

# 画像の圧縮

画像のデータ量は膨大である。テレビ画像は約 500 本の走査線で構成される。これを縦 480、横 640 の画素で表し、1 画素 8 ビット (1B) で表せば、その情報の量は約 300kB である。この動画像を蓄えたり、高速に送るために、画像を圧縮できたら都合がよい。

ファックスは白黒の 2 値画像であるが、標準的には 1mm 当たり 8 点で表示される。したがって、A4 版の原稿 1 枚は約 4000K ビット (500KB) となる。現在ファックスは圧縮されているおかげで高速に送ることができる。

画像の圧縮は以下の 2 つのステップからなる。

1. 画像データを圧縮された別なデータ (変換信号) に変換する。
2. 変換信号を適当な符号に変換する。

第 1 のステップは、画像特有の性質を利用してデータを圧縮する。第 2 のステップは、画像に依存しない一般的な符号化であり、符号化理論で扱っているのだから、ここでは取り上げない。第 1 のステップは、圧縮を行なっていて、厳密には符号化ではないが、符号化ともいわれている。

画像を圧縮して符号化することに対し、元の画像を復元することを復号化という。符号化は、元の画像を完全に復元できる可逆符号化と、必ずしも完全に復元できない不可逆符号化がある。

ファックスのような 2 値画像の圧縮には可逆符号化が使われている。濃淡画像やカラー画像、動画像に対しては、不可逆符号化が使われることが多い。

## 1 ランレングス符号化

文書のような 2 値画像は、白い紙に黒で細い線が書かれている部分が多い。1mm 当たり 8 点で走査すれば、白い点あるいは黒い点が連続することが多い。したがって、画像を走査する順に、同じ点が続いたかを符号化する。例えば、白を 0、黒を 1 とし、

```
000000000011110000011110000000
```

は「白 10、黒 4、白 5、黒 4、白 7」と書ける。さらに、必ず白と黒が繰り返すので「白 10、4、5、4、7」で十分である。

白あるいは黒のつながりをラン (run) といい、ランの長さを符号化するのでランレングス (run length) 符号化という。普通の文書画像には、この例より長いランがたくさんあるので、符号の長さの合計を 1/7 程度にすることができる。

## 2 予測符号化

一般に、画像は特定の性質をもっているのだから、その性質を利用して予測することができる。例えば、ほとんどの部分で、隣合う画素値は類似の値をもっている。予測符号化は、既知の画素値から現在の画素値を予測し、真の値と予測値との差を符号化する方法である。真の値と予測値の差である予測誤差が小さければ、符号を短くできる。すなわち、予測符号化は予測ができる画像に対して効果がある。

予測に用いる既知の画素は、画像をどのような順で走査するかに依存する。例えば、画像を左から右へ走査し、その走査を上から順に行なう場合は、現在の画素の上の画素と同じ走査線の左

の画素が既知である．予測には既知の画素をすべて用いるのではなく，現在の画素の近傍だけを用いる．予測に用いる画素を参照画素という．参照画素が同じ走査線上の画素だけの場合を1次元予測，他の走査線上の画素も含まれる場合を2次元予測という．動画像の場合は，1つ前の画像（フレーム）も参照することができる．これをフレーム間予測という．これに対して，同一フレーム内の画素だけの場合をフレーム内予測という．

予測の方法の代表例は線形予測である．これは，現在の画素値を参照画素値の線形関数として予測する．簡単のため，現在の画素値の予測  $\hat{f}(i)$  を1次元予測で求めるとし，参照画素はその直前の  $k$  個の画素とすれば，

$$\hat{f}(i) = c_1 f(i-1) + c_2 f(i-2) + \cdots + c_k f(i-k) \quad (1)$$

ここで， $c_1, c_2, \dots$  を予測係数という．

例えば， $\hat{f}(i) = f(i-1)$  は，直前の値と同じであると予測し，その差を符号化する．この方法は，DPCM (Difference Pulse Code Modulation) として実際に用いられている．また，滑らかに変化する場合は，以下の傾斜予測が用いられる．

$$\hat{f}(i) = 2f(i-1) - f(i-2) \quad (2)$$

予測符号化の効率を上げるために，必ずしも原画像を復元しなくてもよい不可逆符号化を行なう場合を考えよう．人間は，変化の少ない部分では画像の誤差に敏感であるが，変化の多いところでは少しの誤差には気がつかないという性質がある．そこで，予測誤差の大きい画素に対しては，復元結果に誤差を許すことにする．例えば，予測誤差を以下の値で代表する．

$$-8, -4, -2, -1, 0, 1, 2, 4, 8$$

すなわち，実際の予測誤差に一番近い代表値を予測誤差とする．予測誤差を定められた値のいずれかにすることを量子化というこのようにすると，誤差が生じることがある．この誤差を量子化誤差という．

符号化した後でノイズが入ると以後の復元に誤差が生じてしまう．そこで，ノイズが入るおそれのある場合には，周期的に（例えば走査線の最初に）画素値自体を与えることにすればよい．

## 3 フーリエ変換

### 3.1 1次元フーリエ変換

もし信号  $f(x)$  が周期信号ならば，フーリエ級数展開によって，その周期の整数倍の周期の周波数の成分を求めることができる．信号  $f(x)$  が周期信号でない場合は，フーリエ級数展開ができない．ここで述べるフーリエ変換 (Fourier transform) は非周期信号でも可能である．

いま  $f(x)$  のフーリエ変換を  $F(j\omega)$  とすれば，

$$F(j\omega) = \int_{-\infty}^{\infty} f(x) e^{-j\omega x} dx \quad (3)$$

$F(j\omega)$  は，角周波数  $\omega$  の成分を表す．

もとの信号  $f(x)$  は，以下のフーリエ逆変換 (inverse Fourier transform) によって再構成できる．

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(j\omega) e^{j\omega x} d\omega \quad (4)$$

フーリエ逆変換もフーリエ変換と類似の形であり，その性質も同じである．

### 3.2 離散フーリエ変換

これまで、 $f(x)$  がアナログ信号であるとみなしてきたが、デジタル信号処理では、信号  $f(x)$  を一定間隔ごとにサンプルする場合を扱う。信号  $f(x)$  を間隔 1 でサンプルした信号を、以下のようにデルタ関数  $\delta(x)$  を用いて表す。

$$\begin{aligned} f_s(x) &= f(0)\delta(x) + f(1)\delta(x-1) + \cdots + f(N-1)\delta(x-(N-1)) \\ &= \sum_{n=0}^{N-1} f(n)\delta(x-n) \end{aligned} \quad (5)$$

このフーリエ変換は、

$$\begin{aligned} F_s(j\omega) &= \int_{-\infty}^{\infty} \left\{ \sum_{n=0}^{N-1} f(n)\delta(x-n) \right\} e^{-j\omega x} dx \\ &= \sum_{n=0}^{N-1} f(n) \int_{-\infty}^{\infty} \delta(x-n) e^{-j\omega x} dx \\ &= \sum_{n=0}^{N-1} f(n) e^{-jn\omega} \end{aligned} \quad (6)$$

信号の時間長は  $N$  であるので、最小の角周波数は  $\omega_0 = 2\pi/N$  である。最大の角周波数  $\Omega_0$  は周期 1 に対応するので、 $\Omega_0 = 2\pi$  である。角周波数が  $\omega_0$  の整数倍におけるフーリエ変換は

$$F_s\left(jk\frac{2\pi}{N}\right) = \sum_{n=0}^{N-1} f(n) e^{-jkn\frac{2\pi}{N}}, \quad k = 0, 1, \dots, N-1 \quad (7)$$

このフーリエ変換は、数値計算で求めることができる。

フーリエ変換は、 $k = 0$  と  $k = N$  で同じ値になるので、 $\omega = 0$  から  $\omega = \Omega_0$  まで、 $\omega_0$  の刻みで得られることになる。これを離散フーリエ変換 (discrete Fourier transform, DFT) という。

離散フーリエ逆変換 (inverse DFT, IDFT) は以下となる。

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F_s\left(jk\frac{2\pi}{N}\right) e^{j\frac{2\pi}{N}kn}, \quad n = 0, 1, \dots, N-1 \quad (8)$$

簡単のため

$$\left. \begin{aligned} F(k) &\triangleq F_s\left(jk\frac{2\pi}{N}\right) \\ W &\triangleq e^{-j\frac{2\pi}{N}} \end{aligned} \right\} \quad (9)$$

とおけば、以下の離散フーリエの基本公式が得られる。

$$F(k) = \sum_{n=0}^{N-1} f(n) W^{kn}, \quad k = 0, 1, \dots, N-1 \quad (10)$$

$$f(n) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) W^{-kn}, \quad n = 0, 1, \dots, N-1 \quad (11)$$

ここで、離散フーリエ変換の意味を考えてみよう。離散信号  $f(n)$  を  $N$  次元のベクトル  $\mathbf{f} = (f(0), f(1), \dots, f(N-1))^T$  と表す (ここで、 $^T$  は、転置を表す)。また、 $W^0, W^k, \dots, W^{(N-1)k}$  をつぎのようにベクトルで表す。

$$\mathbf{W}_k = (W^0, W^k, \dots, W^{(N-1)k})^T, \quad k = 0, 1, \dots, N-1 \quad (12)$$

式(10)は、ベクトルの内積の形をしている。一般に複素ベクトル  $X$  と  $Y$  の内積  $(X, Y)$  は、 $Y$  の共役ベクトルを  $\bar{Y}$  と表せば

$$(X, Y) = X^T \bar{Y}$$

式(10)を  $W_k$  と  $f$  を用いて表せば、 $f$  は実数ベクトルであるから

$$F(k) = W_k^T f = (W_k, f) \quad (13)$$

すなわち、 $F(k)$  は、 $W_k$  と  $f$  との内積である。

この  $W_0, W_1, \dots, W_{(N-1)}$  は直交ベクトルである。なぜなら、複素ベクトル  $W_k$  と  $W_l$  の内積は以下の性質をもつ。

$$(W_k, W_l) = \begin{cases} 0, & k \neq l \\ N, & k = l \end{cases} \quad (14)$$

以上より、離散フーリエ変換は  $f$  の  $W_0, W_1, \dots, W_{(N-1)}$  への投影とみなすことができる。この  $W_0, W_1, \dots, W_{(N-1)}$  を直交基底という。

つぎに、離散信号  $f(n)$  と同様に、離散フーリエ変換の結果を  $N$  次元のベクトル  $F = (F(0), F(1), \dots, F(N-1))^T$  と表す。また、 $W_{-n}$  を以下で定義する。

$$W_{-n} = (W^0, W^{-n}, \dots, W^{-(N-1)n})^T, \quad n = 0, 1, \dots, N-1 \quad (15)$$

式(11)は

$$f(n) = \frac{1}{N} W_{-n}^T F = (W_{-n}, \bar{F}) \quad (16)$$

この  $W_0, W_{-1}, \dots, W_{(-N-1)}$  も直交ベクトルである。

どうようにして、信号  $f$  は  $F$  の  $W_0, W_{-1}, \dots, W_{(-N-1)}$  への投影である。今度は、信号  $f$  が、直交基底  $W_0, W_{-1}, \dots, W_{(-N-1)}$  によって表されている。

ところで、信号  $f$  は、直交基底  $W_0, W_{-1}, \dots, W_{(-N-1)}$  によって完全に表されるが、サンプル数  $N$  が大きくなると計算量が多くなる。普通は、適当な数の直交基底を用いて近似する。もし、信号  $f$  の角周波数  $M\omega_0$  以上の成分が十分小さければ ( $M < N$ )、直交基底を  $W_0, W_1, \dots, W_{(M-1)}$  とすることができる。

### 3.3 2次元への拡張

つぎに二次元信号  $f(x, y)$  のフーリエ変換を  $F(ju, jv)$  とする。

$$F(ju, jv) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j(ux+vy)} dx dy \quad (17)$$

ここで、 $u, v$  は、それぞれ  $x, y$  方向の角周波数を表す。

もとの信号  $f(x, y)$  は、以下のフーリエ逆変換によって再構成できる。

$$f(x, y) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(ju, jv) e^{j(ux+vy)} du dv \quad (18)$$

同様に、以下のような2次元離散フーリエの基本公式が得られる。

$$F(u, v) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f(m, n) W^{um+vn}, \quad u, v = 0, 1, \dots, N-1 \quad (19)$$

$$f(m, n) = \frac{1}{N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) W^{-um-vn}, \quad m, n = 0, 1, \dots \quad (20)$$

なお，逆変換の場合だけ  $N^2$  で割っているが，以下のように割算を均等に配分することもできる．

$$F'(u, v) = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f(m, n) W^{um+vn}, \quad u, v = 0, 1, \dots, N-1 \quad (21)$$

$$f(m, n) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F'(u, v) W^{-um-vn}, \quad m, n = 0, 1, \dots \quad (22)$$

## 4 変換符号化

画像に何らかの変換を施し，重要な部分とそうでない部分に分けることができれば，重要な部分だけを符号化することができる．あるいは，重要な部分は正確に符号化し，そうでない部分は大雑把に符号化すれば，画像を圧縮することができる．

例えば，画像に離散フーリエ変換を施す場合を考えよう．縦横  $N \times N$  の画像を式 (22) によって離散フーリエ変換すれば， $N \times N$  個の  $u, v$  の組に対して  $F(u, v)$  が得られる．この  $F(u, v)$  は，もとの画像と同じ情報量であるので，このままでは圧縮したことにはならない．

ところで，大きい  $u, v$  に対する  $F(u, v)$  は，高い周波数成分に対応する．いま，画像がある程度滑らかに変化していて，一定以上の高い周波数成分が重要でないとするれば，それを除くことができる．周波数  $U$  以上を除けば，それだけ画像を圧縮したことになる．圧縮された画像は，以下のような離散フーリエ逆変換によって復元できる．

$$f(m, n) = \frac{1}{N} \sum_{u=0}^{U-1} \sum_{v=0}^{U-1} F'(u, v) W^{-um-vn}, \quad m, n = 0, 1, \dots, U-1 \quad (23)$$

このようにして復元した画像は，元の画像とあまり変わらないことが期待される．

### 4.1 離散コサイン変換

離散フーリエ変換は直交基底への写像であることは前節で述べた．直交基底はこれ以外にもある．直交基底を  $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_{N-1}$  とし，それらを転置して縦に並べた行列を  $A$  とする．

$$A = [\mathbf{A}_0^T, \mathbf{A}_1^T, \dots, \mathbf{A}_{N-1}^T]^T \quad (24)$$

簡単のため，1次元の信号をベクトル  $\mathbf{f}$  とし，次式のように  $\mathbf{f}$  を  $\mathbf{A}_i$  で変換した結果を  $F(i)$  とする．

$$F(i) = \mathbf{A}_i^T \mathbf{f}$$

すべての変換結果をベクトル  $\mathbf{F} = (F(0), F(1), \dots, F(N-1))^T$  で表せば，式 (24) の  $A$  を用いて

$$\mathbf{F} = A\mathbf{f}$$

この行列  $A$  を変換行列という．

直交基底による変換の一種である離散コサイン変換 (Discrete Cosine Transform, DCT) は，変換行列の成分がコサイン関数で表される．離散コサイン変換の変換行列のすべての成分は実数であるので扱いが簡単である．

まず、1次元の信号  $f$  の DCT を考えてみよう。その変換行列  $A$  の  $i$  行  $j$  列の要素  $a_{ij}$  は次式で与えられる。

$$a_{ij} = \begin{cases} \sqrt{1/N}, & i = 0 \\ \sqrt{2/N} \cos\{(2j + 1)i\pi/2N\}, & i \neq 0 \end{cases} \quad (25)$$

例えば、 $N = 4$  の場合の変換行列は

$$A = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{1/2} & \sqrt{1/2} & \sqrt{1/2} & \sqrt{1/2} \\ \cos(\pi/8) & \cos(3\pi/8) & \cos(5\pi/8) & \cos(7\pi/8) \\ \cos(2\pi/8) & \cos(6\pi/8) & \cos(10\pi/8) & \cos(14\pi/8) \\ \cos(3\pi/8) & \cos(9\pi/8) & \cos(15\pi/8) & \cos(21\pi/8) \end{bmatrix}$$

この例では次式が成り立つことがわかる。

$$AA^T = I_4, \quad A^{-1} = A^T$$

ここで  $I_4$  は、4次元単位行列である。

式(24)を参照すれば、 $A_0, A_1, \dots, A_{N-1}$  が直交基底ベクトルをなすことがわかる。なお  $N$  が4以外でも同様なことが示される(問題2.3を参照)。

この変換行列を用いた DCT を以下のように表す。

$$F_c = Af \quad (26)$$

元の信号  $f$  は次式の離散コサイン逆変換 (Inverse Discrete Cosine Transform, IDCT) で復元される。

$$f = A^T F_c \quad (27)$$

以上は、DCT と IDCT をベクトルで表したが、ベクトルを用いなくて、つぎのように表すこともできる。

$$F_c(i) = \sqrt{2/N} C(i) \sum_{j=0}^{N-1} f(j) \cos\{(2j + 1)i\pi/(2N)\} \quad (28)$$

$$f(i) = \sqrt{2/N} \sum_{j=0}^{N-1} C(j) F_c(j) \cos\{(2i + 1)j\pi/(2N)\} \quad (29)$$

ただし、

$$C(i) = \begin{cases} \sqrt{1/2}, & i = 0 \\ 1, & i \neq 0 \end{cases} \quad (30)$$

つぎに、2次元の信号(画像)の DCT を考えよう。普通は、画像全体を変換するのではなく、画像を矩形のブロックに分割し、それぞれのブロックに対して2次元の DCT を施す。 $N \times N$  の正方形のブロックに対する2次元の信号  $f(j, k)$  の DCT と逆変換 (IDCT) は以下の式で与えられる。

$$F(u, v)_c = (2/N) C(u) C(v) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \cdot \cos[(2j + 1)u\pi/2N] \cos[(2k + 1)v\pi/2N] \quad (31)$$

$$f(j, k) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} (2/N)C(u)C(v)F_c(u, v) \cdot \cos[(2j+1)u\pi/2N] \cdot \cos[(2k+1)v\pi/2N] \quad (32)$$

この意味を考えてみよう。いま

$$(2/N)C(u)C(v) \cos[(2j+1)u\pi/2N] \cos[(2k+1)v\pi/2N]$$

を  $j$  行  $k$  列の要素とする 2 次元行列を  $B_{uv}$  としよう。式 (32) は、画像  $f$  が  $B_{uv}$  に重み  $F_c(u, v)$  をつけて、すべての  $u, v$  について加えたものであることを意味している。この  $\{B_{uv} | u = 0, 1, \dots, N-1, v = 0, 1, \dots, N-1\}$  は直交基底をなしている。行列の集合が直交基底をなしているとは、その集合の要素  $B_{uv}$  と  $B_{st}$  の内積 (2 つの行列の対応する要素の積の和) が、 $u = s, v = t$  の場合を除いて 0 となることである。すなわち、2 次元の DCT は、直交基底の行列による展開である。

## 4.2 JPEG

JPEG(ジェイペグ) は、静止画圧縮のために国際標準で、Joint Photographic Experts Group とよばれる専門家グループが作った。ここでは、その代表的な方法を述べる。

画像を前節の DCT のように、 $8 \times 8$  の画素からなるブロックに分割して、各ブロックに DCT を施す。明度の平均 (DCT の直流成分である  $F_c(0, 0)$ ) をなるべく小さくしたほうが、圧縮効率が高くなるので、明度の中央値が 0 となるように明度を変換してから、各ブロックで DCT を行なう。

DCT の結果の  $F_c(u, v)$  は、普通  $u, v$  の小さいほど大きい値となる。また、 $u, v$  の大きいところは高周波成分に対応するので、あまり重要でなく、あまり精度を必要としない (人の目には、画像の高周波数の部分では精度の劣化に対して鈍感である)。そこで、高周波成分を量子化するとき刻み幅を大きくすることができる。各  $u, v$  に対して、量子化するときの刻み幅  $Q_{uv}$  を決めておき、量子化テーブルとして蓄えておく。この量子化テーブルを用いて、 $F_c(u, v)$  は次式で量子化される

$$r_{uv} = \text{round}(F_c(u, v)/Q_{uv})$$

ここで round は最も近い整数値にすることを意味する。この結果ハフマン符号化する。

JPEG の圧縮法の圧縮の割合や復元した画像の精度は、量子化テーブルによって決められる。代表的な  $Q_{uv}$  は、10 から 120 程度の範囲である。この圧縮は不可逆であるが、普通の濃淡画像は、1/10 に圧縮してもあまり差がわからない。

カラー画像に対しては、明るさ (輝度成分) と色 (色差成分) に分けて (6 章の色彩画像処理を参照)、それぞれをここで述べた方法で圧縮する。

## 5 動画画像の圧縮

動画画像はデータ量が膨大であるので、圧縮率を上げなければならない。動画画像の連続するフレーム (画面) は、類似している場合が多いので、フレーム間予測が有効である。すなわち、静止画の圧縮に用いたフレーム内予測の手法とフレーム間予測を組み合わせて圧縮の効果を上げるのである。

ここでは、最初に動画画像特有なフレーム間予測の方法を説明し、つぎに動画画像圧縮の標準である MPEG について述べる。

## 5.1 フレーム間予測

最も簡単な方法は、予測符号化で述べた方法で、参照画素として一つ前の画像を用いることである。

簡単のため、画像を1次元信号で表し、現在のフレーム  $t$  の画素値の予測  $\hat{f}_t(i)$  を一つ前のフレームの画像の線形予測で求めるとする。参照画素は対応する  $i$  番目の画素とその近傍の画素 ( $k$  画素離れた画素まで) とすれば、式 (1) と同様に、

$$\hat{f}_t(i) = c_0 f_{t-1}(i) + c_1 f_{t-1}(i-1) + \cdots + c_k f_{t-1}(i-k) \\ + c_{k+1} f_{t-1}(i+1) + \cdots + c_{2k+1} f_{t-1}(i-k)$$

この方法は、画像の中で動いている部分が少なければ有効である。しかし、カメラがゆっくり動く場合は、画像のほとんどの部分が変化しているので、あまり有効でない。そこで、画像のそれぞれの部分がどのように動いたかを検出し、その動きを考慮して予測する方法 (動き補償予測) が考えられた。

動きベクトルを検出するための代表的な方法は、画像を一定の大きさの小領域 (ブロック) に分割し、それぞれのブロックの動きベクトルを求める。これはブロックマッチング法とよばれる。各ブロックが前のフレームのどこに対応するかを探し、対応するブロックの位置の差を動きベクトルとする。

ブロックマッチング法では、以下の計算を行なう。いま、第  $t$  フレームの  $N \times N$  の大きさのブロックの動きベクトルを求めるものとする。ブロックの左上端の画素の位置を  $(i_0, j_0)$  とする。このブロックの画像と直前のフレームの対応する位置から  $(p, q)$  だけずらした位置のブロックの画像との差を評価する。評価は、以下の式に示すように、画素値の差の絶対値の和 (sum of absolute difference, 略して SAD) が用いられる。

$$S(p, q) = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} |f_t(i_0 + i, j_0 + j) - f_{t-1}(i_0 + i + p, j_0 + j + q)| \quad (33)$$

対応候補のブロックの位置  $(p, q)$  を変えて  $S(p, q)$  を計算し、最小となる  $(p, q)$  を動きベクトル  $(p_m, q_m)$  とする。探索範囲  $R_s$  は、 $-l \leq p, q \leq h$  の正方形であることが多い ( $l, h$  は正数)。以上を式で示せば

$$(p_m, q_m) = \arg \min_{(p, q) \in R_s} S(p, q) \quad (34)$$

ブロックマッチングによって求めなければならない動きベクトルの数はブロックサイズ  $N$  が大きいほど少なくなるので、なるべく  $N$  を大きくしたい。しかし、 $N$  が大きくなると、ブロックの中の動きが一様でなくなるため、予測誤差が大きくなる。普通は、 $8 \times 8$  から  $16 \times 16$  程度である。また、探索範囲は、 $32 \times 32$  程度である。

## 5.2 MPEG1

前に述べた静止画の圧縮の標準である JPEG に対して、動画の圧縮の標準もいくつか定められている。その中で、国際標準化機構 (ISO) の中の動画符号化グループ (Motion Picture Experts Group, MPEG) が作った標準が広く用いられている。ここでは、最初に作られた MPEG1 とよばれる標準を説明する。MPEG1 では、動画の符号化の枠組が決められてるが、その枠組の範囲内では、多様な符号化が可能である。どのような符号化を行なっているかを示すパラメータを指定す



ることになっていて、そのパラメータをどのような形式で書いておくかも決められている。ここでは、簡単のために典型的な場合を説明する。

1 フレームの画像は、240本の走査線で構成され、1本の走査線は252画素で構成される。動画は、このフレームの連続した時系列によって表される。普通のテレビ画像は走査線が480本で、1本の走査線は640画素で構成されるので、その4分の1弱の画素数である。MPEG1は、比較的low resolutionの画像を対象とするので、伝送速度は約1.5Mbps (メガビット/秒) までである。

ビデオテープなどに蓄積された動画画像を見る場合は、任意の時刻の時系列を再生したい。もし、動画画像がフレーム間予測符号化だけで圧縮されていれば、最初の画像から順番に復元しなければならない。それでは困るので、適当な間隔でフレーム内予測符号化画像を置いておく。この画像をIピクチャ(Intra-Picture, フレーム内予測符号化画像の略)とよぶ。なお、ピクチャはフレームと同じ意味である。復元の場合は、このIピクチャから始めれば、最初にそのフレームだけで復元できる。それ以後は、フレーム間予測符号化を適用できる。

Iピクチャの符号化はJPEGと同様に、 $8 \times 8$ 画素からなるブロック毎にDCTを用いて圧縮している。

Iピクチャの間に、フレーム間予測を行なって復元するフレームが入るが、そのフレームも2種類ある。まず、普通にIピクチャから順方向に予測符号化を行なう画像で、Pピクチャ(Predictive-Picture, フレーム間順方向予測符号化画像の略)とよばれる。このPピクチャはIピクチャからつぎのIピクチャまで、 $M$ フレーム毎に現れる(普通は $M = 3$ )。

Pピクチャのフレーム間予測符号化は、MB(Macro Block, マクロブロック)単位に行なわれる。MBは、 $16 \times 16$ 画素からなり、 $8 \times 8$ 画素のブロック4個で構成される。動きベクトルの探索範囲は、 $M$ によって異なる。MPEG1では、毎秒30フレームの動画画像の $L$ フレーム離れたフレーム間予測の場合の水平方向および垂直方向の探索範囲は、 $L = 1$ のとき $[-16, 15]$ 、 $L = 2$ のとき $[-32, 31]$ 、 $L = 4$ のとき $[-64, 63]$ である。

普通は、 $M = 3$ であるから $L$ も3となるべきであるが、圧縮効率を高めるために $L = 3$ が用意されていないので、それに近い $L = 4$ が採用され、探索範囲は $[-64, 63]$ である。探索の刻み幅は、普通 $M = 1$ のときは1画素、 $M = 2$ のときは2画素で、探索の数は一定となる。

最後に、IピクチャあるいはPピクチャの間に挿入されるフレームで、前後のIあるいはPピクチャの双方向から予測される画像で、Bピクチャ(Bidirectionally predictive-Picture, 双方向予測符号化画像の略)とよばれる。Bピクチャは、 $M - 1$ フレームが連続する(普通は2フレーム)。

Bピクチャの動きベクトルを求める場合、1方向からだけの予測と双方向の予測がある。1方向の予測は、Pピクチャと同様に行なう。普通、BピクチャはIピクチャあるいはPピクチャから1か2離れているので、前述の $L = 1$ か $L = 2$ に対応する探索範囲となる。

双方向からの予測は、まずそれぞれの方向から1方向の場合と同様にして動きベクトルを求めて、画素値を予測する。各画素に対して2つの予測値が得られるので、その平均を丸めた値を最終的な予測値とする。

Bピクチャは他の画像の予測には用いられない。PピクチャはBピクチャと一部のPピクチャの予測に用いられ、IピクチャはBピクチャとPピクチャの予測に用いられる。したがって、Bピクチャの精度は少し低くてもかまわないので、他より量子化の粗さを倍にしていることが多い。

符号化された動画画像は、IピクチャからIピクチャまでを周期とする画像系列で構成されている。1周期の長さは普通15フレーム(0.5秒)である。この1周期の画像系列をGOP(Group of Pictures, 画像グループ)とよぶ。Iピクチャからは、その直前の2つのBピクチャを予測することができるので、1つのGOPは図に示すように、Iピクチャのフレームより2つ前の画像から始まる15フレー

ムの動画像に対応している。

1つのGOPの符号化は以下の手順で行なう。

1. 原動画像の最初の2フレーム(B0, B1)を蓄える。
2. 3番目のフレーム(I2)をフレーム内予測符号化する。
3. I2からB0, B1をフレーム間予測を行なって符号化する。
4. I2からP5をフレーム間予測を行なって符号化する。
5. I2とP5からB3, B4を双方向のフレーム間予測を行なって符号化する。
6. P5からP8をフレーム間予測を行なって符号化する。
7. P5とP8からB3, B4を双方向のフレーム間予測を行なって符号化する。
8. 以下6.および7.と同様な処理を繰り返す。

この手順では、1.で最初の2フレームを蓄える以外は画像を蓄える処理が書かれていないが、実際には処理を行なっている画像と、原画像が必ずしも一致しないので、原画像は一度蓄えてから、必要に応じて使うようにしなければならない。蓄えなければならない画像は2フレーム分である。

復元の場合は、図(講義で示す)に示すように、符号化の場合と同様な順序で行なう。この場合も再生画像の順番に復元されないので、一度蓄えてから順序を整えて表示する。この場合も、2フレーム分蓄えなくてはならない。同図からわかるように、動画像を中継で送る場合は、符号化と復号化で合計3フレーム遅れる。また、画像の符号化と復号化に要する時間による遅れもある。しかし、MPEG1は動画像を圧縮して蓄えておき、必要に応じてそれを再生することを想定しているので、このような遅れはほとんど問題ない。

MPEG1は、GOPを単位としてランダムにアクセスできる特徴がある。早送り再生する場合、Iピクチャ毎に表示することや、Pピクチャ毎に表示することは簡単である。また、逆転再生を行なう場合は、GOP単位で再生することができるから、1つのGOP(15画像)を逆順に蓄えてから表示すればよい。

MPEG1の圧縮データのフォーマットが決められている。各GOPの前にはシーケンスヘッドがあり、画像の大きさ、表示周期、DCT圧縮に用いられる量子化マトリクスなどの情報が書かれている。同様にGOPの前、ピクチャの前などにも必要な情報が書かれている。その他カラー画像に関する圧縮についても決められているが、ここでは省略する(以上の詳細は参考文献「最新MPEG教科書」を参照)。

### 5.3 MPEG1以降

MPEG1は、伝送速度が約1.5Mbpsまでを対象としているが、より高品質の画像の圧縮が必要になった。現在のテレビ画像の品質を実現するために、MPEG2が提案された。これは、より高解像度のテレビ(HDTV)にも適用できる。事実、米国とヨーロッパの次世代テレビにMPEG2が採用されている。圧縮の原理はMPEG1と同じであるが、テレビの走査法であるインタレース画像を扱えること、種々の画像の品質を扱えることなどである。現在は、さらに空間分解能や時間分解能の高い画像を対象とするMPEG4も提案されている。高品質の画像のデータ量は膨大であるので、圧縮効率を上げるために正確な動きベクトルを高速に求めることが大切になる。画像の圧縮は、しだいに複雑な画像処理を必要とするようになっていく。