルについて行う。出力されたインデックスが入力画像についての符号化の結果であり，codeである。

⑤復元は，得られたインデックスの順にコードブックを参照し，対応するベクトルを出力し，最終的な復元画像を得る。

手順②で，コードブックのすべての要素について探すのがフルサーチ（full search）VQと云われる。

3.1　コードブックの作成[5)]

ロイド・アルゴリズムを用いた学習系列からのコードブック作成の手順について説明する。

①分岐（splitting）

ベクトル空間での学習系列のデータ分布が図３のような場合を想定する。この際，Q 枚の画像からなる学習系列とすると，学習系列のデータは Q 枚の画像の和となる。2×2 の画像ベクトル $X=(X_1, X_2, X_3, X_4)^T$ とすると，4 次元の空間が必要であるが，説明上，2 次元平面の分布で考える。図の×印が学習系列の画像ベクトルであり，これらベクトルから分岐により量子化代表ベクトルを求めて行く。

分岐を実現するロイド・アルゴリズムの手順は，基本的には，

ステップ 1：重心近傍に点をマップする。

ステップ 2：重心を移動する。

の繰り返しで，停止は，全体のひずみ率が設定したある値より小さくなった時である。

図３で，まず最初，全体の重心(centroid：濃度平均値) X_A を計算する。X_A は次式により与えられる。

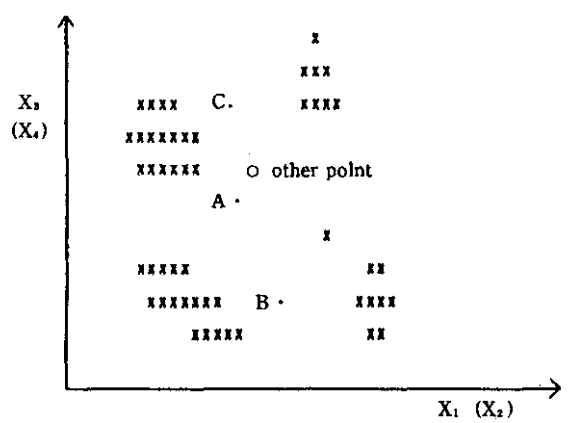


図3　分岐

$$X_A = (X_{A1}, X_{A2}, X_{A3}, X_{A4})$$

ここで，X_{A1}～X_{A4} は各ベクトル要素についての平均値である。また，この時のひずみ D_0 は，次式で与えられる。

$$D_0=\frac{1}{N_A}\sum_{i=0}^{N-1}\| X_i-X_A \|^2$$

N（＝N_A）は全ベクトル数で，画像の大きさ（n×n）とベクトル・サイズ（次元数：subblock）（m×m）によって決まり，

N ＝ ((n×n)/(m×m))×(学習系列に用いた画像の枚数)

である。

例えば，2×2 の画像ベクトル（次元数＝ 4），学習系列の画像の大きさが 512×512 で，画像枚数が 10 枚とすると，全ベクトル数 N は，10×(512×512)/(2×2) ＝ 653600 となる。図３の X_1 がそれらベクトルに当たる。上式の重心 X_A を座標上に求め，この点の近傍にベクトル（other point）を 1 つセットする。これがステップ 1 になる。これら各点（重心または近傍点）と距離が近いベクトルを選択し，グループ化する。次に，各グループの重心を求め，各点をそれら重心へ移動する。この処理が，ステップ 2 になる。図３での A が重心で，B がその移動点，C が近傍点についての移動点である。以下，同様に，B，C 各点近傍にベクトルを 1 つづつセットし（ステップ 1），ステップ 2 を実行する。この分岐処理は，次に示されるひずみ率の値にしたがって停止が決定される。

②分岐の停止の条件

バランス・トリー（第 4 章参照）の場合を例として説明する。分岐のレベル（level）を　L とし，この時のひずみ率は次式で表される。

$$\text{ひずみ率}=\frac{D_{L-1}-D_L}{D_L}$$

ここで，

$$D_L=\sum_{i=0}^{M-1}\left(\frac{1}{N_{Ai}}\sum_{j=0}^{N_{Ai}-1}\| X_{Ai}-X_{Aij} \|^2\right)$$

M　　：レベル L において分岐したグループ数である。

X_{Ai}　：レベル L において得られた各グループの重心で，M 個ある。

N_{Ai}　：レベル L におけるグループの第 i 番目（Ai）に属する全ベクトル数である。

X_{Aij}：レベル L におけるグループの第 i 番目（Ai）のグ

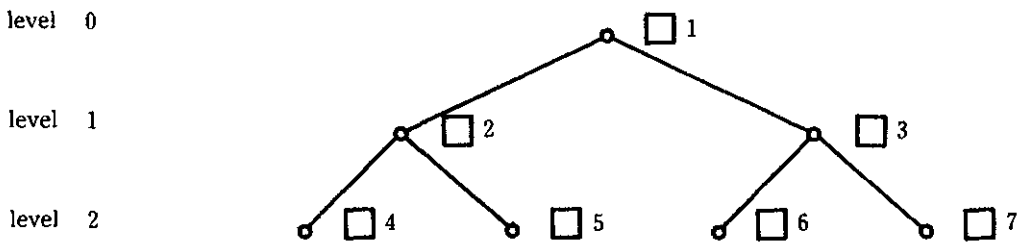


図4 treeによる表現

ループに属するベクトルで，N_{Ai} 個ある。

$$X_{Ai}=\frac{1}{N_{Ai}}\sum_{j=0}^{N_{Ai}-1}X_{Ai,j}$$

ここで，停止は，最小2乗誤差（mean squared error：MSE）であるこの D_L によって決定される上記ひずみ率が，ある設定した値を越えた時点である。

最終的な分岐レベルをL＝2とすると，この分岐の様子を図4に示すようなトリーにより表現することができる。この際，得られたベクトルは□4．□5，□6，□7及びL＝0での□1，L＝1での□2，□3となり，これらがコードブックのベクトル集合になる。また，標準的VQの場合には，コードブックは□4，□5，□6，□7になる。この違いは，入力された画像ベクトル□xについての照合探索法の違いによる。すなわち，標準的VQでは，コードブックの内容をすべて探索し，最も照合するベクトルのインデックスを出力するフル・サーチ（全探索）である。後述するTSVQの場合には，最初，□xはルート・ノードにあり，｜□x－□2｜と｜□x－□3｜を計算し，小さい他のノードへ移動する。min（｜□x－□i｜）$_{i=2,3}$＝｜□x－□2｜ならば，□xは□2へ移動する。同様に，min（｜□x－□i｜）$_{i=4,5,6,7}$＝｜□x－□5｜とすると，□xは□5へ移動し，探索結果として□5が求められる。

3.2 圧縮（低減）効果[2)]

コードブック中のベクトルの数をN，そのベクトルの次元数をkとしたとき，コードブックの中のすべてのベクトルを指定するために必要となるインデックスのビット数は $b=\log_2 N$ となる。したがって，1ベクトルのサンプル（画素）数はkゆえ，1サンプル当りの所要ビット数，すなわち符号化レートRはb/k（bit/sample）となる。これが1画素のサンプル当りのビット数Bに較べて小さければそれだけビット数を低減できたことになる。

例えば，画像ベクトルの次元数2×2，すなわち，k＝4で，B＝8としたとき，コードブック中のベクトル数がN＝24×8＝232以下であれば，低減効果があることになる（k＝4，B＝8から，8≧b/4となるためにはbが32以下であればよい）。

3.3 量子化演算量とメモリ量[1)]

サンプル当たりの符号化レートRと次元数kとが指定されると，レベル数は $2^{k\cdot R}$ で与えられる。したがって，Rやkを大きくして行くにしたがって，レベル数Nは指数関数的に増大してしまい，量子化代表ベクトルを格納するのに必要なメモリ量はN×kと膨大なものになる。しかし，大容量のメモリが安価に利用できるようになってきており，高符号化レートは別にして，低符号化レートではメモリ量については大きな問題ではなくなってきた。

一方，最適ベクトル量子化では，量子化操作は全探索で行われるので，量子化には入力される画像ベクトルごとにN×k回の乗算が必要である。この量子化演算量の問題を解決するのが本研究で取り上げたTSVQ，PTSVQの基本となるトリー・サーチ（tree search）である。この結果，一般に，$k\cdot\log_2 N = k^2\cdot R$ 回の乗算回数で量子化操作を行えるようになる。

4．TSVQ[6),7)]

4.1 TSVQ（バランス・トリー）

TSVQ（バランス・トリー）の場合のブロック図を図5に示す。

この図のようにENCODING FILE（以下，符号化ファ

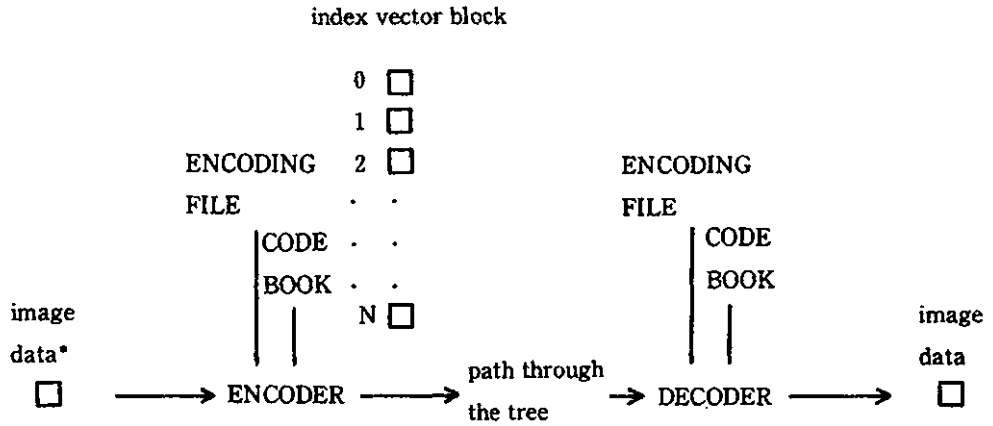


図5　TSVQ(balanced tree)のブロック図

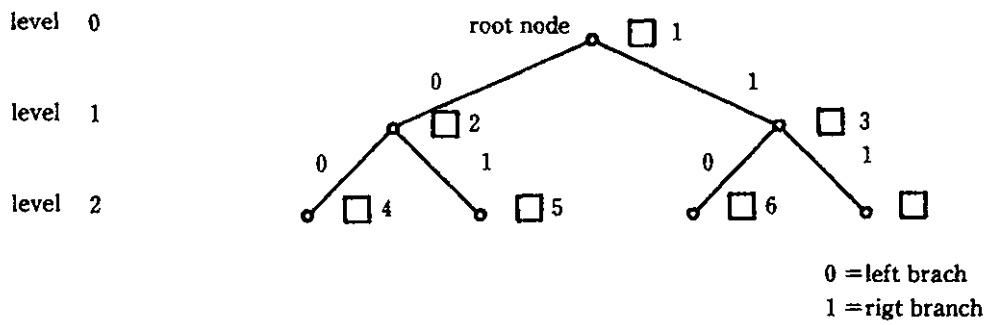


図6　ENCODING FILE for TSVQ(balanced tree)

イルと呼ぶ）を圧縮及び復元側が持っている。このファイル例を図6（L＝2）に示す。コードブック作成時の分岐により得られたトリーに基づいて，right branch＝1，left branch＝0という規則で，枝に1または0の情報を与える。この結果，ルート・ノードから各ベクトルへのパスは，□2＝（0），□3＝（1），□4＝（00），□5＝（01），□6＝（10），□7＝（11）となる。このベクトルへのパス情報が符号化ファイルに格納されて，圧縮時の照合探索と復元において参照される。

4.2　TSVQ（アンバランス・トリー）

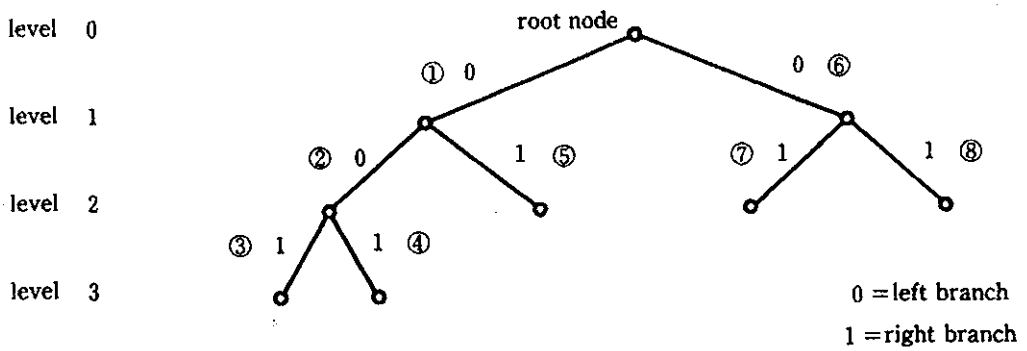
TSVQ（アンバランス・トリー）のTSVQ（バランス・トリー）との違いは，DESCRIPTION FILE（以下，記述ファイルと呼ぶ）が圧縮・復元の参照データとして加わっていることである。このファイルは，アンバランス・トリーの構造を記述したファイルで，学習系列からコードブックを得る過程で得られる。その作成規則は，terminal node（終端ノード）＝1，internal node（内部ノード）＝0である。terminal nodeはトリーが成長し最終的に分岐が停止したノード，internal nodeはトリーの成長過程で得られるノードである。ここで，図7に，例としての符号化ファイルとともに，記述ファイルを示す。このトリーの記述ファイルは00111011となる。この記述順は，図の①～⑧の順である。左側の枝が優先で，次に戻るときに右の枝を順に記述する。ルート・ノードに達したなら，同様に左枝優先で記述して行く。

TSVQ（アンバランス・トリー）の分岐アルゴリズムについて図8のようなトリーを考え，以下，説明する。

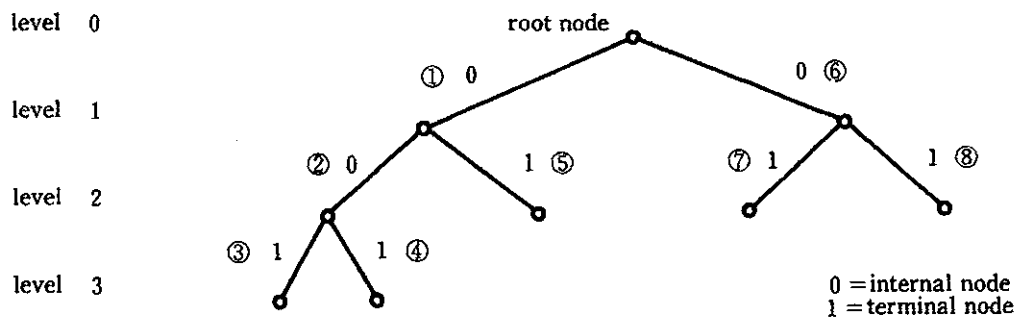
このとき，トリーの成長で，レベル1において，

$(\triangle D/\triangle R)_A > (\triangle D/\triangle R)_B$

ならば，ノードAから分岐する。ここで，$(\triangle D/\triangle R)_A$の△Dはこの分岐の一つ前の全体のひずみと分岐後の全体のひずみとの差分，△Rはこの分岐の一つ前の全体のビットレート（bit rate）と分岐後の全体のビットレート



(a) ENCODING FILE for TSVQ (unbalanced tree)



(b) DESCRIPTION FILE for TSVQ (unbalanced tree)

図7 TSVQ(unbalanced tree)

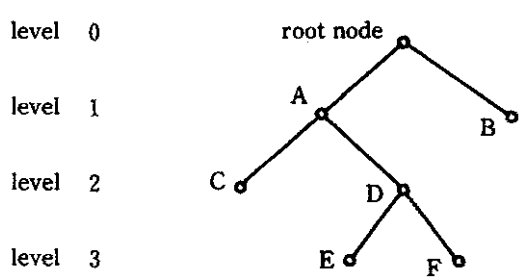


図8 unbalanced tree

との差分である。

次に，ノードC，D，Bの比較で，

$(\triangle D/\triangle R)_D > (\triangle D/\triangle R)_C > (\triangle D/\triangle R)_B$

ならば，ノードDから分岐する。このようにして上図のトリーが生成された。

このようにアンバランス・トリー成長の条件は，各終端ノードから分岐したと仮定して max(△D/△R) であるノードから常に分岐する。

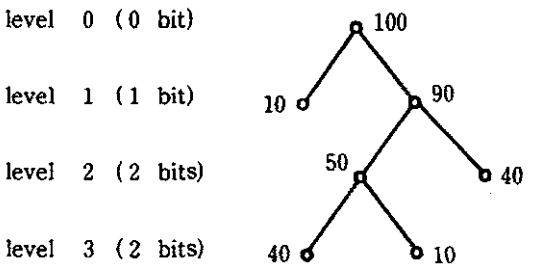
ここでビットレートは，

$\sum_i (b_i \times Tnd_i)/Rnd$

により，与えられ，b_i はトリーのレベル i におけるビット数，Tnd_i はレベル i における終端ノードに属する全ベクトル数，Rnd はルート・ノードの全ベクトル数である。

例えば，ルート・ノードのベクトル数を 100 として，次の図9のようなトリーを考える。ノードの添え字はそのノードに属するベクトル数である。ビットレートは，2.4 と求められる。なお，画像の大きさが 512×512 で，ベクトル・サイズが 4×4 の画像 5 枚からなる学習系列の場合のルート・ノードのベクトル数は，$(512\times512)/(4\times4))\times5images = 81920$ となる。

このように，アンバランス・トリーでは，探索対象となるトリーの分岐が平衡に行われない。したがって，探索時間が少なくなる。以下の実験での TSVQ はこのアンバランス・トリーを用いた。



このtreeについての
ビットレートハ、
(1 * 10+2 * 40+3 * 50)/100≒
2.4 bits/vector
として求められる。

図9 ビットレートの計算例

5. PTSVQ (Pruned TSVQ)[5)]

TSVQの成長においてはmax(△D/△R)であったが，Pruned TSVQではTSVQ(アンバランス・トリー)の結果生成されたトリーについて，あるノードから下のトリー(sub-tree:下位トリー)を切り取ったと仮定して，△D/△Rを計算し，その値が最小となる場合のノードから下のトリーを切り取ったトリーを探索に用いる。例えば，アンバランス・トリーの成長によって生成されたトリー(large tree:図10)を考える。このとき，12個あるノードのうち，ある一つのノードを切り取ったトリーとそれを切り取る前のトリーを考える。この両者に基づいて(△D/△R)を求める。これを全ノードについて行う。その結果，

(△D/△R) cut1 (△D/△R) cut2 (△D/△R) cut3 (△D/△R) cut4
(△D/△R) cut5 (△D/△R) cut6 (△D/△R) cut7 (△D/△R) cut8
(△D/△R)cut9(△D/△R)cut10(△D/△R)cut11(△D/△R) cut12

が得られる。ここで，(△D/△R)cutiはi番目のノードを切り放したトリーとその切り放す前のトリー間の△D/△Rを示す。これらのうち，最小の値を持つノードを切り取る。例えば，ノード⑤がそれであれば，そのノードの下のトリーを切り取ってつくられるトリーが新しいトリーになる。

このように，TSVQ(アンバランス・トリー)のトリー生成における成長においては，ビットレートの増分に対してひずみが最小になるように進行するが，この得られたTSVQのトリーからPTSVQのトリーを生成する場合

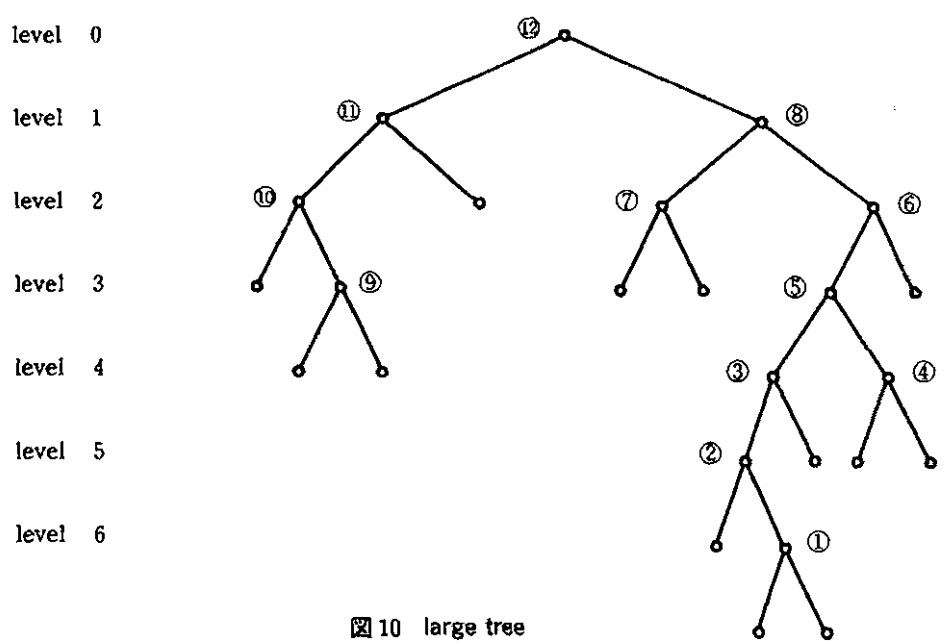


図10 large tree

には，ビットレートの増分に対してひずみが最大になるように得られる。この結果，同じビットレート，例えば，1bpv であっても，常に PTSVQ が TSVQ よりひずみが少ないか等しくなる。

6．実験

6.1　実験 1（TSVQ －アンバランス・トリー－）

6.1.1　実験条件とデータ

実験は，2 種類の学習系列について，ビットレートを変えた場合と，ベクトル・サイズを変えた場合の 2 通りについて行った。実験条件を表 3 に示す。

実験に用いた学習系列の画像データを以下に示す。

① 学習系列：couple,crowd,man,woman1,woman2

各々，計算機ファイル名が image1,image2,image3，image4,image5 に対応する。

image1～5 を図 11（a）～（e）に示す。以下，この系列を USC と呼ぶ。

② 学習系列：usc8,usc9,usc12,usc13,usc14

表 3　実験条件（TSVQ-unbalanced tree-）

experiment No.	subblock	bit rate	training sequence
1	4 by 4	4	(usc)image1-5(couple etc.)
2	4 by 4	4	(usc2)image11-15(usc8 etc.)
3	4 by 4	8	(usc)image1-5(couple etc.)
4	4 by 4	8	(usc2)image11-15(usc8 etc.)
5	4 by 4	12	(usc)image1-5(couple etc.)
6	4 by 4	12	(usc2)image11-15(usc8 etc.)
7	2 by 2	4	(usc)image1-5(couple etc.)
8	2 by 2	8	(usc2)image11-15(usc8 etc.)
9	2 by 2	12	(usc)image1-5(couple etc.)
10	2 by 2	4	(usc2)image11-15(usc8 etc.)
11	2 by 2	8	(usc)image1-5(couple etc.)
12	2 by 2	12	(usc2)image11-15(usc8 etc.)

表 4　歪，ビットレート及び最大深さ（TSVQ）

実験 No.	歪	ビットレート（bits/vector）	最大深さ
1	2280.416438	4.068298	15
2	2293.900371	4.491272	13
3	890.059378	7.984497	18
4	887.403388	9.144775	20
5	618.624811	11.865173	22
6	661.513886	13.670044	24
7	241.457179	4.070663	10
8	247.328300	4.549393	10
9	64.695670	8.018784	17
10	52.937649	9.427307	19
11	23.600917	12.106720	20
12	19.244107	13.412094	20

各々，計算機ファイル名が image11,image12, image13,image14,image15 に対応する。

image11 ～ 15 を図 12（a）～（e）に示す。以下，この系列を USC2 と呼ぶ。

符号化（圧縮）から復号化（復元）に至る手順は，学習系列からコードブック，符号化ファイル，記述ファイルを作成するためのプログラムを実行し，各ファイルを生成し，これらファイルに基づいて TSVQ の分岐条件にしたがってテスト画像についての符号化を行った後，復号化し，テスト画像である原画像との比較などを行う。いずれの学習系列についての実験においても，用いたテスト画像は lena である。lena 画像を図 13 に示す。これらのすべての画像は，南カリフォニア大学（USC）から提供されているもので，実験解析用の画像として広く用いられている。これら画像の大きさは 512×512 画素で，1 画素が 8bits である。

なお，実験において使用した計算機は，主に，Sun4 Sparc Station で，ネットワークは SUNet（Stanford University Network）である。

6.1.2　実験結果

実験データとして，テスト画像である lena 画像の圧縮後の復元画像を各実験 1 ～ 12 において得た（図 14）。また，それら画像の最終的なひずみ，ビットレート及び最大深さ（レベル）を得た（表 4）。実験 1，2 及び実験 11，12 についてのビットレートーひずみ曲線を図 15，16 に示す。

6.2　実験 2（PTSVQ）

6.2.1　実験条件とデータ

実験は，実験 1 と同様に，2 種類の学習系列について，ビットレート及びベクトル・サイズを変えて行った。実験条件を表 5 に示す。実験データは実験 1 と同じである。

符号化から複号化に至る手順は，TSVQ と同じで，分岐は第 5 章に示した条件にしたがう。

6.2.2　実験結果

実験データとして，TSVQ の場合と同様に．テスト画像である lena 画像の圧縮後の復元画像を各実験 13～20 において得た（図 17）。また，それら画像の最終的なひず

表 5　実験条件（PTSVQ）

experiment No.	subblock	bit rate	training sequence
13	2 by 2	8	(usc)image1-5(couple etc.)
14	2 by 2	4	(usc)image1-5(couple etc.)
15	4 by 4	8	(usc)image1-5(couple etc.)
16	4 by 4	4	(usc)image1-5(couple etc.)
17	2 by 2	8	(usc2)image11-15(usc8 etc.)
18	2 by 2	4	(usc2)image11-15(usc8 etc.)
19	4 by 4	8	(usc2)image11-15(usc8 etc.)
20	4 by 4	4	(usc2)image11-15(usc8 etc.)

表 6　歪，ビットレート及び最大深さ（PTSVQ）

実験 No.	歪	ビットレート (bits/vector)	最大深さ
13	64.665862	8.058945	17
14	227.739541	4.301529	15
15	882.680522	7.968201	19
16	1994.606954	4.327820	17
17	54.559753	9.079834	19
18	254.228904	4.641891	13
19	854.468595	9.398682	20
20	2224.014300	4.770508	16

み，ビットレート及び最大深さを得た（表 6）。実験 1，3，5，7 についてのビットレート－ひずみ曲線を図 18，19，20，21 に示す。

7．考 察

7.1 TSVQ（アンバランス・トリー）

図 15，16 ともに，学習系列 USC と USC2 についてのビットレート－ひずみ曲線であり，それぞれ，点線と破線により示している。ビットレートは 12.0 であり，ベクトル・サイズは，図 15 において，2×2，図 16 で，4×4 である。2×2 及び 4×4 のいずれの復元画像についても，ほぼひずみの大きさは似ているが，テスト画像として人物画像 lena を用いているため，couple など人物画像から成る USC の学習系列に基づく TSVQ が，1 枚の人物画像しか含まれていない USC2 に基づく TSVQ より，いずれのベクトル・サイズにおいてもややひずみが小さくなっている。これは，ベクトル量子化による圧縮の場合，適用する画像データ，もしくはそれらと類似の画像データから選択した学習系列に基づいてコードブックを作成すべきであることを示している。したがって，学習系列と量子化対象画像（画像ベクトル）の統計的性質に大きな相違があるときには，この不適合により性能劣化が生じる。相関の強さが弱い画像にも適用可能な汎用性の高いコードブックを作成する方法として，コードブックを画像ごとにあるいは画像中の小部分ごとに更新して行くなどの方法が提案されている。

2×2 が 4×4 よりひずみが小さくなっている。これはビットレートの低い図 14（1）と（7）及び（2）と（8）の比較からも分かる。しかし，一般に，ベクトル量子化においてベクトル・サイズすなわち次元数を大きくするにしたがって，ひずみなどの量子化性能は向上し，いわゆるレートひずみ限界と呼ばれる理論的な情報圧縮限界に近づいて行くことが証明されている。本実験で，4×4 のひずみ向上が確認できなかったのは，より高いビットレートにおいての比較実験が行われなかったからと考えられる。しかし，低いビットレートにおいてもひずみの少ない圧縮が行われていることが分かる。いずれにせよ，最終的な量子化ひずみは原画との差分により与えられる。

ここで，画素あたりのビットレート（bits/pixel＝bpp）r は，レベルすなわち深さが同じとすると，r＝（L bits/vector）×（1vector/（n×n）pixels）から，ベクトル 2×2 が 4×4 の場合の 4 倍のレートになる。例えば，レベル　L が同じ 256 パターンを持つ 8 とすると，2×2 は 2bpp，4×4 は 1/2bpp となり，同じレートで見るには，4×4 についてレベルを 10 にしなければならない。

学習系列が同じであれば，ビットレート R が大きいほど，レベル（最大深さ）が深いほどひずみが小さくなり，復元画像も原画に近いものとなる。これは図 14（1）（3）（5），（2）（4）（6），（7）（9）（11），（8）（10）（12）の各復元画像の比較からも明らかである。ただし，ビットレートを大きくすると，トリーが成長し，同時に計算時間がかかり，そのトリーの情報（コードブックなど）を保存するためのメモリーも増える。なお，レベル　L でのビット R（＝L）における TSVQ と VQ（フル・サーチ）のひずみ計算量を比較すると，VQ ではコードブック中の全ベクトル 2R について計算比較しなければならない。TSVQ では 2R 回のひずみ計算とどちらに分岐するかという R 回の比較でよい。ただし，作成されたコードブックから求められる各画像ベクトルについての圧縮ひずみは VQ に比し，最良のベクトルを得ることができない場合がある可能性を持っている。このことは，トリーの成長の仕方（コードブックの作成法）から説明できる。しかし，ほとんどの応用において，TSVQ が VQ と同じように動作することが経験的にも知られていて，問題はないとされている。これは，量子化性能が学習系列の選択法と量子化対象画像に依存することが明らかなことから，応用として，主に医用画像など相関の強い画像への適用が行われているからと考えられる。

7.2 PTSVQ

図 18 において，点線は TSVQ による結果で，条件はベクトル・サイズ 2×2，ビットレート 12.0，学習系列 USC であり，PTSVQ はビットレート 8.0 のところで行い，そのレートまでのビットレート－ひずみ曲線を破線により示す。同様に，図 19 において，条件はベクトル・サイズ 4×4，ビットレート 12.0，学習系列 USC で，PTSVQ のビットレートは 8.0，図 20 において，条件はベクトル・サイズ 2×2，ビットレート 12.0，学習系列 USC2 で，PTSVQ のビットレートは 8.0，図 21 において，条件はベクトル・サイズ 4×4，ビットレート 12.0，学習系列 USC2 で，PTSVQ のビットレートは 8.0 である。明らかにほとんどのビットレートに渡り，ひずみについて

(a)image 1

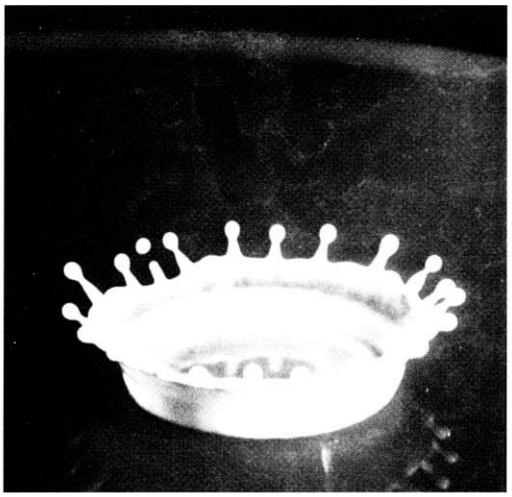
(b)image 2

(c)image 3

(d)image 4

(e)image 5

図 11　学習系列 (USC)

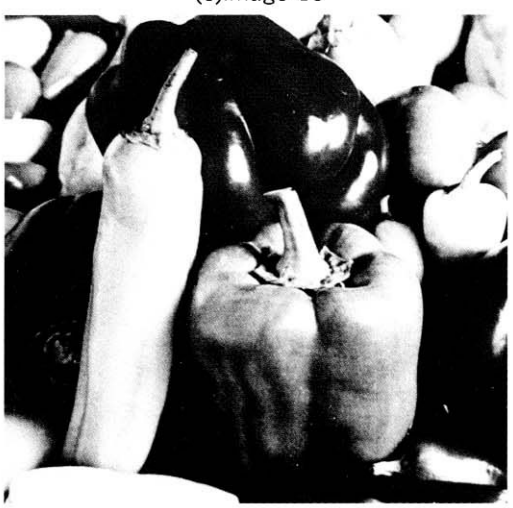


(a)image 11

(b)image 12

(c)image 13

(d)image 14



(e)image 15

図 12　学習系列（USC）

図 13　テスト画像（lena image）

(1) 実験No.1　(2) 実験No.2

(3) 実験No.3　(4) 実験No.4

(5) 実験No.5　(6) 実験No.6

図 14　復元画像（TSVQ）

(7)　実験No. 7

(8)　実験No. 8

(9)　実験No. 9

(10)　実験No.10

(11)　実験No.11

(12)　実験No.12

図 14　復元画像（TSVQ）

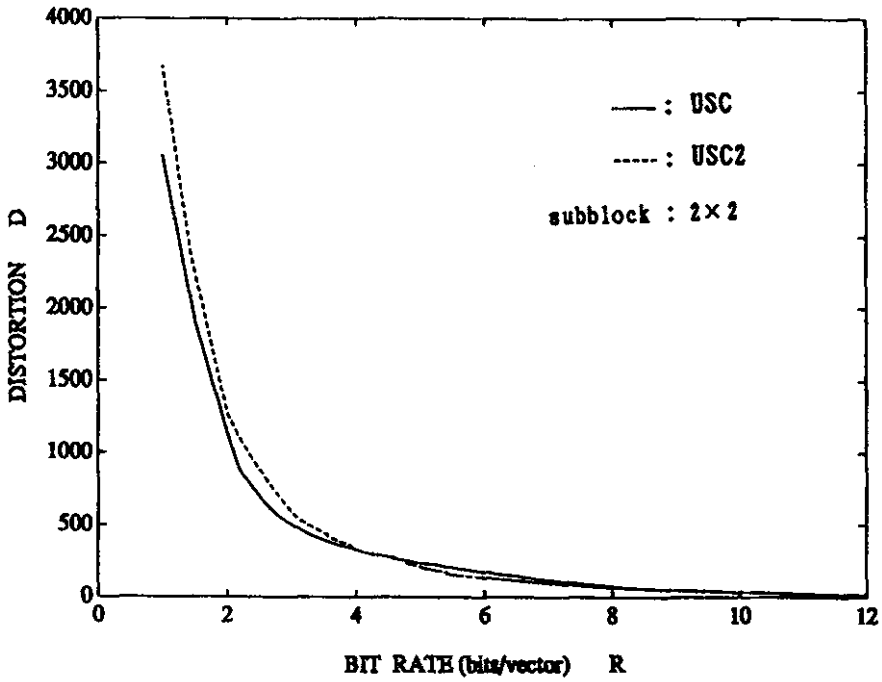


図 15 ビットレート (bits/vector)—ひずみ曲線:実験1,2

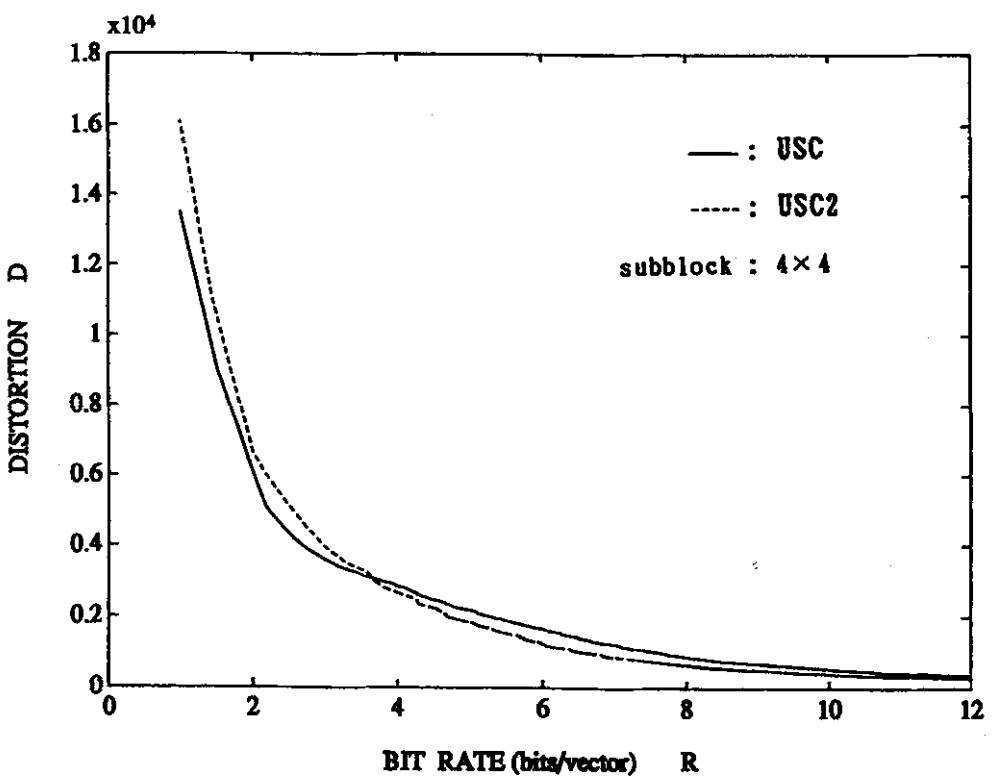


図 16 ビットレート (bits/vector)—ひずみ曲線:実験 11,12

図 17　復元画像 (PTSVQ)

(7) image 7　　(8) image 8

図 17　復元画像 (PTSVQ)

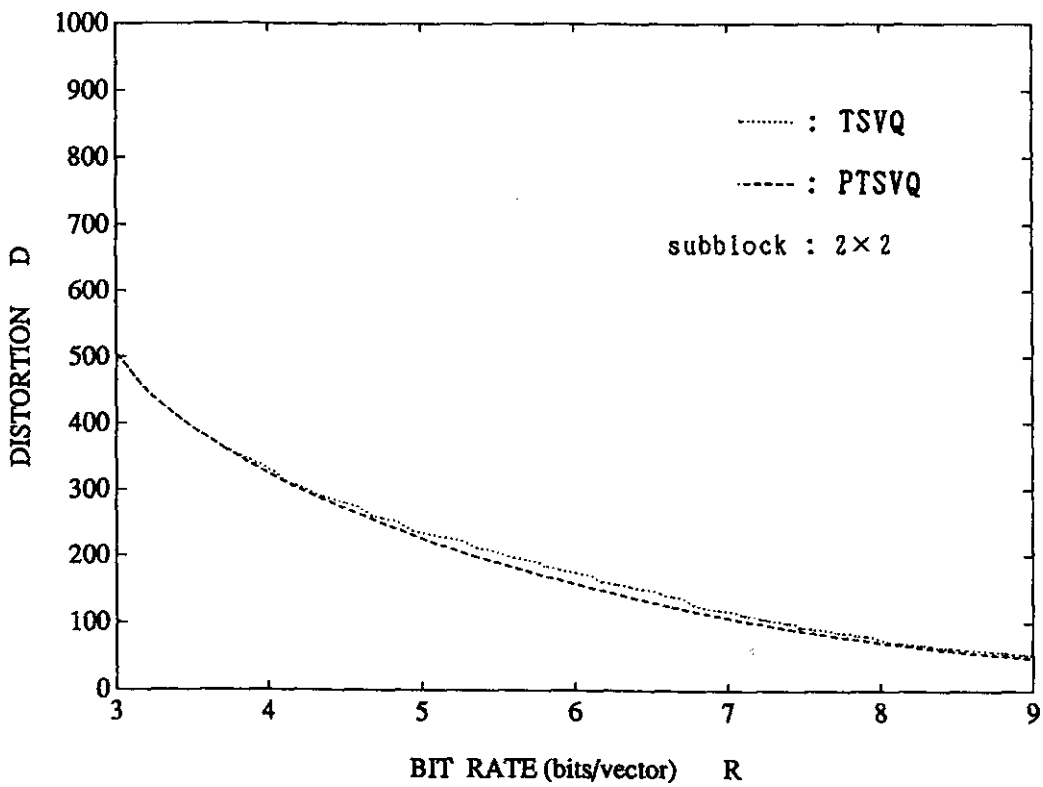


図 18　ビットレート (bits/vector)—ひずみ曲線:実験 13

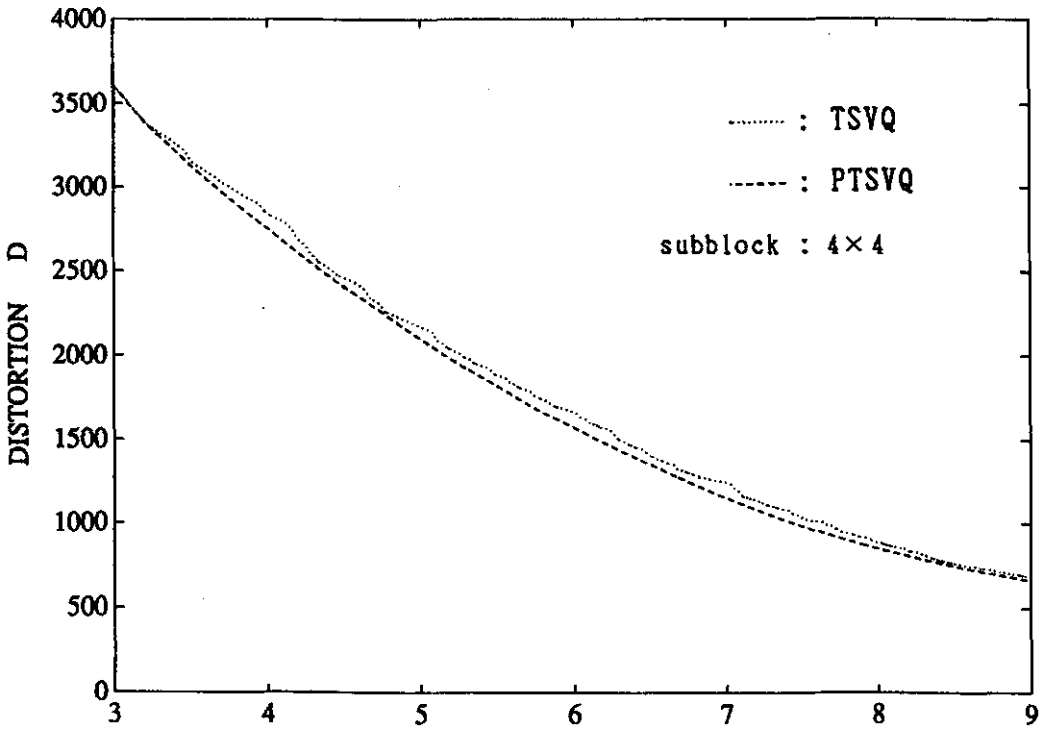


図19 ビットレート (bits/vector)—ひずみ曲線:実験15

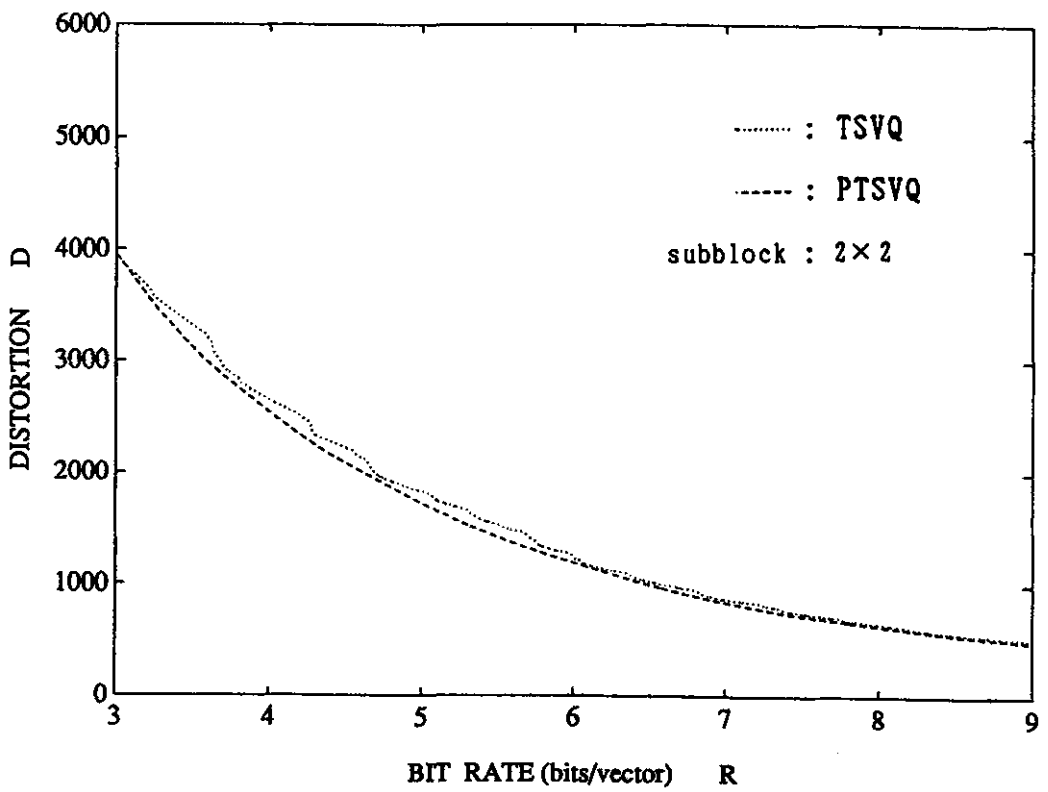


図20 ビットレート (bits/vector)—ひずみ曲線:実験17

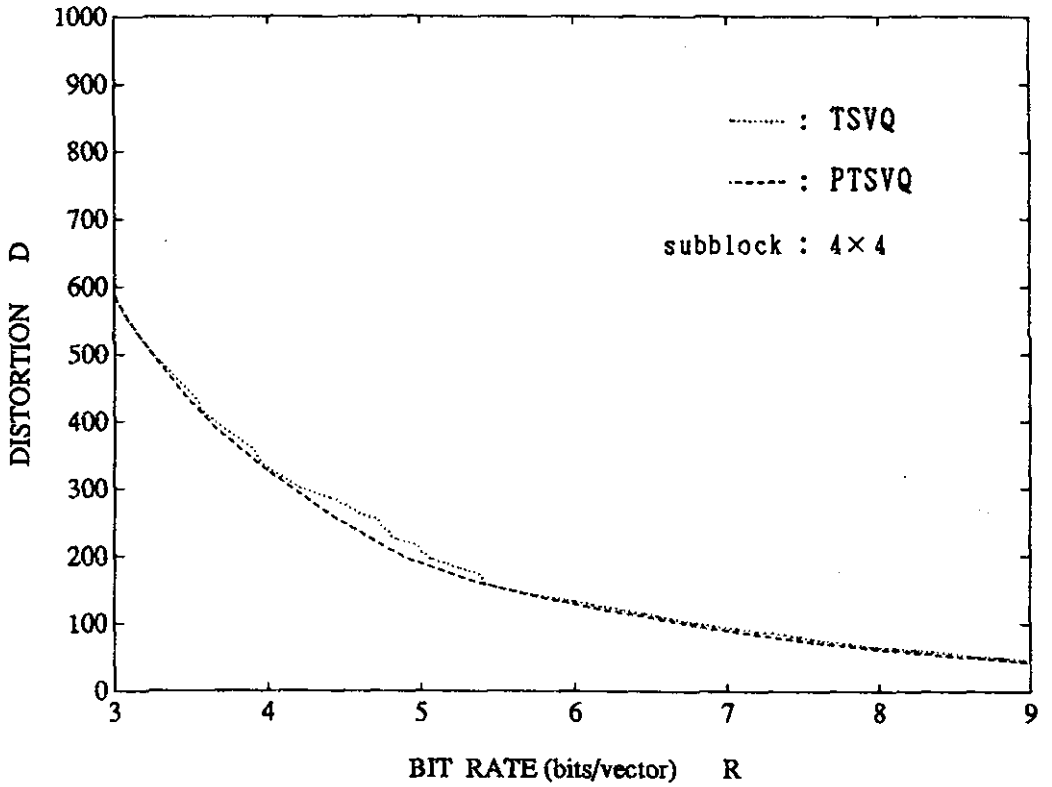


図21 ビットレート (bits/vector)—ひずみ曲線:実験19

PTSVQの方が改善されている。このようにPTSVQがTSVQ（アンバランス・トリー）に比して，常にひずみが改善されるか，等しくなる。また，計算機操作手順でも示されているように，PTSVQは，希望のビットレートで圧縮をかけることができる。そこで，例えば，図18ではビットレート6.0前後のひずみがTSVQに比して改善されていることから，計算時間とひずみを考慮した効率の良い圧縮をかけるには6.0前後のビットレートを選択するとよい。図17（1）～（8）には各復元画像が示されている。TSVQの場合と同様に，USCによるPTSVQがUSC2によるものよりひずみが小さいことが，この復元画像からも分かる。特に，ビットレート4.0における（2）と（6）及び（4）と（8）の比較から明らかである。

なお，ビットレートを8.0と決めても，当然，ビットレートの計算法から明らかなように，そのレート前後で最も8.0に近いところで圧縮が行われる。

PTSVQ及びTSVQ（バランス・スリー）は，いずれもトリー構造による探索ゆえ，VQ（フル・サーチ）に比して，計算量は少なく，同じトリー構造のTSVQ（バランス・トリー）に比してメモリ容量は少なくてよい。

8．まとめ

本研究では，現在，スタンフォード大学において医用画像などへの応用が進められているTSVQとPTSVQを取り上げ，テスト画像により実験検討した。

研究成果を以下に要約する。

1）圧縮の先端的理論であるVQを用いたテスト画像による実験により，VQの基礎及び応用技術を修得し，特に，TSVQ及びPTSVQ応用のシステム開発技術を得た。

2）特定のハードウェアを必要とせず，ソフトウェアだけで高能率圧縮を実行できた。これは新しいマルチメディア規格の提案が可能であることを示す。

3）VQの応用として，ロイド・アルゴリズムを用いた分類と認識が可能であることが明らかになった。

4）画像のような大量のデータを扱うとき，送信及び受信側がコードブックを備えることにより画像圧縮システムを構築でき，それを導入して，高速で，コンパクトなシステム開発が可能であることが明らかに

なった。

ベクトル量子化による符号化法は，予測符号化や直交交換符号化など他の手法に比べて比較的単純な手法であるにもかかわらず，画像を含めいろいろな情報源に対して優れた符号化性能を示すことが明らかになっている。このことからも，上記成果及び今後の研究成果を道内企業における画像圧縮・認識システム開発に積極的に応用して行きたい。また，ベクトル量子化の画像圧縮・復元への適用は，1980 年頃から進められてきており，現在においても研究が続けられている技術であり，今後のさらなる進展が期待されている。

今後，本情報圧縮技術の応用として，動画像のファイリング・システム，ステレオ視システム，文字認識システム，新しいマルチメディア規格の提案，ロイド・アルゴリズムを用いた識別分類，プラットホームとしてのデータ圧縮・復元，MRI 医用画像システムなど，広範囲の技術展開が可能であり，特定のハードウェアを必要としないで画像の圧縮 / 伸張を実行できることから，データを入出力する全てのシステムへの応用が可能である。

画像情報圧縮のレベルは，画像の認識技術を駆使することにより，さらに大幅な情報圧縮を実現しようとする試みもあり，画像の高能率符号化は，今後とも先端技術であり続けると考えられる。

9．謝 辞

本研究は，企画振興部平成 3 年度長期海外研究事業において，客員研究員として 7 ヶ月間に渡り在籍した米国スタンフォード大学電気工学科において行った研究をまとめたものである。

本研究遂行に当たり，当学科教授 Robert M.Gray，研究助手であり PhD の学生である Pamela Cosman,Karen L.Oehler,Sang Ju Park の各氏には有益な御指導，御助言をいただき，ここに深く感謝いたします。また，本研究の機会を与えていただきました丸山場長はじめ試験場の皆さん，企画振興部及び商工労働観光部の担当者の方々に深く感謝いたします。

参考文献

1）原島博監修；”画像情報圧縮”，オーム社，pp115－139(1992)

2）中田和男；”音声の高能率符号化”，森北出版，pp34－42

3）Robert M.Gray：”Vector Quantization”，IEEE ASSP Mag.,1,2,pp.4 － 28（Apr.1984）

4）Yoseph Linde,Andres Buzo and Robert M.Gray: ”An Algorithm for Vector Quantizer Design”,IEEE Trans.Commun.,COM－28,1,pp84－95（Jan.1980）

5）Allen Gersho and Robert M． Gray：”VECTOR QUANTIZATION AND SIGNAL COMPRESSION” ,KLUWER ACADEMIC PUBLISHERS，PP410－423（1991）

6）Philip A．Chou，Tom Lookabaugh and Robert M．Gray:”Optimal Pruning with Application to Tree-Structured Source Coding and Modeling”,IEEE Tran. on Info. Theory, Vol.35, No.2, pp299 － 315（March 1989）

7） Robert M， Gray and Eve A． Riskin：”Tree-Structured Vector Quantization for Progressive Transmission Image Coding”,Supported by ESL etc,.Vol.129,pp142－150（1989）