信号画像処理特論 I

担当： 工藤博幸

講義を始めるにあたって

1. 『画像・音声の符号化（圧縮）』を取り扱う.
画像符号化とは，予測符号化，変換符号化，その他の符号化法

2. 教科書は使用しない. スライド，板書，プリントを中心とする.
（講義の内容と合致した参考書を紹介）

3. 毎回後半に重要な項目を文章に要約したりクイズに答えたりする『小テスト』を行い出席点とする.

4. 成績評価は『レポートまたは筆記試験（70%）』+『出席点（30%）』により行う.

5. 勉強の目安としては，標準的手法である音声符号化のDPCMや画像符号化のJPEGをきちっと理解する.

スライドコピーの入手先

以下のURLから各自ダウンロードして2回目以降の講義に持参してください.

http://www.cs.tsukuba.ac.jp/~kudo/japanese.html

（認証のユーザIDとパスワードは授業中に教えます）

講義の内容と合った参考書

（1）吉田俊之，鈴木輝彦，広沢敏彦，『画像情報符号化』，コロナ社，2008年

（2）原島博，『画像情報圧縮』，オーム社，1991年

（3）酒井善則，吉田俊之，『映像情報符号化』，オーム社，2001年

（4）J.W.Woods, Multidimensional signal, image, and video processing and coding, Academic Press, 2006

（5）R.J.Clarke, Digital compression of still images and video, Academic Press, 1995

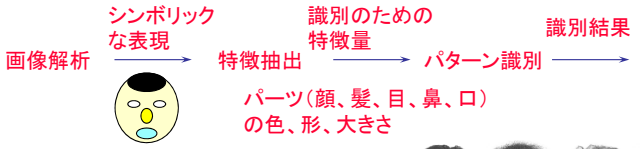
（6）J.S.Lim, Two-dimensional signal and image processing, Prentice Hall, 1990

（前おき）
画像処理にはどんな分野があるか

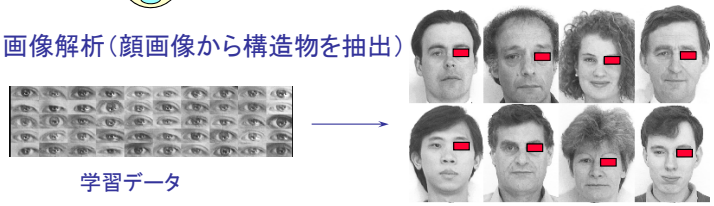
画像の認識・理解

画像に写っているものをコンピュータに認識させる技術
（文字認識、ロボットの環境認識、個人照合、医用画像認識）

画像認識の手順

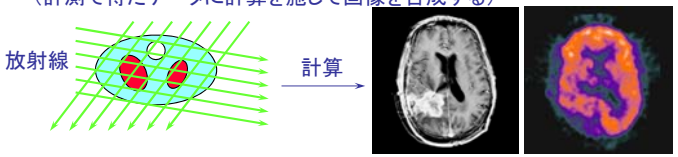


画像解析（顔画像から構造物を抽出）



医用イメージングと計算機支援医療

コンピュータの利用による医療の変化
・計算イメージング（CT、MRI、超音波）の出現（1973年）
（計測で得たデータに計算を施して画像を合成する）

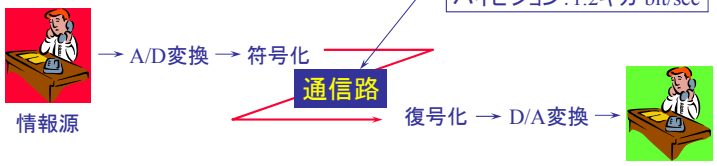


・コンピュータによる医用メディア処理技術の発展（1990年代）

	診断に利用されるメディア	診断・治療の方法
CTの発明以前	レントゲン写真、心電図、脳波	医師
CTの発明以降	上記＋X線CT、MRI（磁気共鳴イメージング）、エミッションCT、超音波	
将来		計算機支援 完全自動化

メディア符号化

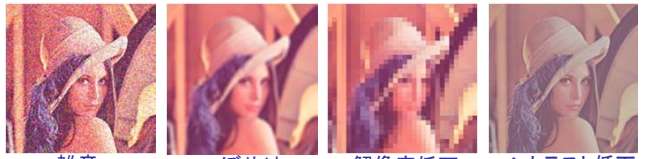
音声・画像通信のモデル



メディア符号化の目的
（1）通信路の効率的使用のためデータを圧縮する
（2）信頼性向上のため通信路の誤りが訂正が可能な形式にする

劣化画像の品質改善

画像の劣化要因を取り除き人間に見やすい画像に改善

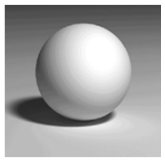


劣化のモデル $\vec{g} = A\vec{f} + \vec{n}$ に基づき $\vec{g}$ から $\vec{f}$ を推定

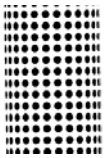
（$\vec{f}$：原画像、$\vec{g}$：劣化画像、A：劣化を表す演算子、$\vec{n}$：雑音）

コンピュータ・ビジョン（CV）

2次元画像から3次元空間（奥行き情報）を知覚する機能の実現



陰影からの形状復元



模様からの形状復元

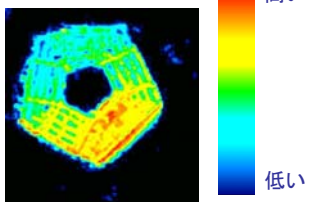
左眼画像　右眼画像

ステレオ視

飛行機から撮影

計算

建物の高さ（飛行機からの距離）



高い

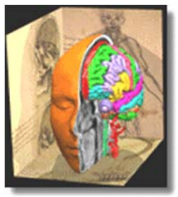
低い

コンピュータ・グラフィックス（CG）

画像の生成過程をコンピュータでシミュレーションして画像合成

コンピュータ
（1）対象物をデータとして表現
（2）光の伝搬のシミュレーション

代表的な画像の作り方
・レイトレーシング（対象物が球、直方体などの幾何学的形状の組み合わせ）
・ボリュームレンダリング（対象物がボクセルデータとして表現）



第1章：画像の符号化

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.3 & 0.59 & 0.11 \\ 0.6 & -0.28 & -0.32 \\ 0.21 & -0.52 & -0.31 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

1．画像符号化とは

1．1　画像符号化の目的

画像をできるだけ少ないビット数で表現する

応用分野
（1）画像伝送（テレビ信号の帯域圧縮・インターネット）
・現在の放送用テレビの帯域幅
輝度信号（Y）4.2MHz 色差信号（I，Q）1.5MHz,0.5MHz
-> 2倍の周波数でサンプリングして8bitで量子化すると
情報量（4.2+1.5+0.5）×2×8 ＝ 99.2(Mbit/s)
・ハイビジョン 1.2(Gbit/s)
・音声 64(Kbit/s)のディジタル伝送実用化
（2）画像の記録
ハードディスク，CD-ROM

情報圧縮が必要

画像符号化器の構成

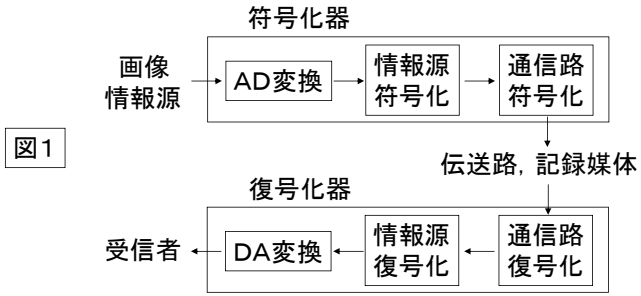


図1

情報源符号化（Source coding）-> データの圧縮を目的とした符号化
通信路符号化（Channel coding）-> 伝送路，記録媒体の信頼性向上を目的とした符号化
（誤り訂正符号など）

1．2　画像の統計的性質

画像符号化 -> 画像の冗長性を削減することで圧縮を行う

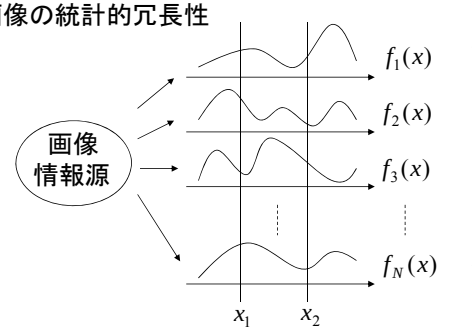
画像情報源 → 冗長度削減 → 伝送路 → 冗長度付加 → 受信者

画像に含まれる冗長性

1．統計的冗長度（空間的冗長度，時間的冗長度）　図2
   -> 画像を確率信号と考えたときの冗長度
2．構造的冗長度 -> 画像を領域の集合とみたときの構造的冗長度
3．知識的冗長度 -> 送受信端で共有している知識に関する冗長度
4．エントロピー的冗長度 -> 符号の出現確率の偏り
5．視覚的冗長度 -> 人間の歪知覚特性に起因する冗長度

ほとんどの画像符号化法は統計的冗長度の削減を利用している

画像の統計的冗長性



情報源の性質を特徴づける量 -> 共分散関数

$$\phi(x_1,x_2)=E[f(x_1)f(x_2)]\ (=\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}f_i(x_1)f_i(x_2))$$　式（1）

期待値

・定常性 -> $\phi(x_1,x_2)$ が x_1-x_2 だけの関数

time lag

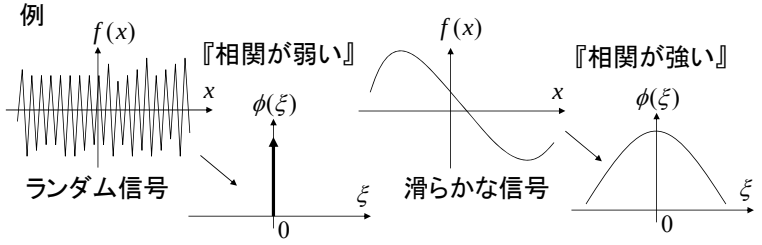
$$\phi(\xi)=E[f(x)f(x+\xi)]$$　式（2）

・エルゴート性 -> 期待値を空間平均で置きかえてよい

$$\phi(\xi)=\lim_{L\to\infty}\frac{1}{2L}\int_{-L}^{L}f(x)f(x+\xi)dx$$　式（3）

$\phi(\xi)$を自己相関関数と呼ぶ
-> ξ だけ離れた位置にある信号の値の関係

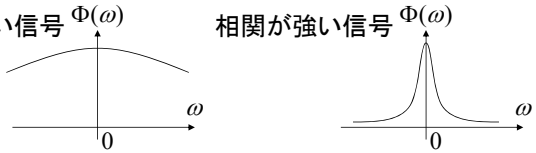
例



パワースペクトル密度 -> 自己相関関数のフーリエ変換

$$\Phi(\omega)=\int_{-\infty}^{\infty}\phi(\xi)e^{-j\omega\xi}d\xi$$　式（4）

相関が弱い信号 $\Phi(\omega)$　　相関が強い信号 $\Phi(\omega)$



画像信号の統計的冗長性 -> 自己相関関数やパワースペクトル密度で表現
（相関が強い＝冗長，相関が弱い＝冗長でない）

2次元への拡張

$$\phi(\xi,\eta)=\lim_{L_x\to\infty}\lim_{L_y\to\infty}\frac{1}{4L_xL_y}\int_{-L_y}^{L_y}\int_{-L_x}^{L_x}f(x,y)f(x+\xi,y+\eta)dxdy$$

$$\Phi(u,v)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi(\xi,\eta)e^{-j(u\xi+v\eta)}d\xi d\eta$$　式（5）

実際の画像で計算したもの -> 図3

画像はきわめて冗長な（相関が強い）信号である

$$\rho_H = \frac{\phi(1,0)}{\phi(0,0)} > 0.9 \qquad \rho_V = \frac{\phi(0,1)}{\phi(0,0)} > 0.9$$

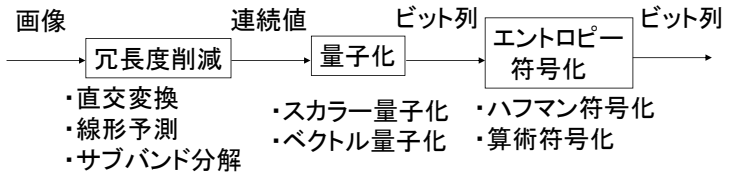
式（6）

水平方向相関係数　垂直方向相関係数

『冗長な』＝『相関が強い』＝『滑らかな』
＝『ある画素から隣りの画素が予測できる』

1．3　画像符号化の考え方

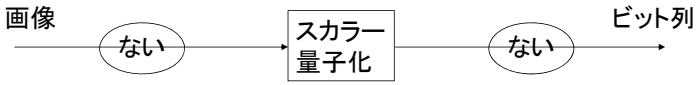
画像符号化法の一般的構成



冗長度削減 -> 相関の強い信号を相関の弱い信号に変換
量子化 -> 連続値をとる信号を離散値をとる信号（ビット列）に変換
エントロピー符号化 -> 0と1の発生頻度の偏りを利用して情報削減

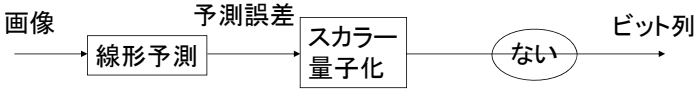
3要素の組み合わせにより様々な符号化法

例　（1）PCM（パルス符号変調）

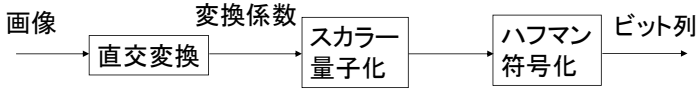


・アナログをディジタルに変換するのが目的
・冗長度削減モジュールない -> 情報圧縮は期待できない

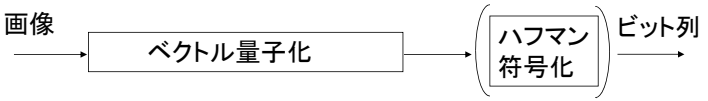
（2）DPCM（差分パルス符号変調）　音声64(Kbit/s)符号化



（3）変換符号化　JPEG　　1970～1980年代

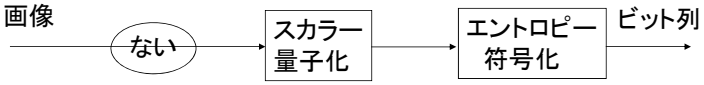


（4）ベクトル量子化　　1980年以降



ベクトル量子化は冗長度削減＋量子化の効果がある

（5）GIF（Graphics Interchange Format）

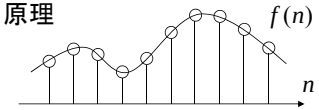


量子化誤差以外の歪みがないが，冗長度削減モジュールがないため情報圧縮はあまり期待できない

符号化法は冗長度削減の方法で分類
- 予測符号化
- 変換符号化
- ベクトル量子化
- サブバンド符号化

第2章：予測符号化（Predictive coding）

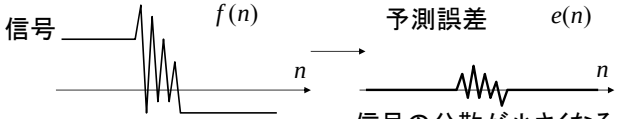
原理



サンプリングされた信号

これまで送った信号の値 $f(0), f(1), \cdots, f(n-1)$ から次の信号の値 $f(n)$ を予測し予測誤差 $e(n) = f(n) - \hat{f}(n)$ だけを伝送する
-> 冗長度の削減

予測値

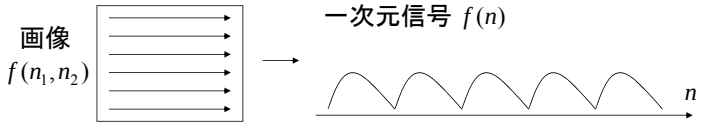


信号の分散が小さくなる
-> 少ないビット数で量子化

特徴
1．変換符号化と比較して構成が簡単
2．性能は変換符号化と比較して悪い

2．1　デルタ変調（DM）

画像 $f(n_1, n_2)$ をラスタ走査で1次元信号 $f(n)$ に変換し隣りの信号の値との差を1ビットで量子化する



符号化器と復号化器の構成

次ページ

$$e(n) = f(n) - \hat{f}(n-1) \quad \text{予測誤差}$$

$$\hat{e}(n) = \begin{cases} \Delta/2 & (e(n) > 0 \text{ のとき}) \\ -\Delta/2 & (\text{その他}) \end{cases}$$

量子化 -> 1つ前の値から増えたか減ったかを伝送

式（7）

$$\hat{f}(n) = \hat{f}(n-1) + \hat{e}(n) \quad \text{復号}$$

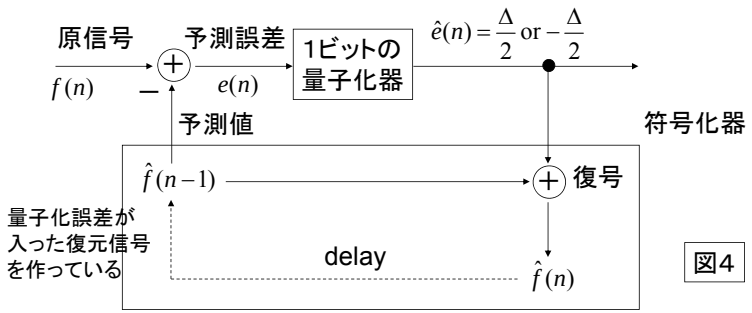
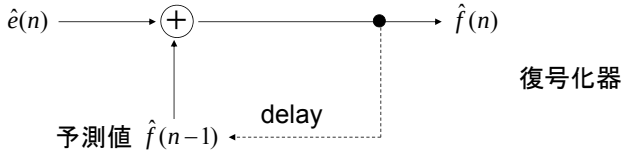


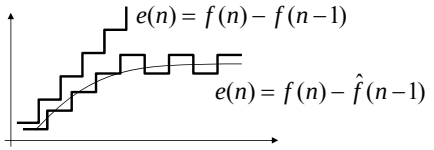
図4



注意

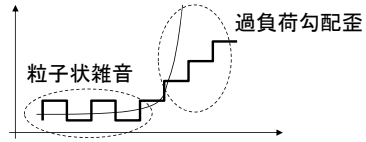
1．予測誤差の求め方　$e(n) = f(n) - \hat{f}(n-1)$　復元信号

$e(n) = f(n) - f(n-1)$ とすると量子化誤差が蓄積する
（復号化器で $\hat{f}(n) = \hat{f}(n-1) + \hat{e}(n)$ とするため）



2．ステップサイズ Δ を適切に選ぶ必要

- Δ が大きすぎる -> 粒子状雑音（平坦部での雑音）
- Δ が小さすぎる -> 過負荷勾配歪（信号の変化に追従できない）

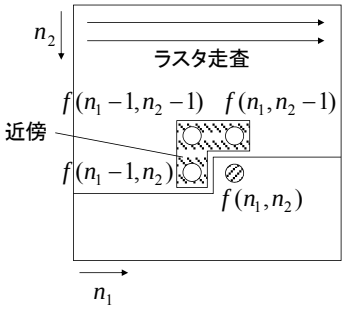


実験例
-> 図5

2．2　差分パルス符号変調（DPCM）

デルタ変調の一般化

1．次の画像の値 $f(n_1,n_2)$ の予測を複数の画素値で行う



予測誤差

$$e(n_1,n_2)=f(n_1,n_2)-f'(n_1,n_2)$$

（$f'(n_1,n_2)$：予測値）

$$f'(n_1,n_2)=F[f(n_1-1,n_2),f(n_1,n_2-1),f(n_1-1,n_2-1)]$$

（F：予測演算子）

式（8）

2．予測誤差 $e(n_1,n_2)$ を複数ビットで量子化

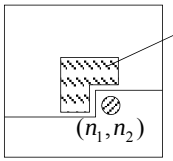
$$\begin{cases} f'(n_1,n_2)=F[\hat{f}(n_1-1,n_2),\hat{f}(n_1,n_2-1),\hat{f}(n_1-1,n_2-1)] & \text{予測} \\ e(n_1,n_2)=f(n_1,n_2)-f'(n_1,n_2) & \text{予測誤差} \\ \hat{e}(n_1,n_2)=Q[e(n_1,n_2)] & \text{量子化} \\ \hat{f}(n_1,n_2)=f'(n_1,n_2)+\hat{e}(n_1,n_2) & \text{復号化} \end{cases}$$

式（9）

予測の方法 -> 線形予測

$$f'(n_1,n_2)=\sum_{(k_1,k_2)\in R_a}\sum a(k_1,k_2)\hat{f}(n_1-k_1,n_2-k_2)$$

式（10）



$R_a=\{(1,0),(0,1),(1,1)\}$ 近傍

$a(k_1,k_2)$ 予測係数

予測係数は予測誤差の2乗期待値が最小となるように求める

$$E[e^2(n_1,n_2)]=E[(f(n_1,n_2)-\sum_{(k_1,k_2)\in R_a}\sum a(k_1,k_2)\hat{f}(n_1-k_1,n_2-k_2))^2] \text{ -> min}$$

式（11）

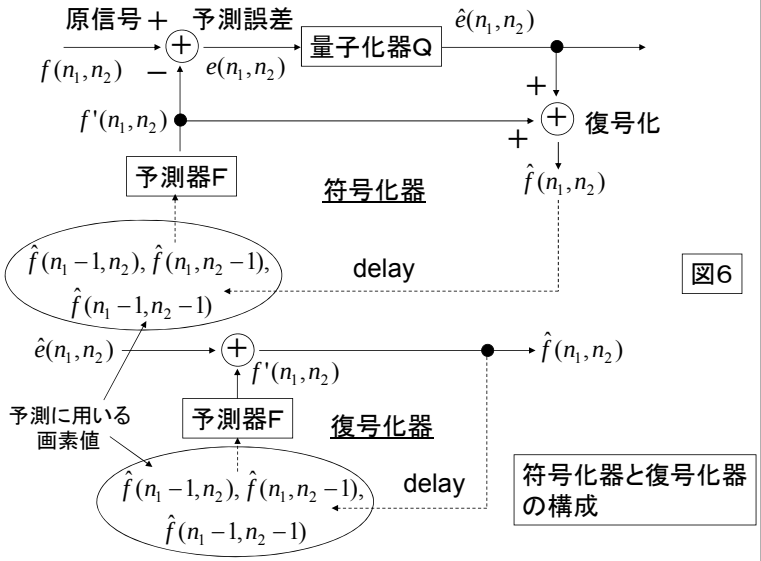


図6

符号化器と復号化器の構成

$\hat{f}(n_1,n_2)\approx f(n_1,n_2)$ で近似すると

$$E[(f(n_1,n_2)-\sum_{(k_1,k_2)\in R_a}\sum a(k_1,k_2)f(n_1-k_1,n_2-k_2))^2] \text{ -> min}$$

式（12）

$a(k_1,k_2)$ について微分して零とおく

自己相関関数 $\phi(\xi,\eta)=\sum_{(k_1,k_2)\in R_a}\sum a(k_1,k_2)\phi(\xi-k_1,\eta-k_2)$

$$\begin{bmatrix}\phi(0,0) & \phi(1,1) & \phi(0,1)\\ \phi(1,1) & \phi(0,0) & \phi(1,0)\\ \phi(0,1) & \phi(1,0) & \phi(0,0)\end{bmatrix}\begin{bmatrix}a(1,0)\\ a(0,1)\\ a(1,1)\end{bmatrix}=\begin{bmatrix}\phi(1,0)\\ \phi(0,1)\\ \phi(1,1)\end{bmatrix}$$

Yule-Walkerの方程式

ブロックToeplitz行列

式（13）

実験例 -> 図7

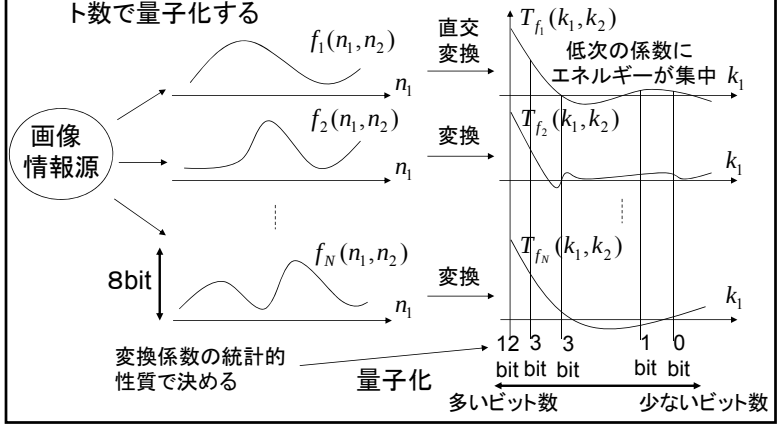
ADPCM（適応DPCM）

予測係数を画像全体で同じにするのではなく，場所に依存させることで性能改善

第3章：変換符号化（Transform coding）

原理

相関の強い信号を直交変換すると低次の係数にエネルギーが集中するので，各変換係数をその統計的性質に応じたビット数で量子化する



ビット割り当て

復元画像の歪（2乗誤差）Dを認めたとき，最も低いビット率 R とそれを実現する各係数へのビット配分 $N_B(k_1,k_2)$

仮定

$T_f(k_1,k_2)$ は独立なガウス分布（変換により相関除去）

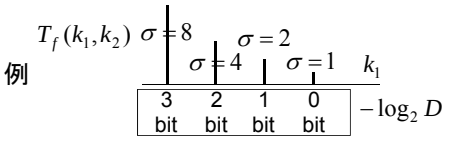
$$P_{k_1,k_2}(x)=\frac{1}{\sqrt{2\pi}\sigma(k_1,k_2)}e^{-\frac{x^2}{2\sigma^2(k_1,k_2)}}$$

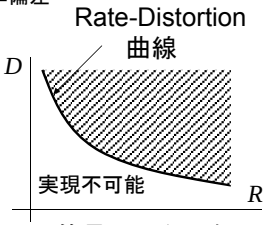
標準偏差

解 -> Shannon

$$\begin{cases} N_B(k_1,k_2)=\frac{1}{2}\log_2\sigma^2(k_1,k_2)-\log_2 D \\ R=\sum_{k_1}\sum_{k_2}N_B(k_1,k_2) \quad \text{ビット率} \end{cases}$$

例



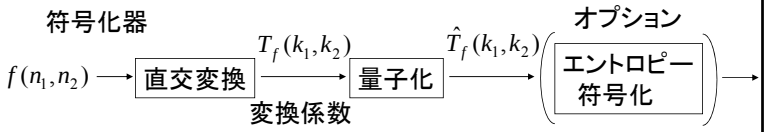


特徴

1. 予測符号化より性能が良い
2. 予測符号化より計算量が多い -> 直交変換のため
3. 現在主流となっている

3．1　変換符号化の構成

符号化器



復号化器

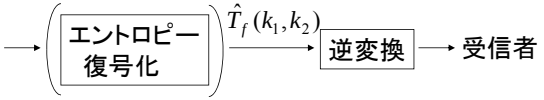
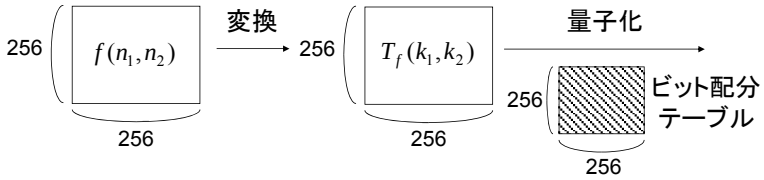


図8

全フレーム符号化とブロック符号化

（1）全フレーム符号化

原画像全体で直交変換を行う



各変換係数 $T_f(k_1,k_2)$ へのビット割り当ては，様々な画像の変換係数の統計的性質から求めておく -> ビット配分テーブル

特徴

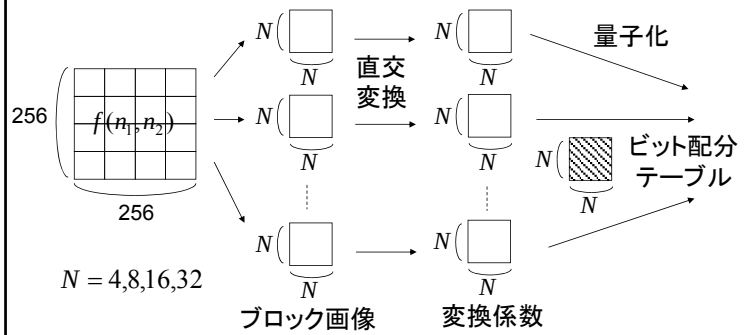
1. 原理的にはブロック符号化より高性能 -> 冗長性除去効果高い
2. 計算量が多い
3. ブロック歪みが生じない

医用画像で用いられている

※ブロック歪みの例

（2）ブロック符号化

原画像をブロック分割しブロック単位で直交変換



ビット配分テーブルは複数ブロックの変換係数の統計的性質から決める

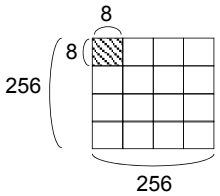
特徴

1．計算量が少なく実時間処理向き

2．ブロック歪みが生じる

※ 直交変換の計算量の違い

FFTなどの高速計算法を用いた場合



一回の直交変換の計算量 $N^2 \log_2 N +$ 低次

全体 $256^2 \log_2 256 = 524{,}288$

ブロック $32^2 \times 8^2 \times \log_2 8 = 196{,}608$

3．2 様々な直交変換

変換符号化に適した直交変換

1．エネルギー集中率（冗長性除去効果）が高い

2．高速な計算法が存在する

水平方向と垂直方向に分離可能な直交変換の一般形

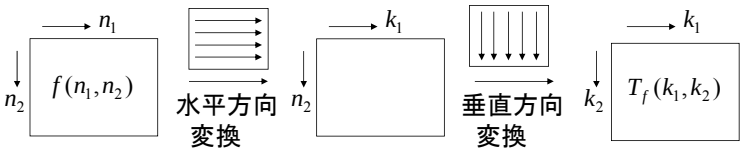
$$\begin{cases} \text{順} & T_f(k_1,k_2) = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} f(n_1,n_2)\, a_R(n_1;k_1)\, a_C(n_2;k_2) \\ \text{逆} & f(n_1,n_2) = \sum_{k_1=0}^{N_1-1}\sum_{k_2=0}^{N_2-1} T_f(k_1,k_2)\, b_R(n_1;k_1)\, b_C(n_2;k_2) \end{cases}$$

基底

式（14）

（一般には基底は$a(n_1;n_2;k_1;k_2)$以上簡単にならない）

1次元変換のくり返しで計算可能



1次元変換のみを考えて十分

（1）アダマール変換

変換の行列表現

$$N\left(\begin{bmatrix} \\ T_f(k) \\ \\ \end{bmatrix}\right) = N\left(\begin{bmatrix} \overbrace{\quad\quad}^{N} \\ H_N \\ \\ \end{bmatrix}\begin{bmatrix} \\ f(n) \\ \\ \end{bmatrix}\right) N \qquad N = 2^l$$

図9

$$H_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad H_{2n} = \frac{1}{\sqrt{2}}\begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix} \quad (n = 2,4,8,\cdots)$$ 式（15）

特徴

1．加減算のみで実現

2．エネルギー集中率はあまり良くない

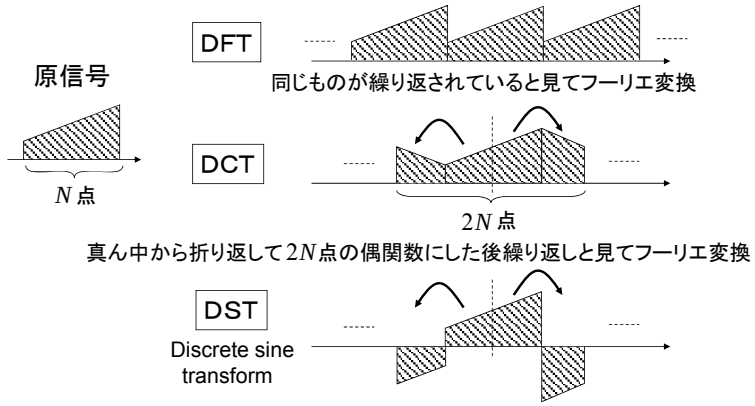
（2）離散フーリエ変換（DFT）

$$\frac{2\pi k}{N} \to \omega$$

複素数 → $$T_f(k) = \sum_{n=0}^{N-1} f(n) e^{-j\frac{2\pi kn}{N}}$$ 式（16）

※ 正弦波の基底関数を持つ直交変換が多数存在してエネルギー集中率が違う理由

有限長の信号は外側をどうみるかに様々な解釈がある



（3）離散コサイン変換（DCT）

実数

順 $$T_f(k) = \alpha(k)\sum_{n=0}^{N-1} f(n)\cos\left(\frac{\pi(2n+1)k}{2N}\right)$$ 式（17）

$$\alpha(0) = \sqrt{\frac{1}{N}},\ \alpha(k) = \sqrt{\frac{2}{N}}\ (1 \le k \le N-1)$$ 図10

逆 $$f(n) = \sum_{k=0}^{N-1} \alpha(k) T_f(k)\cos\left(\frac{\pi(2n+1)k}{2N}\right)$$ 式（18）

エネルギー集中率が高い -> 多くの変換符号化で採用

くり返し波形の不連続性が一番小さいDCTが高周波成分が最小
-> エネルギー集中率が最も良い

エネルギー集中率　DCT ＞ DFT ＞ DST

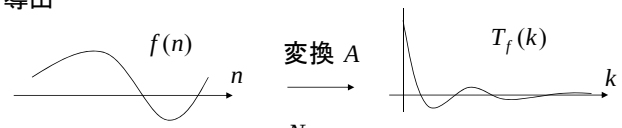
このことが書いてある歴史的な論文

A.K.Jain,” A sinusoidal family of unitary transforms,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.1, pp.356-365, 1979

（4）KL（Karhunen-Loève）変換

信号の相関を完全に取り除く変換

導出



$$N\left(\left[\begin{array}{c} \\ T_f(k) \\ \\ \end{array}\right]\right)_{=\vec{t}} = N\left(\overbrace{\left[\begin{array}{ccc} & & \\ & A & \\ & & \end{array}\right]}^{N}\left[\begin{array}{c} \\ f(n) \\ \\ \end{array}\right]\right)_{=\vec{f}} N \qquad \vec{t} = A\vec{f}$$

信号 $\vec{f}$ の共分散行列　$\phi_{\vec{f}} = E[\vec{f}\vec{f}^{\,T}] = N\left(\overbrace{\left[\begin{array}{c} \\ E[f_i f_j] \\ \\ \end{array}\right]}^{N}\right)$　式（19）

$$A = \left[\begin{array}{c} \vec{e}_1^{\,T} \\ \vec{e}_2^{\,T} \\ \vdots \\ \vec{e}_N^{\,T} \end{array}\right] \qquad \phi_{\vec{f}}\vec{e}_k = \lambda_k\vec{e}_k \ (k = 1,2,\cdots,N) \quad \text{固有値問題}$$
$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$$

このとき

$$\phi_{\vec{t}} = \left[\begin{array}{ccc} \lambda_1 & & \\ & \lambda_2 & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{array}\right]$$

変換係数は無相関　式（21）

KL変換の基底を用いた信号の展開

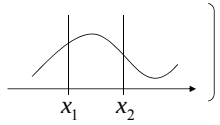
$$\vec{f} = \sum_{k=1}^{N}(\vec{f},\vec{e}_k)\vec{e}_k = A^T(A\vec{f}) \quad \text{-> KL展開} \qquad \text{式（22）}$$
$$T_f(k)$$

KL変換の性質

1. 信号の相関を完全に取り除く
2. 低次の変換係数へのエネルギー集中率が最良

（※共分散関数

$$\phi(x_1,x_2) = E[f(x_1)f(x_2)]$$



）

$\vec{f}$ の相関が強い -> $\phi_{\vec{f}}$ の対角以外の要素が大きい
$\vec{f}$ の相関が弱い -> $\phi_{\vec{f}}$ は対角行列に近い

変換係数 $\vec{t}$ の共分散行列

$$\phi_{\vec{t}} = E[\vec{t}\vec{t}^{\,T}] = E[A\vec{f}\vec{f}^{\,T}A^T] = A\phi_{\vec{f}}A^T \qquad \text{式（20）}$$

信号の相関を完全に除去するには $\phi_{\vec{t}}$ が対角行列となるように A を決めればよい

解　A が $\phi_{\vec{f}}$ を対角化するユニタリ行列であれば良い（線形代数）

※エネルギー集中率の調べ方

$\hat{\vec{f}} \approx \sum_{k=1}^{M}(\vec{f},\vec{e}_k)\vec{e}_k \ (M < N)$ と展開を少数項で打ち切る

このとき $E\left[\left\|\vec{f} - \hat{\vec{f}}\right\|^2\right]$ が小さいほどエネルギー集中率が高い

$$E\left[\left\|\vec{f} - \hat{\vec{f}}_{KL}\right\|^2\right] \leq E\left[\left\|\vec{f} - \hat{\vec{f}}\right\|^2\right] \qquad \text{式（23）}$$

KL変換　その他の変換

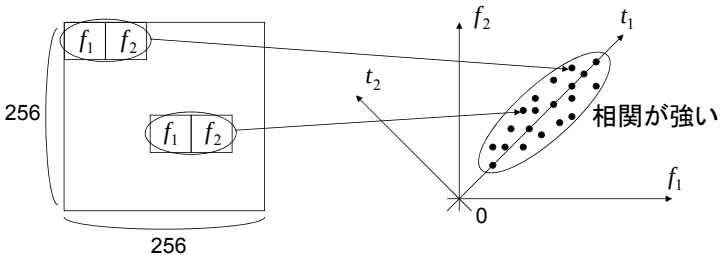
種々の変換のエネルギー集中率の比較 -> 図11　図12

KL変換の問題点

1. 高速な計算法がない
2. エネルギー集中率もDCTとあまり違わない → ほとんど使われていない
3. 基底が情報源の性質に依存する

変換符号化の幾何学的解釈

初等的な2画素を1ブロックとする符号化



(f_1, f_2)で符号化 -> f_1と f_2に同じビット数

(f_1, f_2)に以下の直交変換を施す

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

45度の座標軸の回転

(t_1, t_2) で符号化 -> t_2のビット数は少なくてよい（情報圧縮）

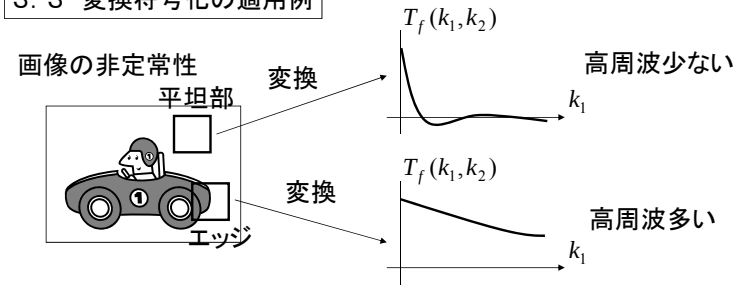
歴史的に重要な論文

W.Chen and C.Smith," Adaptive coding of monochrome and color images," IEEE Transactions on Communications, Vol.25, pp.1285-1292, 1977

W.Chen and W.Pratt," Scene adaptive coder," IEEE Transactions on Communications, Vol.32, pp.225-232, 1984

この2つがJPEGの基礎となっている

3．3　変換符号化の適用例



平坦部とエッジ部を同じに扱ってビット割り当て -> 性能劣化

性能をあげるキーポイント -> 信号のアクティビティに応じてビット配分を変える

適応

適応をどう取り入れるかの違いで様々な符号化法が存在

（1）適応DCT符号化（Chen and Smith, 1977）

ブロック符号化 { 変換 -> DCT ／ 量子化 -> 適応量子化 } ← キーポイント

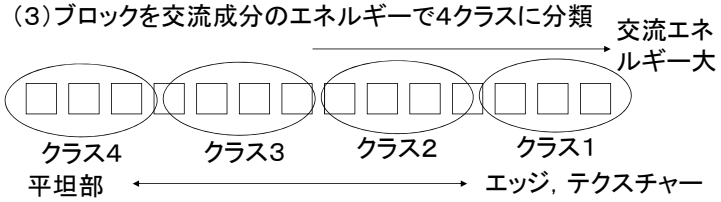
適応量子化 -> 量子化ビット配分をブロックによって変える

手順

（1）原画像をブロックに分割

（2）各ブロックをDCT

（3）ブロックを交流成分のエネルギーで4クラスに分類



（4）量子化ビット配分を各クラスについて別々に計算
（クラス1 -> 多くのビット，クラス4 -> 少ないビット）

（5）量子化

図13

実験例 -> 図14

（2）シーン適応符号化（Chen and Pratt, 1984）

JPEGにかなり近い

ブロック符号化
- 変換 -> DCT
- しきい値処理＋ジグザグ走査 ← キーポイント
- ハフマン符号化

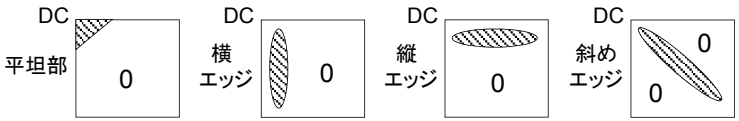
手順
（1）ブロック分割
（2）DCT
（3）しきい値処理 -> 小さな変換係数を零にする

$$T_f(k_1,k_2)=\begin{cases} T_f(k_1,k_2) & (|T_f(k_1,k_2)|\geq T) \\ 0 & (|T_f(k_1,k_2)|<T) \end{cases}$$

← しきい値

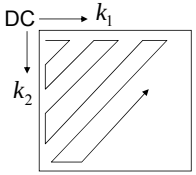
信号のアクティビティが低い所を適応的に捨てる

非零変換係数の分布



（4）非零変換係数を量子化
（5）ジグザグ走査で1次元列に変換

零の続きはラン長で記憶 -> 大幅な圧縮



零がたくさん続くように

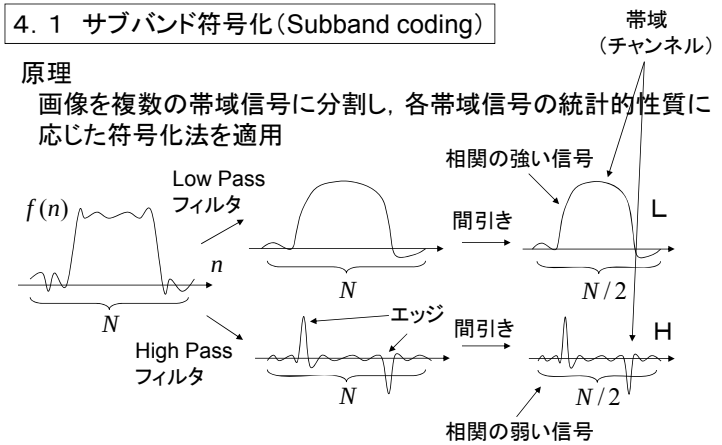
（6）ハフマン符号化

実験例 -> 図15

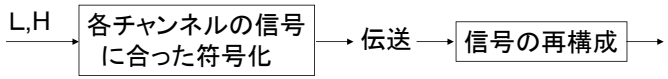
第4章：その他の符号化法

4．1　サブバンド符号化（Subband coding）

原理
画像を複数の帯域信号に分割し，各帯域信号の統計的性質に応じた符号化法を適用



相関の強い信号のデータ数半減＝冗長度削減

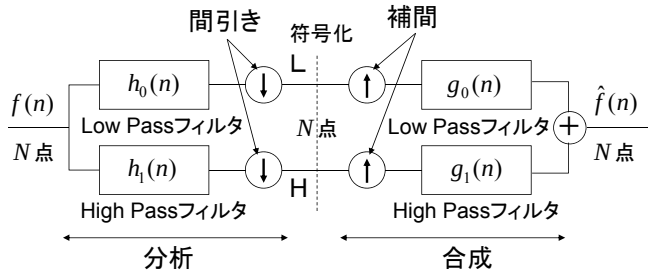


例 Low Pass -> DPCM
High Pass -> PCM

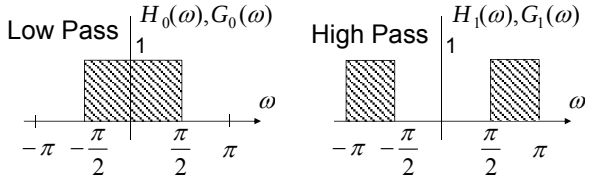
キーポイント
1．信号の分解による冗長度削減
2．各チャンネルの信号に異なる符号化法を適用

（1）信号の分解と再構成

フィルタバンク　次ページ



各フィルタの特性



※ディジタル信号のフーリエ変換は周期 2π の周期関数だから
$-\pi \le \omega \le \pi$ の周波数特性を決めれば良い

間引き（decimation） -> 信号の値を1つおきに間引く

$f(n)$ 入力 → ↓ → $f(2n)$ 出力

補間（interpolation） -> 信号の値の1つおきに零を挿入

$f(n)$ 入力 → ↑ → $f'(n)$ 出力　　$f'(n) = \begin{cases} f(n/2) & (n:\text{even}) \\ 0 & (n:\text{odd}) \end{cases}$

完全再構成（$f(n) = \hat{f}(n)$）の条件

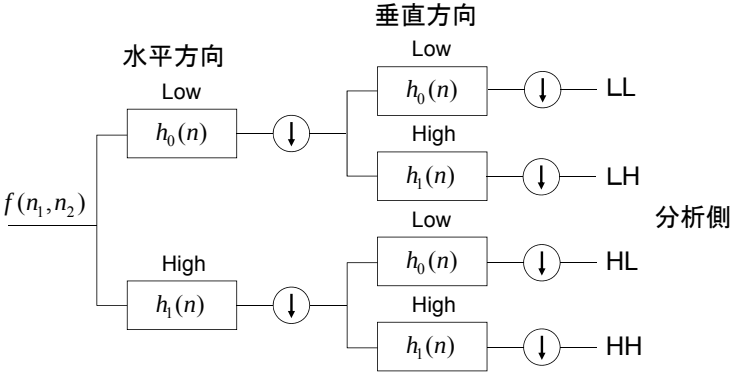
$$\begin{cases} H_0(\omega)G_0(\omega) + H_1(\omega)G_1(\omega) = 2 & \text{（分析＋合成が平坦な周波数特性）} \\ H_0(\omega+\pi)G_0(\omega) + H_1(\omega+\pi)G_1(\omega) = 0 & \text{（エリアシングが消える）} \end{cases}$$

式（24）

完全再構成フィルタの設計 -> 理論がある

2次元の場合

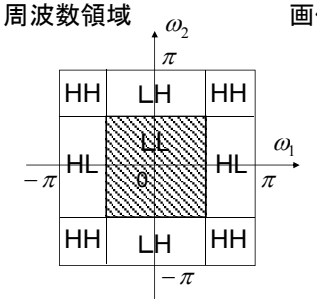
4channelに分割する



合成側も同様

図16

周波数領域



画像

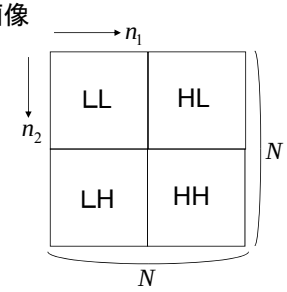
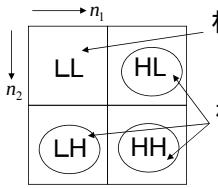


図17

分解例 ->

図18

（2）初期のサブバンド符号化（Woods and Neil, 1986）



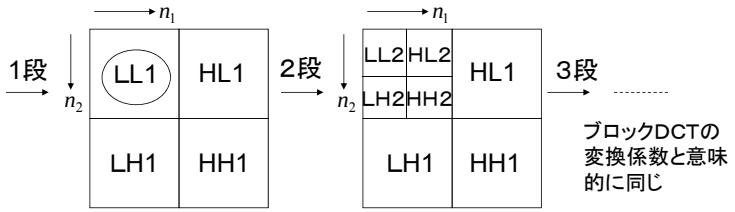
相関が強い

相関が弱い

LL -> 相関が強いので予測符号化
LH,HL,HH -> 相関が弱いのでPCM
（スカラー量子化）

（3）最近のサブバンド符号化

4channelのフィルタバンクを縦続接続する



段数を増やす -> 相関の強いLL成分のサイズが小さくなる
（より大きい冗長性の削減）

量子化の方法
1. 各チャンネルの統計的性質からビット配分を決め，全チャンネルをPCM（スカラー量子化）
2. シーン適応符号化の考え方を導入 -> JPEG2000

信号の分解がWavelet変換と関係 -> Wavelet符号化

実験例 -> 図19

特徴
1. 変換符号化で生じるブロック歪みが出ない
2. 階層的伝送（Progressive transmission）に適している

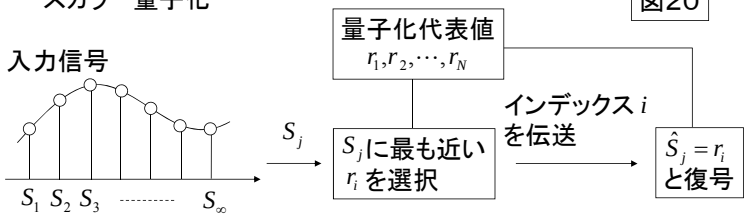
4．2 ベクトル量子化（Vector Quantization）

量子化すること自体がデータ圧縮になっている

（1）スカラー量子化とベクトル量子化

スカラー量子化

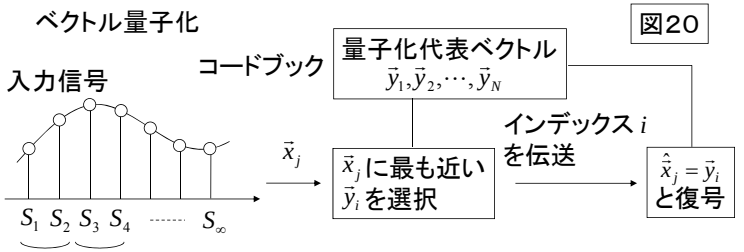
図20



各 S_j を独立に量子化する

ベクトル量子化

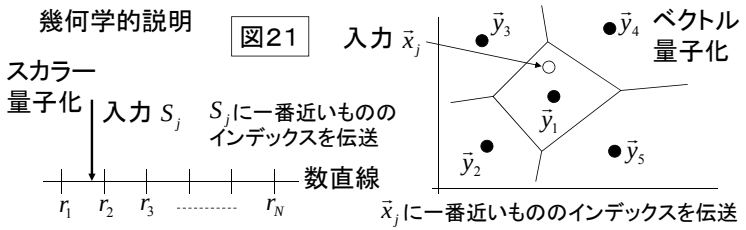
図20



$\vec{x}_1 = (S_1, S_2)^T$ $\vec{x}_2 = (S_3, S_4)^T$ 複数個を組にする

$\vec{x}_3 = (S_5, S_6)^T$ 普通ベクトルの次元4,16,64ぐらい

幾何学的説明 図21

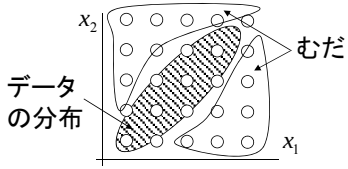


（2）ベクトル量子化が優れている理由

量子化レベルの選択に柔軟性がある

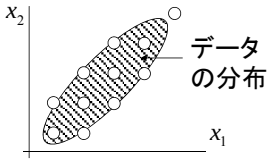
例（a）相関の強い信号の量子化

スカラー量子化



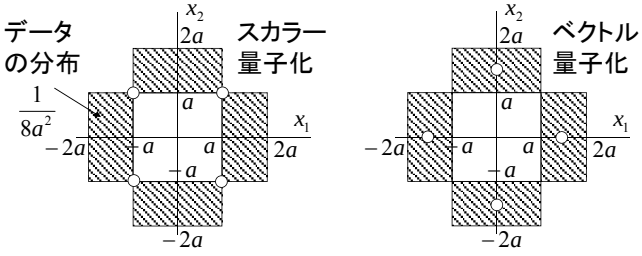
格子状にしか量子化
レベルを設定できない

ベクトル量子化



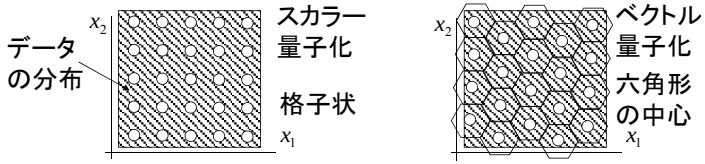
任意に量子化レベルを設定可能
-> ビット数の削減

（b）データの分布が複雑な場合



同じビット数でもベクトル量子化の方が量子化歪が少ない

（c）相関の弱い信号の量子化

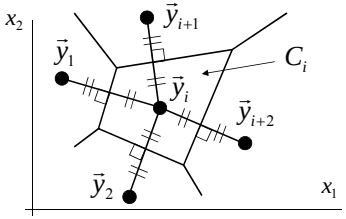


あまり効果がないがビット数が少し少なくて済む

（3）コードブックの設計

LBGアルゴリズム（Linde,Buzo,Gray, 1980） -> 反復法

K平均法，クラスタリング



$\vec{y}_i$ -> 量子化レベル

C_i -> ボロノイセル（$\vec{y}_i$が量子化レベルとして選択される範囲）

量子化歪の期待値

$$\begin{cases} D = E[d(\vec{x},\hat{\vec{x}})], \ \underbrace{d(\vec{x},\hat{\vec{x}}) = \left\|\vec{x}-\hat{\vec{x}}\right\|^2}_{\text{距離}} \\ \hat{\vec{x}} = \underbrace{VQ(\vec{x})}_{\text{ベクトル量子化}} \end{cases} \quad \text{式（25）}$$

考え方

トレーニングベクトル $\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_M$ を与え，D が最小になるような $(\vec{y}_i, C_i)\,(i = 1,2,\cdots,N)$ を決める

$$D \approx \frac{1}{M}\sum_{j=1}^{M} d(\vec{x}_j, \hat{\vec{x}}_j) \quad \text{式（26）}$$

手順

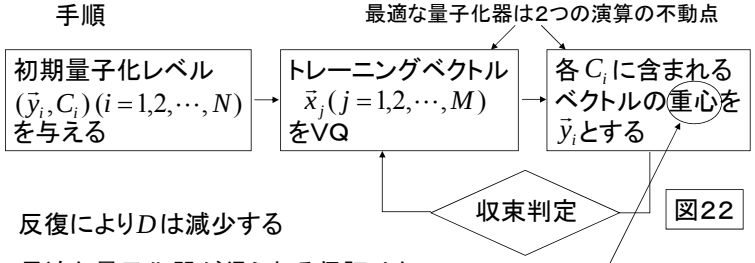


図22

反復によりDは減少する

最適な量子化器が得られる保証はない
-> 局所解のため

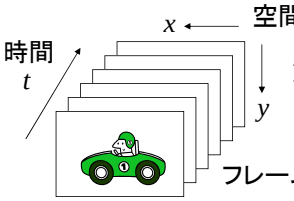
平均 $\vec{y}_i = \frac{1}{N_i}\sum_{\vec{x}_j \in C_i} \vec{x}_j$

（4）ベクトル量子化の問題点

・計算量が多い -> 距離の計算，コードブックの記憶
・コードブックの設計

実験例 -> 図23

4．3　動画像の符号化

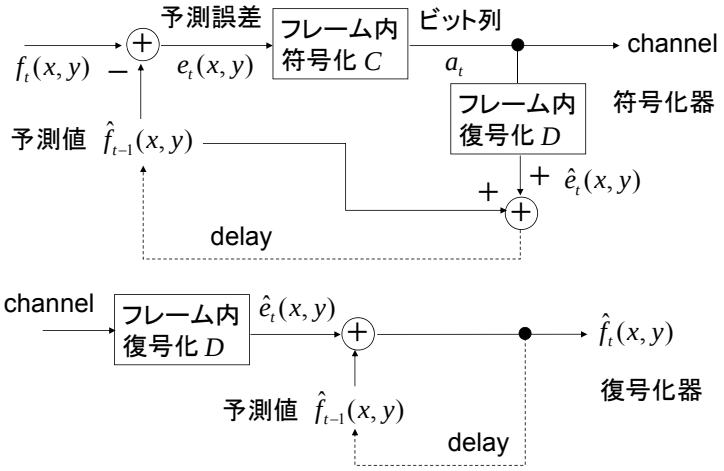


動画像の冗長性
＝フレーム内の相関＋フレーム間の相関
重要

考え方

フレーム内では変換符号化,フレーム間は予測符号化

フレーム間予測符号化（前のフレームを次のフレームの予測に使う）



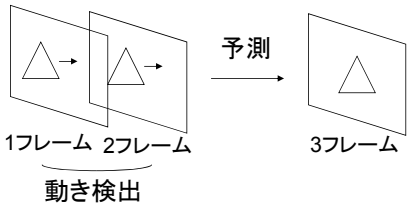
$$\begin{cases} e_t(x,y) = f_t(x,y) - \hat{f}_{t-1}(x,y) & \text{予測誤差} \\ a_t = C[e_t(x,y)] & \text{フレーム内符号化} \\ \hat{f}_t(x,y) = \hat{f}_{t-1}(x,y) + D[a_t] & \text{復号化} \end{cases} \quad \text{式（27）}$$

予測誤差 $e_t(x,y)$ -> 動いた部分のみ大きくなる　図24

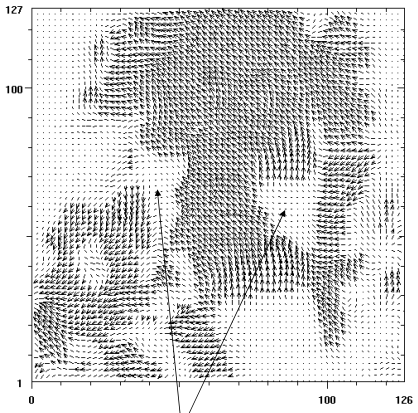
動き補償予測 -> より正確な予測

動物体の動きを検出し，動き量を補正して次のフレームを予測する

オプティカルフロー



※オプティカルフローの例



色が均一な部分は正確に求めるのが困難