

データ圧縮機能を内蔵した 画像データベース・システム

山田 芳郎 河本 幸生 田崎 三郎
愛媛大学工学部

あらまし データ圧縮機能を内蔵した汎用的な画像データベース・システムの概要を示し、そこで画像データ圧縮手法として用いられる「再帰的な差分ベクトル量子化手法」について述べている。

IMAGE DATABASE SYSTEM WITH EMBEDDED DATA COMPRESSION FUNCTIONS

Yoshio Yamada, Kohsei Kawamoto, and Saburo Tazaki

Faculty of Engineering, Ehime University
Matsuyama 790, Japan

Abstract This paper discusses an image database system with embedded data compression functions. As an data compression scheme, we propose a differential vector quantization algorithm with recursive codebook structure.

1. まえがき

CD-ROMや光磁気ディスクなどランダム・アクセス可能な大容量記憶装置が低廉に供給されるようになってきた⁽¹⁾。それに伴って、データベースの新しい時代が訪れようとしている。それは、データベースのパーソナル化、データベースのメディア化である。静止画、動画、音声、音響、音楽、文書などを含むマルチメディア・データベースが情報メディアとして流通する可能性も技術的なレベルでは十分現実的なものになってきている。

筆者らは、従来から音声および画像のデータ圧縮（高能率符号化）に関する研究を行っている。また、これらのデータ圧縮技術の応用分野として、通信はいうまでもなく、マルチメディア・データベースにも大きな関心を持っている。マルチメディア・データベース・システムは、データ圧縮機能を自然な形で取り込んだものとして構築されるべきであると考えている。

本稿では、筆者らが構築しようとしているデータ圧縮機能を内蔵した汎用的な画像データベース・システムの概要を示す。また、本データベース・システムにおいて、標準データ圧縮手法の一つとして用いられる「再帰的な差分ベクトル量子化手法」について述べる。

2. 画像データベースと画像符号化

データ圧縮機能を内蔵した汎用的な画像データベース・システムを構築するために、現在のところデータ圧縮技術の観点からシステムの内部仕様の検討を行っている。この画像データベース・システムは、以下に示すような特徴を持ったものとなる予定である。

(1) 複数のデータ圧縮手法の混在を許容する。

一言で画像と言っても様々な形態がある。たとえば、モノクロ画像の場合、階調数によって、

2値、中間調（～16値程度）、多階調のいわゆる自然画像（～256値程度が一般的であるが医用画像などで1024値の例もある）などに分類できる。符号化効率の観点から、それぞれの階調数に適した符号化手法を選択する必要がある。

カラー画像の場合には、階調数に関してはモノクロ画像の場合と同様であるが、さらに色彩の表現方法（RGB、YUV、HVCなどの表色系）によって各色成分の統計的性質が異なる。また、解像度も符号化効率に影響を与える大きな要因の一つである。

以上のことから、汎用の画像データベース・システムとしては、対象画像によって最適な（あるいは準最適な）符号化方式を選択できるような柔軟性のあることが重要となってくる。さらに、次の段階としてデータベースのマルチメディア化を実現するためには、複数の符号化方式の存在を許容することは必須の事項といえる。そこで我々としては、システム側で標準的に用意される複数個のデータ圧縮手法以外にも、ユーザが独自に定義したデータ圧縮手法をも組み込むことのできる柔軟な画像データベース・システムの構築を目指している。

(2) 複数のインプリメントを許容する。

データ圧縮手法の研究開発および保守などの観点から、一つのデータ圧縮手法（アルゴリズム）に対して、複数のインプリメントを許容することを考えている。たとえば、ソフトウェアによるインプリメントとハードウェアによるインプリメントの両方が同一のデータベース・システム上で混在して利用できるようにしたい。このような機能を持たせることによって、符号化・復号化のための専用ハードウェアを持たないワークステーションあるいはパーソナルコンピュータとハードウェア化された専用画像端末とを目的に応じて使い分けることができる。た

だし、実用に当たっては、アルゴリズムの同一性の検証や配付の手段を確立しておく必要がある。

(3) ネットワーク環境での利用を考慮する

ネットワーク上でデータベースを利用する場合には、ネットワーク上のトラフィック量を低減するために、符号化（圧縮）された画像データがネットワーク（通信媒体）上を流れるようにデータベース・システム構築すべきである。このためには、たとえばクライアント-サーバ型のモデルを採用し、クライアント側において符号化および復号化の処理を行うようにすればよい。

(4) 順次再生機能を持たせる

ここで言う順次再生とは、符号化（圧縮）された画像データの一部を復号化することで、たとえば図1(a), (b), および(c)に示すように、

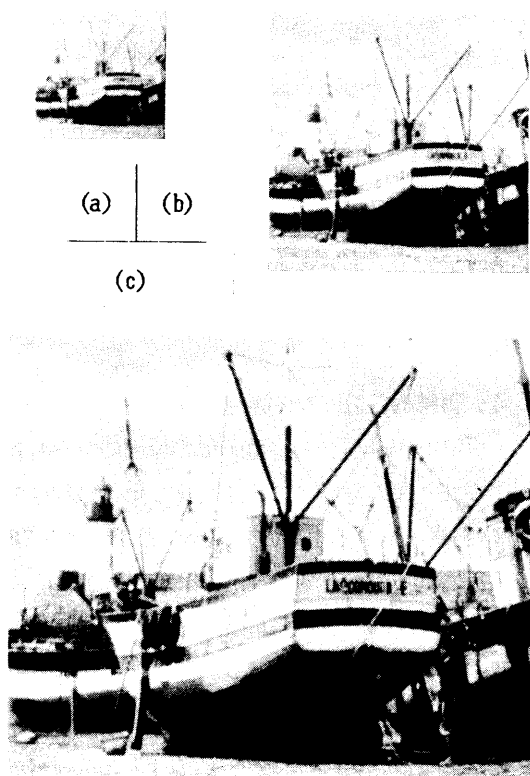


図1 解像度に関する順次再生の一例

低い解像度を持ったアウトライン画像から原画と同じ解像度を持ったディテイル画像へと何段階かの階層的な再生を行うことを指している。従って、復号化を途中で打ち切っても、その時点における再生品質（あるいは解像度）でもって画像全体を把握できることになる。順次再生の機能があれば、対話的な検索を行う環境において、目的の画像データに行き着くまでの総合的な検索応答時間を短縮する効果が期待できる。順次再生する属性としては、上述の解像度の他にも彩度（モノクロ画像からカラー画像への順次再生）や階調数（2値画像から自然画像への順次再生）などが考えられる。

3. ベクトル量子化に基づく画像符号化

自然画像を対象とした標準的なデータ圧縮手法として考慮しておくべき方式のクラスとしては、以下のものが挙げられる。(1)予測符号化、(2)直交変換符号化、(3)ベクトル量子化、(4)算術符号化、(5)その他のブロック符号化（BTCなど）である。それぞれが多くの変種を持っている。たとえば、直交変換符号化であれば、コサイン、アダマールなどの直交変換行列の違い、変換の次数（ブロックサイズ）、変換係数の量子化方法（量子化ビット数の割当方法などを含む）、適応的な符号化を行うのであればその方法、などによる多くのバリエーションが存在する。また、順次再生やランダムアクセス性（符号化画像の部分アクセス性）を実現する場合には、それなりにインプリメントの方法も変わってくる。

我々は、静止画および動画を対象としたデータ圧縮手法として、再帰的差分ベクトル量子化手法を現在開発中である。次章でその概念と具体例を述べるが、ここではその前にベクトル量子化を用いた画像符号化の概要をまとめておく。

ベクトル量子化の符号化および復号化は、一般に次式のように定義できる。

$$Q_E(\underline{x}) = i \quad \text{if } \underline{x} \in S_i \quad (1)$$

$$Q_D(i) = \underline{y}_i \quad (2)$$

ここに、 $i = 0, 1, \dots, N-1$ であり、 \underline{x} は情報源出力（あるいは量子化入力）ベクトルである。モノクロの静止画像の場合、たとえば $M \times M$ 画素からなる矩形のブロックに画像を分割し、ブロック内の各画素の明度値を表すベクトルを入力ベクトル \underline{x} とする。この場合 \underline{x} は、 M^2 次元ベクトルとなる。 $\{S_i\}$ は情報源空間の分割、 $\{\underline{y}_i\}$ は再生アルファベットを表す。なお、 $Q_D(Q_E(\underline{x})) = \underline{y}_i$ のとき、 \underline{y}_i を（入力ベクトル \underline{x} に対する）量子化出力ベクトルと言うことがある。

式(1)の符号化関数 Q_E は、入力ベクトル \underline{x} が部分空間 S_i に属するとき、 S_i の代表元 \underline{y}_i のインデクス i が出力されることを表している。実際には、予め与えられた忠実度規範に基づいて、

$$Q_E(\underline{x}) = i \quad \text{if } \|\underline{x} - \underline{y}_i\| \leq \|\underline{x} - \underline{y}_j\| \quad \text{for all } j \neq i \quad (3)$$

のように符号化関数 Q_E を定義する場合も多い。ここに $\|\cdot\|$ は情報源空間において定義されるノルムである。画像符号化の場合には、ユークリッド・ノルムが比較的良好に用いられる。

一方、式(2)の復号化関数 Q_D は、上述のようにして求められたインデクス i をアドレスとして、再生アルファベットの表から再生ベクトル \underline{y}_i を読み出す操作に過ぎない。一般にデータベースにおいては「書き込み」に比べて「読み出し」の頻度が高らかに多いから、復号化の単純さは大きなメリットとなり得る。

上述の単純な定義に基づくベクトル量子化においては、画素当りの平均符号化ビット数 R を

一定とすると、入力ベクトルの次元数 K が大きくなるにつれて再生アルファベットのサイズ N ($=2^{KR}$) は指数関数的に増大する。符号化の際の計算量に関しては、木探索手法などを適用することで $K \cdot \log N$ に比例する程度に削減できるが、 N が大きくなると再生アルファベットを設計すること自体がきわめて困難になる。このため、 $KR = 8 \sim 12$ [bits] 程度のベクトル量子化器がこれまでに設計されているに過ぎない。しかし静止画像の場合、その統計的性質（自己相関など）を考慮すると、 $M = 8$ 程度以上にすることが望ましい ($K = M^2 = 64$)。また、順次再生の最終段階で視覚的に無ひずみと見なし得るような高品質の再生画像を得ようとするれば、 $R = 1 \sim 2$ [bits] 程度が必要となる。すなわち、 $N = 2^{64} \sim 2^{128}$ 程度となる。この様に巨大な再生アルファベットは直接的には実現できないし、再生アルファベットを記憶しておかなければならないことを考えると現実的に無意味と言える。このため、実現の容易なサイズの小さい再生アルファベットを用いて、上述のような巨大なサイズの再生アルファベットを等価的に構成する組織的なベクトル量子化手法を開発することが重要となる。そこで筆者らは、次章で示すような再帰的差分ベクトル量子化手法を提案する。

4. 再帰的差分ベクトル量子化

再帰的差分ベクトル量子化の原理的な符号化アルゴリズムを以下に示す。符号化の処理は $M \times M$ 画素からなる矩形のブロックを単位として実行されるものとする。

1次元配列 $T[n]$; $1 \leq n \leq 2M^2 - 1$ を次式のように定義する。

(i) $1 \leq n \leq M^2 - 1$ について

$$T[n] = (T[2n] + T[2n+1]) / 2 \quad (4)$$

(ii) $M^2 \leq n \leq 2M^2 - 1$ について

$$T[n] = v[n - M^2] \quad (5)$$

ここに、 $v[m] ; 0 \leq m \leq M^2 - 1$ は、図2に示されるような順序で整列された各画素の明度値を表す。同図は、 $M = 8$ の場合の例を示しているが、ここで用いている順序付けの定義については文献(2)を参照されたい。式(4)、(5)の1次元配列 $T[n]$ は、例えば $T[n]$ に対応する節点（以下では第 n 番目の節点という）が $T[2n]$ と $T[2n+1]$ に対応する節点の直接の親であるような2進木のデータ構造を表現している。従って、根に対応する $T[1]$ はブロック内の全画素の平均の明度値を表す。また $T[2]$ および $T[3]$ は、ブロックの上半分と下半分の画素の平均の明度値をそれぞれ表している。本手法においては、この様に定義される2進木のデータ構造を用いて、その根から終端節点へ向かって再帰的に符号化を進める。以下では、第 n 番目の節点について、その子孫の終端節点に対応する画素の明度値をすべてある値 τ で置き換えたときの絶対値誤差の最大値を $D(n; \tau)$ で表すことにする。

まず、 $T[1]$ をスカラー量子化して、 $T'[1]$ を得る。次に、予め与えられたしきい値 ϵ に対して、

$$D(1; T'[1]) < \epsilon \quad (6)$$

が満たされるならば、ここで符号化を打ち切る。このとき、ブロック内の全画素は $T'[1]$ で近似される。逆に

$$D(1; T'[1]) \geq \epsilon \quad (7)$$

ならば、差分ベクトル

$$(T[2] - T'[1], T[3] - T'[1]) \quad (8)$$

がベクトル量子化される。式(8)の入力ベクトルに対する量子化出力ベクトルを

$$(T''[2], T''[3]) \quad (9)$$

で表すと、この時点での復号化出力は

$$(T'[2], T'[3]) = (T''[2] + T'[1], T''[3] + T'[1]) \quad (10)$$

で与えられる。

一般に、第 n 番目の節点について、

$$D(n; T'[n]) < \epsilon \quad (11)$$

が成立すれば、その節点までで符号化を打ち切る。このとき、その節点の子孫の終端節点に対応する画素はすべて $T'[n]$ で近似される。逆に

$$D(n; T'[n]) \geq \epsilon \quad (12)$$

ならば、次のステップに進んで

$$(T[2n] - T'[n], T[2n+1] - T'[n]) \quad (13)$$

がベクトル量子化される。式(13)の入力ベクトルに対する量子化出力ベクトルを

$$(T''[2n], T''[2n+1]) \quad (14)$$

で表すと、

$$(T'[2n], T'[2n+1]) = (T''[2n] + T'[n], T''[2n+1] + T'[n]) \quad (15)$$

となる。

上述のアルゴリズムは2進木に基づいたものであるが、4進の場合も同様に定義できる。この場合の符号化手順を、2進の場合と同じ記号を用いて表現すれば以下ようになる。すなわち、第 n 番目の節点について、

$$D(n; T'[n]) < \epsilon \quad (16)$$

が成立すれば その節点までで符号化を打ち切

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 1 | 2 | 5 | 6 | 17 | 18 | 21 | 22 |
| 3 | 4 | 7 | 8 | 19 | 20 | 23 | 24 |
| 9 | 10 | 13 | 14 | 25 | 26 | 29 | 30 |
| 11 | 12 | 15 | 16 | 27 | 28 | 31 | 32 |
| 33 | 34 | 37 | 38 | 49 | 50 | 53 | 54 |
| 35 | 36 | 39 | 40 | 51 | 52 | 55 | 56 |
| 41 | 42 | 45 | 46 | 57 | 58 | 61 | 62 |
| 43 | 44 | 47 | 48 | 59 | 60 | 63 | 64 |

図2 1次元化のための画素の整列順序

る。このとき、その節点の子孫の終端節点に対応する画素はすべて $T'[n]$ で近似される。逆に

$$D(n; T'[n]) \geq \epsilon \quad (17)$$

ならば、

$$\begin{aligned} & (T[4n] - T'[n], \\ & T[4n+1] - T'[n], \\ & T[4n+2] - T'[n], \\ & T[4n+3] - T'[n]) \end{aligned} \quad (18)$$

なる4次元ベクトルがベクトル量子化される。式(18)の入力ベクトルに対する量子化出力ベクトルを

$$(T''[4n], T''[4n+1], T''[4n+2], T''[4n+3]) \quad (19)$$

で表すと、

$$\begin{aligned} & (T'[4n], T'[4n+1], T'[4n+2], T'[4n+3]) \\ & = (T[4n] + T'[n], \\ & T[4n+1] + T'[n], \\ & T[4n+2] + T'[n], \\ & T[4n+3] + T'[n]) \end{aligned} \quad (20)$$

となる。 $T'[1]$ の定義は2進の場合と同じである。

上述の再帰的差分ベクトル量子化の符号化アルゴリズムにおいて、2次元ないし4次元ベクトル量子化の部分には全探索や木探索ベクトル量子化⁽⁹⁾あるいは逐次近似ベクトル量子化⁽²⁾などの従来のベクトル量子化手法を用いることができる。また、どの節点で符号化を打ち切ったかを示す補助情報の符号化には、木を符号化するための種々の手法を適用できる。

本手法は、上述のように階層的なサブサンブ

リング構造を持っている。従って、符号化出力情報の順序を上位の階層から順に並べ直すことによって、容易に順次再生が実現できる。すでに示した図1の再生画像は、このようにして実現した順次再生の一例である。同図において、(a)は約8%、(b)は約32%、(c)は100% (画素当り約2.0ビット) 復号化された再生画像である。

5. むすび

ベクトル量子化に基づくデータ圧縮機能を内蔵した汎用的な画像データベース・システムの概要について述べた。また、そこで用いられる再帰的差分ベクトル量子化手法の符号化アルゴリズムを示した。現在、再生アルファベット構成法についての詳細な検討を行っている。これについては、稿を改めてご報告したい。

参考文献

- (1) "第一歩を踏み出した光磁気、高密度FDD, DAT", 日経バイト, NO.57, pp.142-158 (1989).
- (2) 山田, 田崎: "逐次近似ベクトル量子化", 信学論, J68-A, 5, pp.466-473(1986).
- (3) R.M.Gray, Y.Linde: "Vector quantizers and predictive quantizers for Gauss-Markov sources," IEEE Trans. Commun., COM-30,2, pp.382-389(1982).