

インターネット数理科学第6回

～ネットワークのこちら側を支える数理科学その2～

2006年11月9日

株式会社インターネット総合研究所代表取締役所長
東京大学大学院数理科学研究科客員教授

藤原 洋

目 次

1. ネットワークのこちら側を支える数理科学とは？
2. ネットワークと動画像の関係
3. 動画像圧縮の原理
4. 変換符号化 ~空間的圧縮のために~
5. 予測符号化 ~時間的圧縮のために~
6. その他の圧縮手法 ~VQとサブバンド符号化~
7. エントロピー符号化

1. ネットワークこちら側を支える数理科学とは？

③(ネットワークの)あちら側

⇒「グラフ理論」「金融工学理論」に基づくデータベース、検索エンジン最適化、検索連動データベース、ネット金融サービス

①ネットワークそのもの

⇒「グラフ理論」による動的ルーティング、帯域制御、放送型ルーティング
「デジタル信号処理理論」に基づく変復調技術

②(ネットワークの)こちら側

⇒「デジタル信号処理理論」に基づくコンテンツ符号化技術
⇒「HTML」に基づくブラウザ技術

以下の3つの分野にわたって①②③⇒①②③⇒・・・順に

③ネットワークのあちら側を支える数理科学

⇒「グラフ理論」「金融工学理論」に基づくデータベース、検索エンジン最適化、検索連動データベース、ネット金融サービス

①ネットワークそのものを支える数理科学

⇒「グラフ理論」による動的ルーティング、帯域制御、放送型ルーティング
「デジタル信号処理理論」に基づく変復調技術

②ネットワークのこちら側を支える数理科学

⇒「デジタル信号処理理論」に基づくコンテンツ符号化技術

③(ネットワークの)あちら側

Web1.0(ポータル)⇒Web1.5(SNS)⇒Web2.0(ロングテール)

①ネットワークそのもの

ダイヤルアップ/2Gモバイル⇒ブロードバンド/3Gモバイル⇒IP放送/NGN/WiMAX

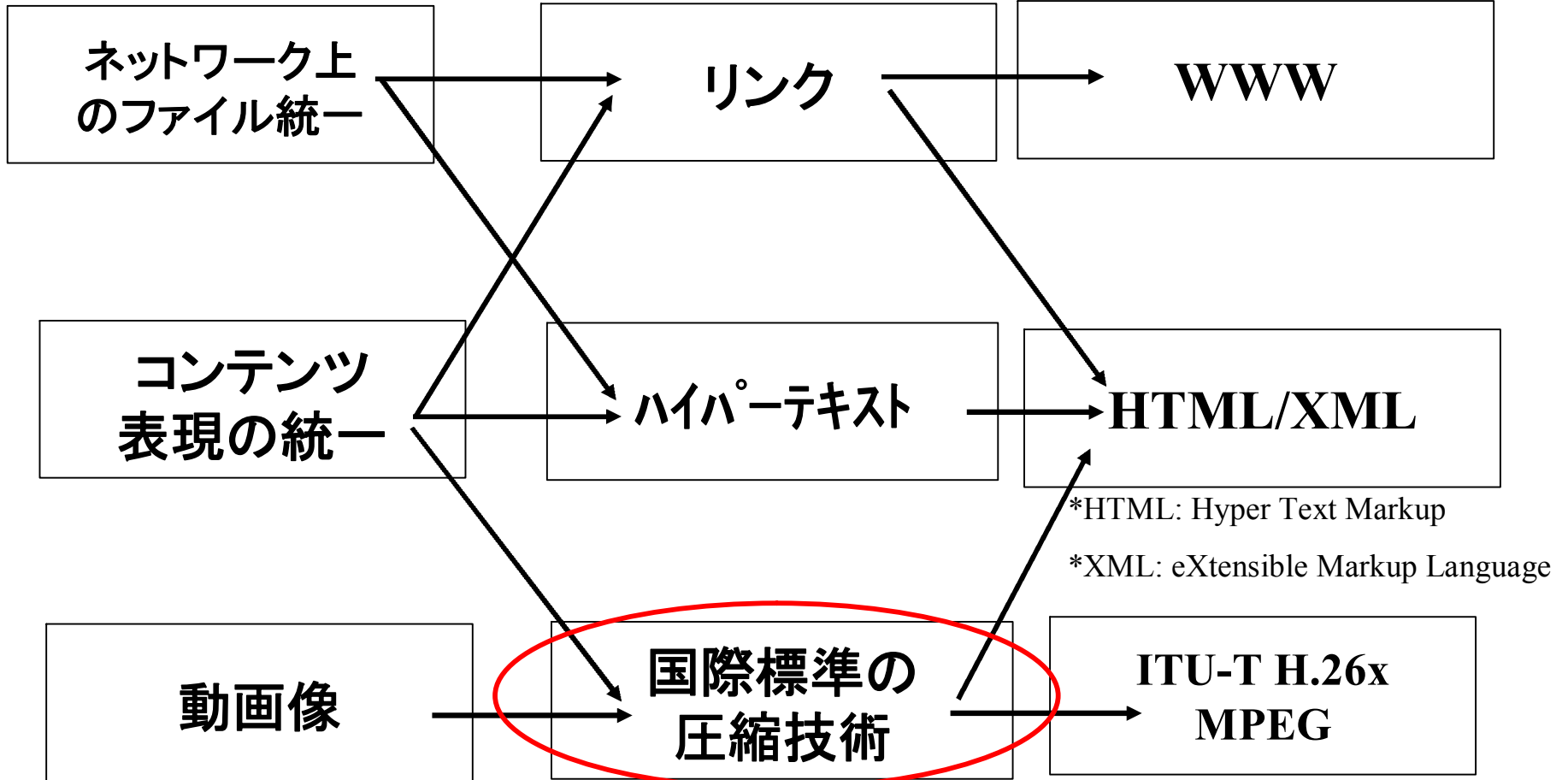
②(ネットワークの)こちら側

文字情報(Eメール)⇒HTML(ブラウザ)⇒動画(デジタル符号変換)

課題

着眼点

具体策



*HTML: Hyper Text Markup
*XML: eXtensible Markup Language

*ITU-T: International Telecommunications Union-Telecommunication Sector

*MPEG: Moving Picture Experts Group

「通信と放送の融合」というキーワードは、いつも通信業界から発せられてきた。旧電電公社時代の三鷹でのINS実験、狭帯域N-ISDNによる**TV電話の発展形としての動画像サービス**、ATM交換に基づくB-ISDNによるVoD、そして今話題となっているのが「ブロードバンド登場後のIP放送」。

通信ネットワークを用いた動画像サービスは、まさに4度目の挑戦となる。これまでの3回は、失敗だったが、今度はモノになるだろうか？この動向を展望する上で、現段階で最も積極的にサービスを展開している、「GyaO」と「TV-Bank」に目を向ける。

現行のTV放送を補完する2つのサービス

GyaOは、約1000タイトルのVoDとリアルタイム配信の組み合わせであり、TV-Bankは、他のサイトへのリンクを含めて約10万タイトルのVoDとリアルタイム配信の組み合わせです。GyaOが、編成と自主制作に力点を置いた擬似TV放送型であるのに対して、TV-Bankは、コンテンツ収集力に力点を置いたWeb1.0検索型であるといえる。また、ビジネスモデルとしては、GyaOが無料広告モデルであるのに対して、TV-Bankは、無料と有料の組み合わせモデルとなっています。どちらも現行のTV放送のテイストを残しつつも、多くの選択肢のあるVoDブロードバンド・インターネットならではの長所を活かしたサービスとなっている。

実際に、財団法人インターネット協会監修の2005年版『インターネット白書』のインターネット利用者の映像配信に対する利用動向調査によると、明らかに現行TV放送とは異なる、あるいはそれを補完するサービスのニーズが高いことがわかる。

以上のことから、GyaOとTV-BankというITベンチャーによる**インターネットを用いた動画像サービス**は、現行放送と競合するサービスではなく、補完するサービスであるといえる。また、両者合わせて1000万人程度のアクティブユーザーを獲得しているとみられ、事業的にも損益分岐点に到達するのは時間の問題という状況にあり、4度目の挑戦は、初めての成功が近いと思われるが、NHK、民放など、現行放送の約3兆円市場と比較すると、IP放送市場がその1%まで到達するには相当の時間がかかるだろう。

2. ネットワークと動画像の関係

マルチメディア情報圧縮（インターネット時代の数学シリーズ）



藤原 洋 編著

税込価格：¥2,520（本体：¥2,400）

出版：共立出版

サイズ：A5判 / 146p

ISBN：4-320-01644-0

発行年月：2000.3

端末AとBとの間に電話番号を発呼して交換機が動作し通信中は回線が接続状態になる ⇒ 「回線交換」: 帯域保証ネットワークの原型

*コネクション型ネットワーク

仮定：端末AとBが128Kbpsで、端末CとDが64Kbpsで通信を行う

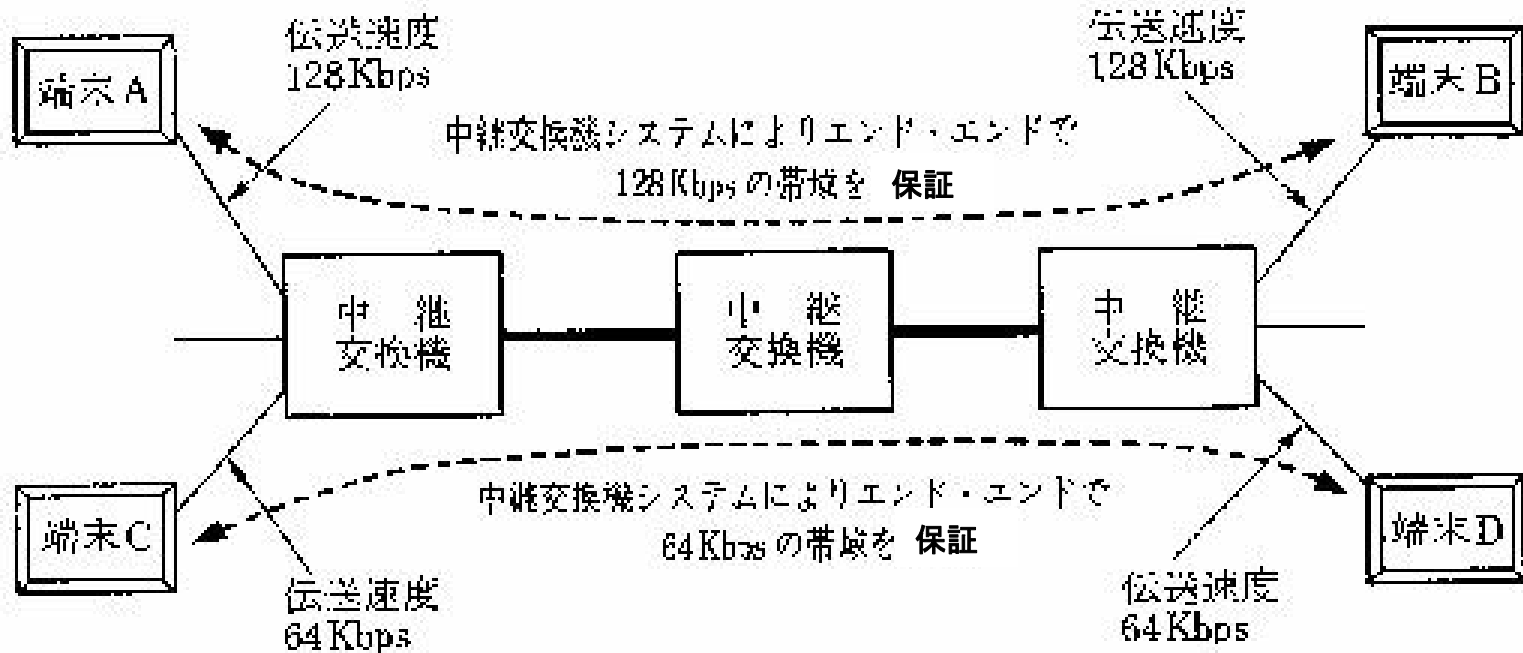


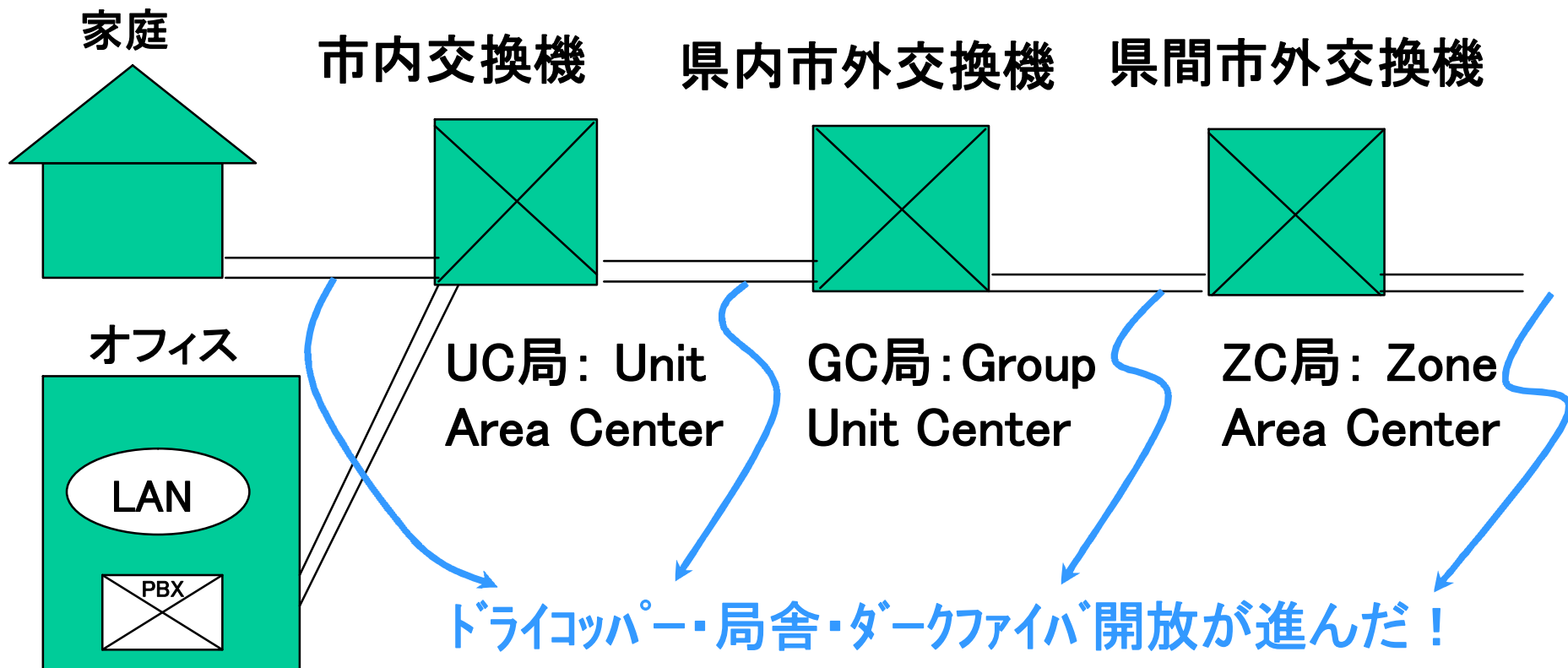
図 1.1 帯域保証ネットワーク

アクセスネットワーク

バックボーンネットワーク

ツイストペア線

光ファイバ



パケット交換によるベストエフォート・ネットワーク

端末AからBへはひとかたまりのデータ(パケット)の先頭に宛先アドレスと送元アドレスを付加して送受信する。パケット交換機の蓄積交換機能によって正しく宛先に届く ⇒ 「パケット交換」: ベストエフォート・ネットワークの原型

* コネクションレス型が基本

* コネクション型もある

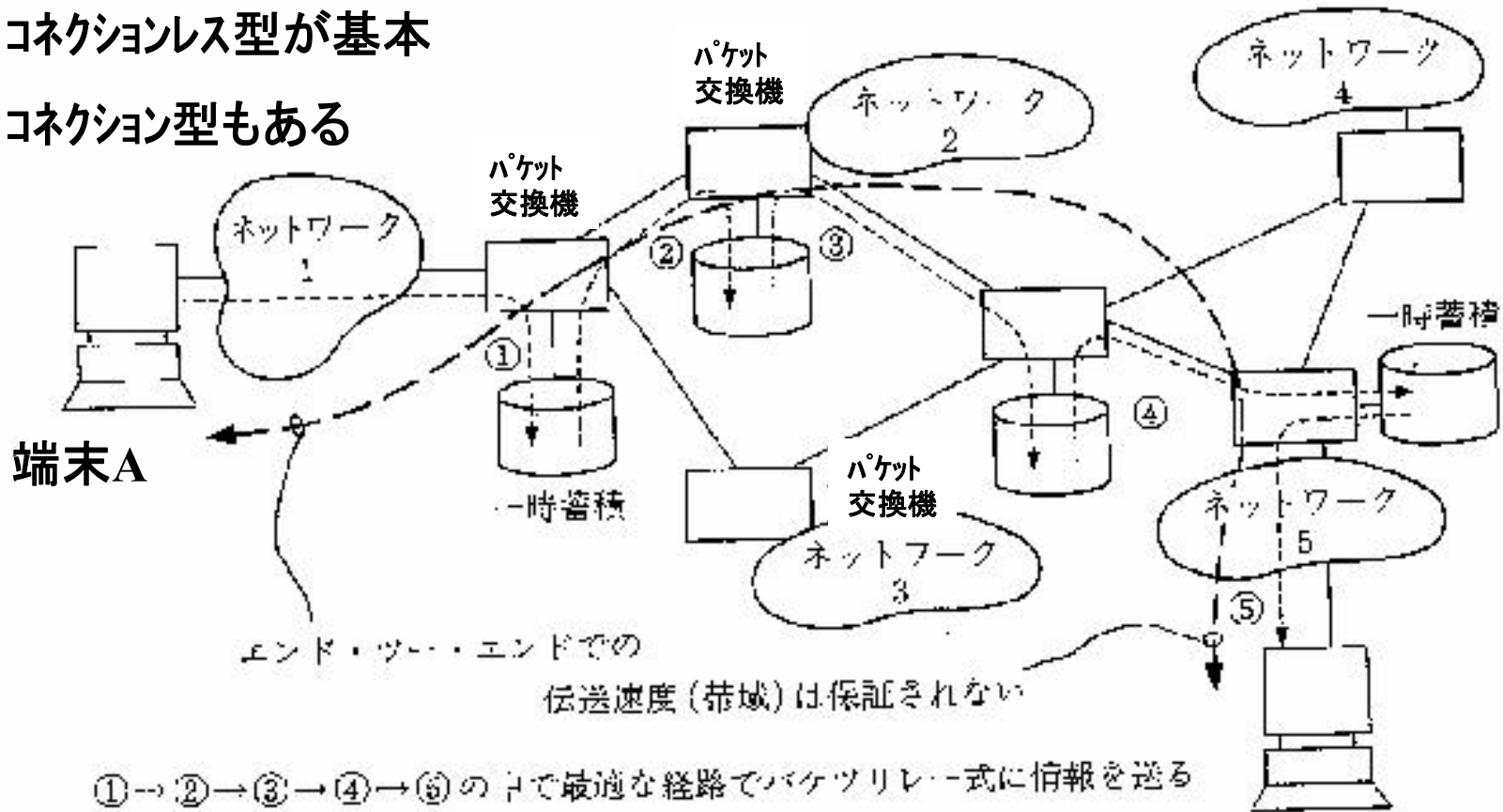


図 1.2 ベストエフォート・ネットワーク

表 1.1 2つのネットワークの比較

比較項目	ベストエフォート	帯域保証
交換方式	パケット交換	回線交換, ATM 交換
パケットロス	あり得る	回線交換 : なし ATM 交換 : あり得る
交換装置	ルータ	交換機
ネットワークコスト比 (大規模システム)	1	10~100
適したサービス	インターネット	音声, 動画像

表 1.2 ネットワーク伝送速度と画像・音声情報の伝送速度との関係

オーディオ	品質	信号帯域 (kHz)	サンプリング 周波数 (kHz)	間引き後 ビット・レート (bps)	圧縮後 ビット・レート (bps)
	電話音声		3.4	8	64k
オーディオ	AM 放送	7	16	130k	24k
	FM 放送	7(ステレオ) または 14(モノラル) 10(ステレオ) または 20(モノラル)	16 または 32	510k	56k
			22.05 または 44.1	700k	64k
音楽 CD	20(ステレオ)	44.1	1.4M	112k~224k	
ビデオ	品質	空間解像度 (縦×横)	フレーム・ レート/秒	間引き後 ビット・レート (bps)	圧縮後 ビット・レート (bps)
	ビデオ・ クリップ	80 × 60	1	55k	1.8~2.8k
			3	165k	6~9k
			10	550k	18~28k
	1/4 画面	160 × 120	3	1.1M	20~33k
			10	2.2M	70~110k
30			6.7M	220~335k	
VTR (VHS)	360 × 240	10	10M	330~500k	
		30	30M	1~1.5M	
テレビ放送	720 × 480	30	120M	4~6M	

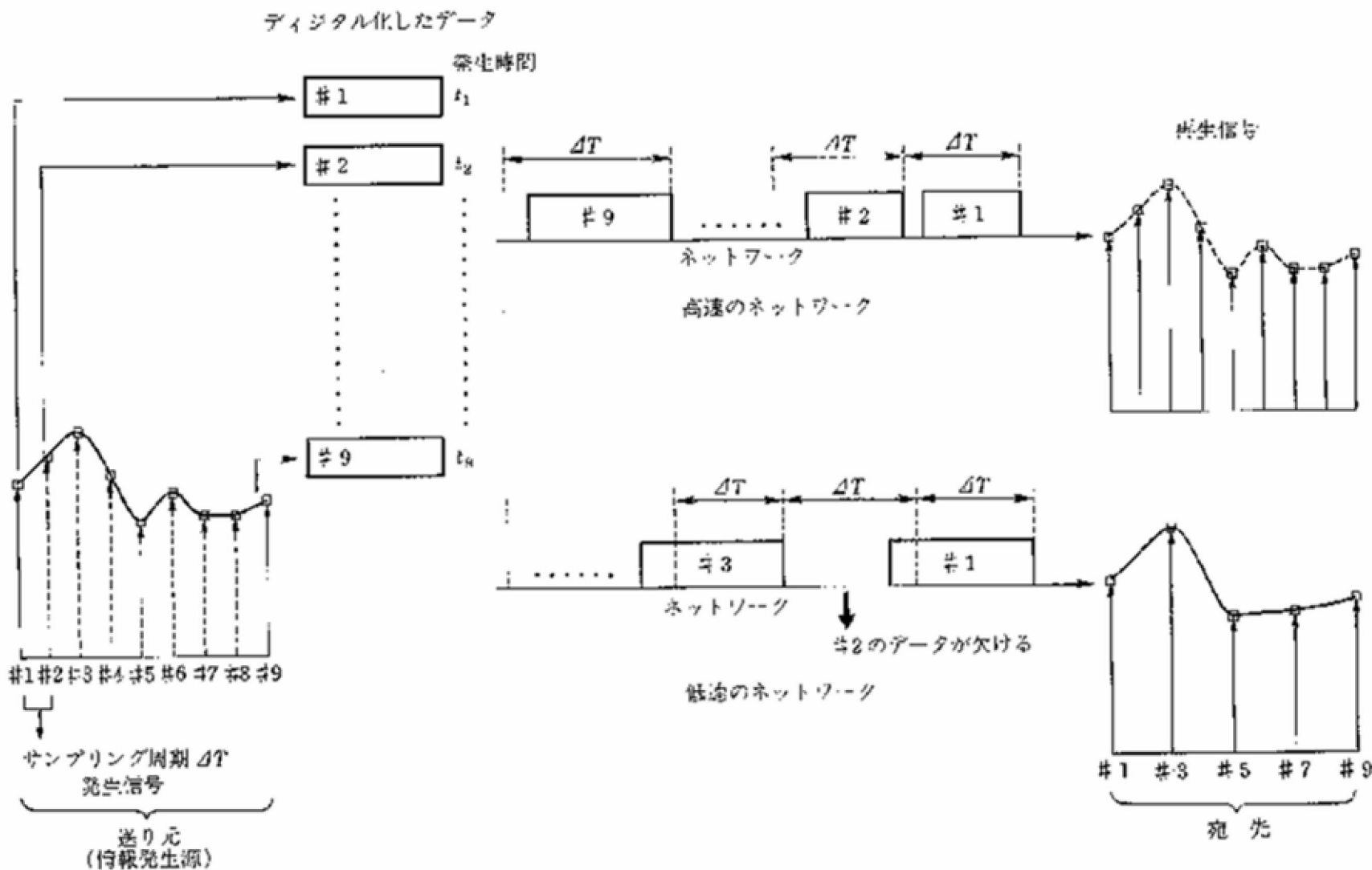


図 1.3 画像・音声情報の発生速度と伝送時間

3. 動画像圧縮の基本原則

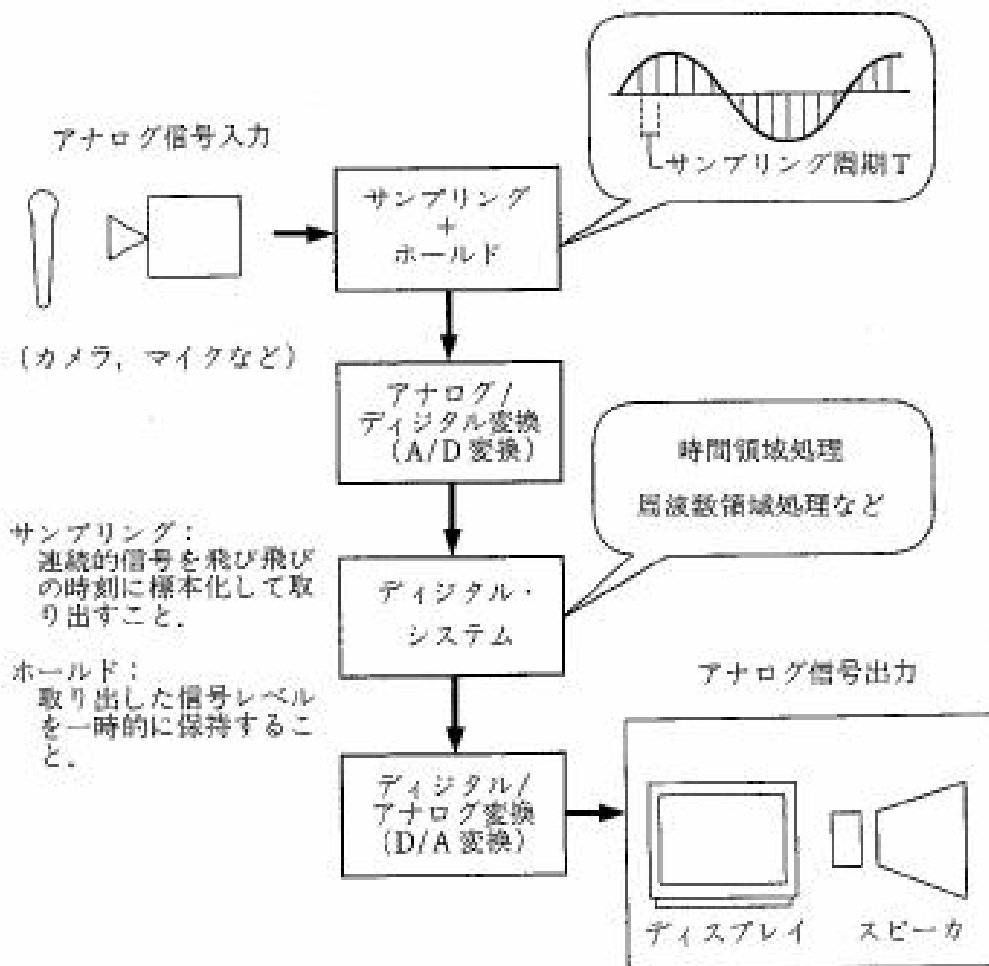


図 1.4 情報圧縮処理の基本モデル

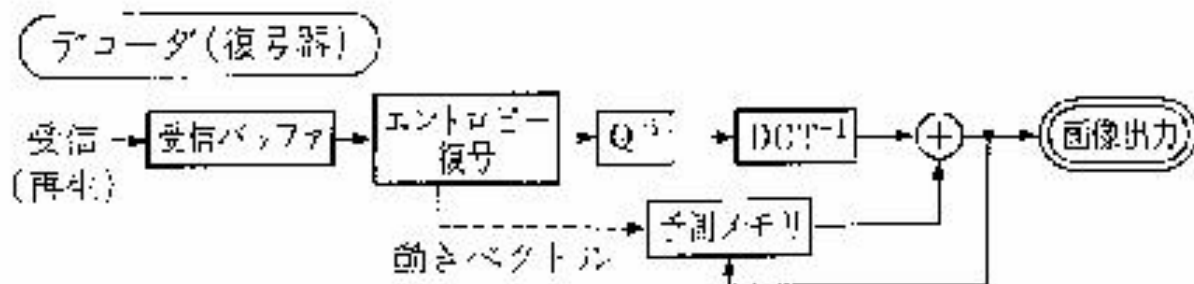
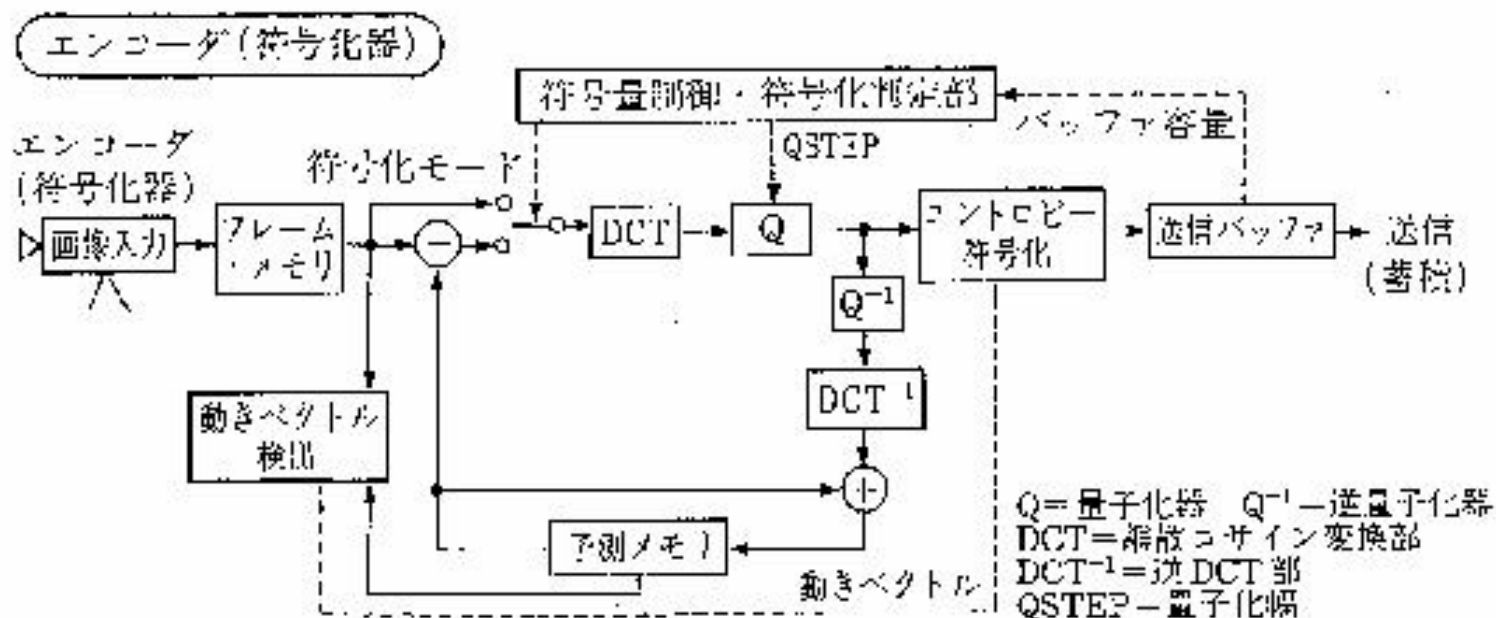


図 1.5 ハイブリッド動画像情報圧縮処理モデル

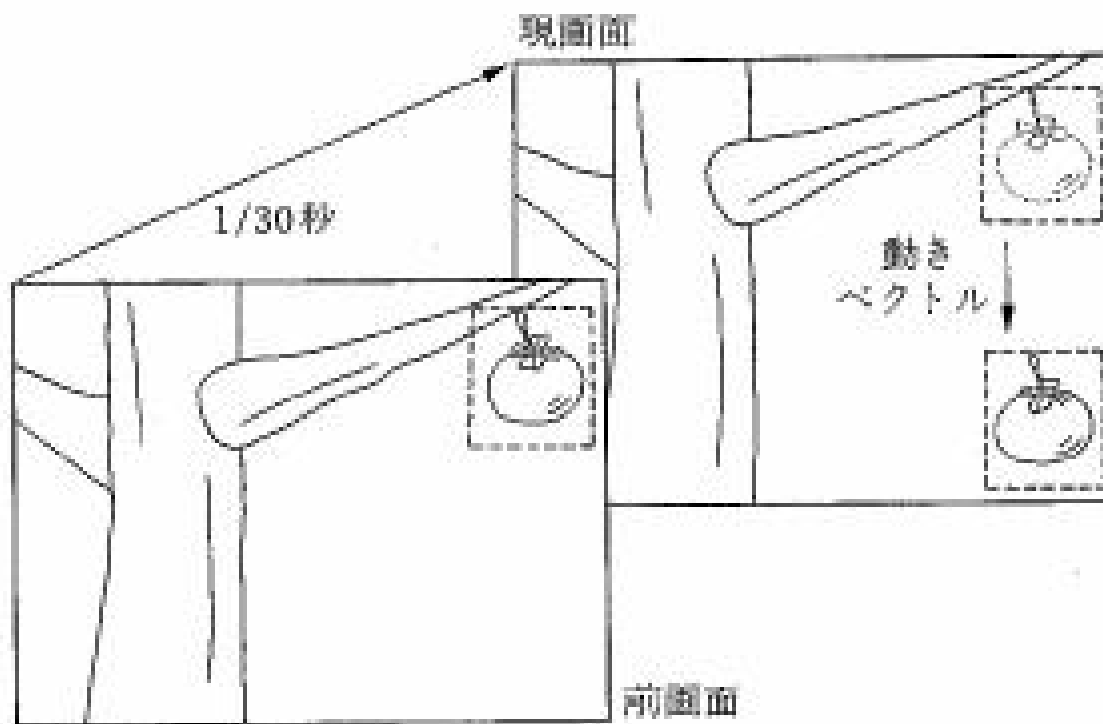


図 1.6 動き補償の原理

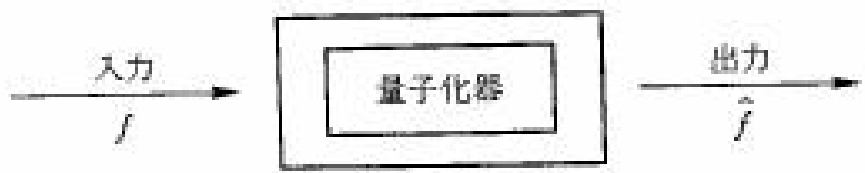


図 1.7 量子化器

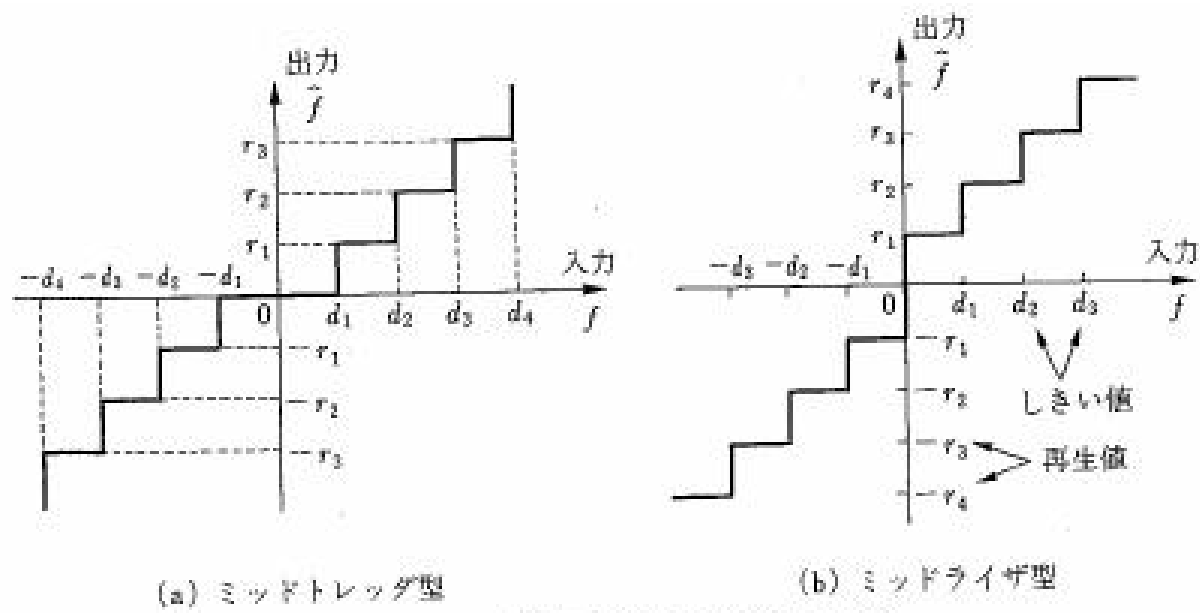


図 1.8 一様量子化器での入出力関係

量子化器設計における設計手法としてまず重要なのは、量子化誤差を抑えることである。これには、主として式 (1.1) から (1.4) に示す 4 つの誤差表現が用いられている。

① 平均二乗量子化誤差 (MSQE : Mean Square Quantization Error)

$$E[(f - \hat{f})^2] = \int_{a_L}^{a_U} (f - \hat{f})^2 p(f) df \quad (1.1)$$

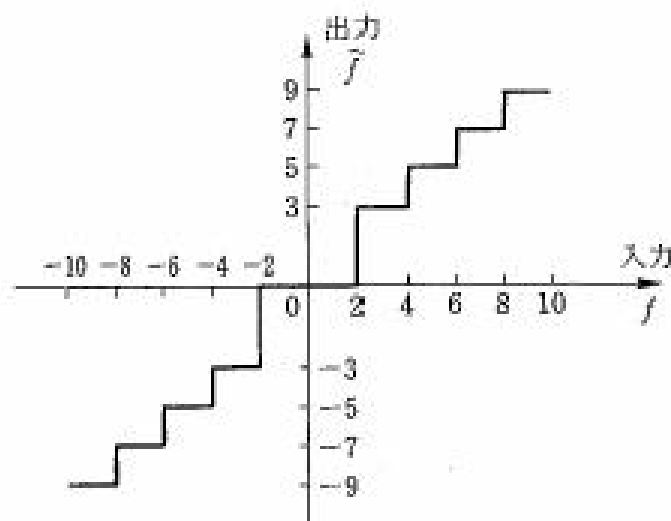


図 1.9 準一様量子化器での入出力関係

② 平均絶対値量子化誤差 (MAQE : Mean Absolute Quantization Error)

$$E[|f - \hat{f}|] = \int_{a_L}^{a_U} |f - \hat{f}| p(f) df \quad (1.2)$$

③ 平均 L_n 正規量子化誤差

$$E[|f - \hat{f}|^N] = \int_{a_L}^{a_U} |f - \hat{f}|^N p(f) df \quad (1.3)$$

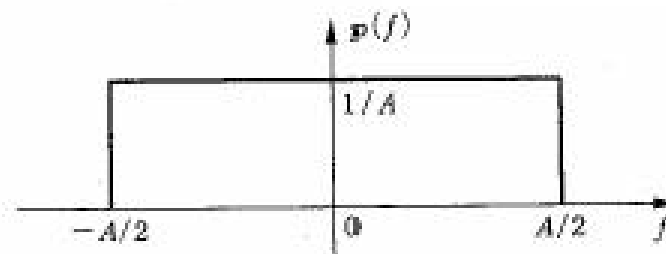
④ 重みづけ量子化誤差

$$\int_{a_L}^{a_U} w(f) |f - \hat{f}| p(f) df \quad (1.4)$$

以上の量子化誤差を一様量子化器の例について評価すると以下のようなになる。

まず、一様量子化器の入力信号の入力信号に対する確率密度関数を図 1.10 のように仮定すると式 (1.5)~(1.7) が成立する。

$$p(f) = \frac{1}{a_U - a_L} = \frac{1}{A} \quad (1.5)$$



1.10 一様確率密度関数

$$\begin{aligned} r_k &= \frac{\int_{d_k}^{d_{k+1}} f p(f) df}{\int_{d_k}^{d_{k+1}} p(f) df} \\ &= \frac{1}{2} (d_{k+1} + d_k) \end{aligned} \quad (1.6)$$

$$d_{(k)} = \frac{r_k + r_{k-1}}{2} = \frac{d_{k+1} + d_{k-1}}{2} \quad (1.7)$$

したがって、

$$\begin{aligned} d_k - d_{k-1} &= d_{k+1} - d_k \\ &= \text{ステップサイズ一定} \\ &= \frac{a_U - a_L}{J} \end{aligned} \quad (1.8)$$

量子化誤差は、全領域で一様であるため上記式 (1.1) の MSQE は、以下のようになる。

$$\varepsilon = \frac{1}{SS} \int_{-SS/2}^{SS/2} f^2 df = \frac{(SS)^2}{12} \quad (1.9)$$

f の範囲は A としたため、分散 (Variance) は、以下のようになる。

$$\sigma_f^2 = \frac{1}{A} \int_{-A/2}^{A/2} f^2 df = \frac{A^2}{12} \quad (1.10)$$

ここで b ビットの量子化器は、出力レベル J で出力されるため次式となる。

$$\text{SNR} = \frac{\text{Variance}}{\text{MSQE}} = \frac{A^2/12}{\left(\frac{A^2}{2^{2b}}/12\right)} = 2^{2b} \quad (1.11)$$

$$(\text{SNR})_{\text{dB}} = 10 \log_{10}(\text{SNR}) = 20b \log_{10} 2 \cong 6b \text{ dB}$$

このことから、一様量子化器の平均二乗誤差は、1 ビット当たり 6 dB 増加することがわかる。

4. 変換符号化

~空間圧縮のために~

1.2.3 変換符号化

変換符号化は、信号領域をある関数によって変換し、別の領域へ写像することである。一般には、直交関数を用いるため可逆である。直交変換では変換後もエネルギーが保存され、逆変換による完全再生が可能である。変換は、

一次元でも多次元でもあり得るが、多次元については一次元への分解が可能な変換が実装上有利であるため、可分多次元変換を情報圧縮処理の対象とする。

情報圧縮に適用可能な直交変換は、1970年代初頭から今日まで実に多くの変換が考えられてきたが、実質的に有効なのはDCTである。このため、ここでは、DCTを中心に考え方を示す。DCTは、現在最も広く情報圧縮アルゴリズムとして採用されているものである。

(1) DCT の基礎

DCT の議論に入る前に、もとになる KLT (Karhunen-Loeve 変換) についてふれておく。KLT (式 (1.12)) は、カルーネン・レーベ変換と呼ばれ、定常確率過程において理論的に最適変換である。最適の意味は、無相関化、最大エネルギー寄与率、および最大符号化利得の 3 つである。しかしながら、KLT は理論的には最適だが、信号に依存する変換であり高速アルゴリズムが存在しないことから非実用的である。

KLT は、変換領域において信号を完全に無相関化するためのもので、 N サンプル点によるゼロ平均ランダムベクトル \mathbf{x} について、

$$\mathbf{x} = \{x(0), x(1), \dots, x(N-1)\}^T \quad (1.12a)$$

が成立し、線型独立の N 次元ベクトル空間を $\{\Phi_i\}$ とすると

$$\mathbf{x} = \sum_{i=0}^{N-1} X_i \Phi_i \quad (1.12b)$$

展開式 X_i は、以下の係数となる。 $\langle \cdot, \cdot \rangle$ は内積を表す。

$$X_i = \langle \mathbf{x}, \Phi_i \rangle / \langle \Phi_i, \Phi_i \rangle, \quad i = 0, 1, \dots, N-1 \quad (1.12c)$$

この式で示す解のうち、はじめの係数 D 個が 0 と大きく異なるとすると、最初の D 個の係数で十分表されるため、打切り表現を

$$\hat{\mathbf{x}} = \sum_{i=0}^{D-1} X_i \Phi_i \quad (1.12d)$$

で表す。このとき、MSE（平均自乗誤差）は、 E を期待値演算子として、

$$\begin{aligned} \epsilon &= E[(\mathbf{x} - \bar{\mathbf{x}})^2] \\ &= E \left[\left\langle \sum_{i=D}^{N-1} X_i \Phi_i, \sum_{i=D}^{N-1} X_i \Phi_i \right\rangle \right] \end{aligned} \quad (1.12e)$$

となる。ここで、基底関数を正規直交性の下に仮定すると、

$$\langle \Phi_i, \Phi_k \rangle = \delta_{ik} \quad (1.12f)$$

となり、 \mathbf{x} は、実ベクトルとなり、

$$\begin{aligned} \varepsilon &= E \left[\sum_{i=D}^{N-1} |X_i|^2 \right] \\ &= E \left[\sum_{i=D}^{N-1} | \langle \mathbf{x}, \Phi_i \rangle |^2 \right] \end{aligned} \tag{1.12g}$$

$$\begin{aligned} \varepsilon &= E \left[\sum_{i=D}^{N-1} \Phi_i^T \mathbf{x} \mathbf{x}^T \Phi_i \right] \\ &= \sum_{i=D}^{N-1} \Phi_i^T E[\mathbf{x} \mathbf{x}^T] \Phi_i \end{aligned} \tag{1.12h}$$

となる。 T は転置であり、 $[A] = E[\mathbf{x} \mathbf{x}^T]$ のようなランダムベクトル \mathbf{x} の自己共分散行列を定義すると、正規直交条件から、

$$\left(\frac{\delta}{\delta \Phi_i} \right) \{ \varepsilon - \mu_i \langle \Phi_i, \Phi_i \rangle \} = 0 \tag{1.12i}$$

となり,

$$([A] - \mu_i [I_N]) \Phi_i = 0, \quad i = 0, 1, \dots, N-1 \quad (1.12j)$$

と変形できる. ここで μ_i は, 束縛条件のために導入したラグランジュ乗算子で $[I_N]$ は, 単位行列である. また, $[A]$ は準正定値である. MSE の最小化は, この固有値問題に帰結し,

$$\begin{aligned} [\phi] &= [\Phi_0, \Phi_1, \dots, \Phi_{N-1}] \\ [\phi]^{-1} [A] [\phi] &= \text{diag}[\mu_0, \mu_1, \dots, \mu_{N-1}] \end{aligned} \quad (1.12k)$$

を解くことになる。この結果、打切りによって得られる MSE は、

$$\epsilon = \sum_{i=D}^{N-1} \mu_i \quad (1.12l)$$

となる。そして、式 (1.12j) の対角化問題として存在し、基底関数が自己共分散行列 $[A]$ に依存し、あらかじめ決定できないが、以下の形の際に解析的な解が得られる。

$$[A]_{ik} = \rho^{|i-k|} \quad i, k = 0, 1, \dots, N-1 \quad (1.12m)$$

ここで $0 < \rho < 1$, ρ : 隣接自己相関関数

DCT は、現在知られている KLT に最も近い特性をもつ極めて実用的な変換である。画像符号化を例にとると、 8×8 の正画面素ブロック単位に 64 個の画素について DCT 演算を実行する。DCT の系統は、以下の式 (1.13)~(1.18) に示す 4 系統がある。

$$\begin{aligned}
 [C_{N+1}^I] &= \sqrt{\frac{2}{N}} \left[K_m K_n \cos \left(\frac{mn\pi}{N} \right) \right], \quad m, n = 0, 1, \dots, N \\
 [C_N^{II}] &= \sqrt{\frac{2}{N}} \left[K_m \cos \left(m \left(n + \frac{1}{2} \right) \frac{\pi}{N} \right) \right], \quad m, n = 0, 1, \dots, N-1 \\
 [C_N^{III}] &= \sqrt{\frac{2}{N}} \left[K_n \cos \left(\left(m + \frac{1}{2} \right) \frac{n\pi}{N} \right) \right], \quad m, n = 0, 1, \dots, N-1 \\
 [C_N^{IV}] &= \sqrt{\frac{2}{N}} \left[\cos \left(\left(m + \frac{1}{2} \right) \left(n + \frac{1}{2} \right) \frac{\pi}{N} \right) \right], \quad m, n = 0, 1, \dots, N-1
 \end{aligned} \tag{1.13}$$

ここで、

$$K_j = \begin{cases} 1, & j \neq \text{or } N \\ 1/\sqrt{2}, & j = \text{or } N \end{cases}$$

である。DCT の同一性は次のように示される。

$$\begin{aligned}
 \left[C_{N+1}^I \right]^{-1} &= \left[C_{N+1}^I \right], \text{ 順変換=逆変換} \\
 \left[C_N^{II} \right]^{-1} &= \left[C_N^{III} \right] = \left[C_N^{II} \right]^{-T} \\
 \left[C_N^{III} \right]^{-1} &= \left[C_N^{II} \right] = \left[C_N^{III} \right]^{-T} \\
 \left[C_N^{IV} \right]^{-1} &= \left[C_N^{IV} \right], \text{ 順変換=逆変換}
 \end{aligned} \tag{1.14}$$

DCT の順変換および逆変換の系列は以下のように定義できる。

DCT-I

$$X^{c1}(m) = \sqrt{\frac{2}{N}} K_m \sum_{n=0}^N K_n x(n) \left[\cos \left(\frac{mn\pi}{N} \right) \right], \quad (1.15)$$

$$m = 0, 1, \dots, N$$

IDCT-I

$$x(n) = \sqrt{\frac{2}{N}} K_n \sum_{m=0}^N K_m X^{c1}(m) \cos \left(\frac{mn\pi}{N} \right),$$

$$m = 0, 1, \dots, N$$

DCT-II

$$X^{c2}(m) = \sqrt{\frac{2}{N}} K_m \sum_{n=0}^{N-1} x(n) \cos \left[\frac{m(2n+1)\pi}{2N} \right], \quad (1.16)$$

$$m = 0, 1, \dots, N-1$$

IDCT-II

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} K_m X^{c2}(m) \cos \left[\frac{m(2n+1)\pi}{2N} \right],$$

$$m = 0, 1, \dots, N-1$$

DCT-III

$$X^{c3}(m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} K_n x(n) \cos \left[\frac{n(2m+1)\pi}{2N} \right], \quad (1.17)$$

$$m = 0, 1, \dots, N-1$$

IDCT-III

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} K_n X^{c3}(m) \cos \left[\frac{n(2m+1)\pi}{2N} \right],$$

$$m = 0, 1, \dots, N-1$$

DCT-IV

$$X^{c4}(m) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)(2m+1)\pi}{4N} \right], \quad (1.18)$$

$$m = 0, 1, \dots, N-1$$

IDCT-IV

$$x(n) = \sqrt{\frac{2}{N}} \sum_{m=0}^{N-1} X^{cd}(m) \cos \left[\frac{(2m+1)(2n+1)\pi}{4N} \right],$$

$$m = 0, 1, \dots, N-1$$

この中で、最も多用されているのが DCT-II である。これは、1974 年に K.R.Rao らによって開発されたもので、MPEG など多くの国際標準で用いられている。I から IV の型は、いずれも分解可能で高速アルゴリズムが見つかっている。ここで、この DCT-II を前提に議論を進めると、実際には多点 DCT 演算が用いられる。

N 点 DCT は、式 (1.19) で表される。

$$X^{c2}(k) = \frac{2}{N} c_k \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)k\pi}{2N} \right], \quad (1.19)$$

$$k = 0, 1, \dots, N-1$$

また、 N 点 IDCT は、式 (1.20) で表される。

$$x(n) = \sum_{k=0}^{N-1} c_k X^{c2}(k) \cos \left[\frac{(2n+1)k\pi}{2N} \right], \quad (1.20)$$

$$n = 0, 1, \dots, N-1$$

ここで、

$$c_k = \begin{cases} 1/\sqrt{2}, & k = 0 \\ 1, & k \neq 0 \end{cases}$$

である。

正規化 N 点 DCT は、式 (1.21) で表される。

$$X^{c2}(k) = \sqrt{\frac{2}{N}} c_k \sum_{n=0}^{N-1} x(n) \cos \left[\frac{(2n+1)k\pi}{2N} \right], \quad (1.21)$$

$$k = 0, 1, \dots, N-1$$

さらに正規化 N 点 IDCT は、式 (1.22) で表される。

$$x(n) = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} c_k X^{c2}(k) \cos \left[\frac{(2n+1)k\pi}{2N} \right], \quad (1.22)$$

$$n = 0, 1, \dots, N-1$$

ここで、式 (1.19) と (1.20) における $k=0$ の係数は、式 (1.23) で表される。

$$\begin{aligned}
 X^{(2)}(0) &= \frac{2}{N} \left(\frac{1}{\sqrt{2}} \right) \sum_{n=0}^{N-1} x(n) \\
 &= \frac{\sqrt{2}}{N} \sum_{n=0}^{N-1} x(n) \quad \begin{array}{l} \text{直流 (dc) 係数,} \\ \text{入行系列の平均値} \end{array}
 \end{aligned} \tag{1.23}$$

$$X^{(2)}(k) = \text{交流 (ac) 係数}, \quad k = 0, 1, \dots, N-1$$

k が増加すると $X^{(2)}(k)$ は増加する周波数を表す。

$x(n), k = 0, 1, \dots, N-1$, データ系列 (時間または空間領域で均等に標本化された)

$X^{(2)}(k), k = 0, 1, \dots, N-1$, DCT 系列

次に DCT と IDCT は、ベクトル行列形式では式 (1.24) と (1.25) のように表される。

$$\begin{array}{c}
 X^{c2}(k) \\
 \left[\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \end{array} \right] \\
 (N \times 1) \\
 \left(\begin{array}{c} \text{実数} \\ \text{ベクトル} \end{array} \right)
 \end{array}
 =
 \left(\frac{2}{N} \right)
 \begin{array}{c}
 n \rightarrow \\
 \downarrow k \\
 \left[\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \end{array} \right] \\
 (N \times N) \\
 \text{(DCT 行列)}
 \end{array}
 \begin{array}{c}
 x(n) \\
 \left[\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \end{array} \right] \\
 (N \times 1) \\
 \left(\begin{array}{c} \text{実数} \\ \text{ベクトル} \end{array} \right)
 \end{array}
 \quad (1.24)$$

$$c_k = \begin{cases} 1/\sqrt{2}, & k = 0 \\ 1, & k \neq 0 \end{cases}$$

$$\begin{aligned}
 & \begin{matrix} x(n) \\ \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \end{bmatrix} \\ (N \times 1) \\ \left(\begin{matrix} \text{データ} \\ \text{ベクトル} \end{matrix} \right) \end{matrix} = \begin{matrix} k \rightarrow \\ \begin{matrix} 0 & 1 & 2 & 3 & \dots & N-1 \end{matrix} \\ \begin{bmatrix} c_k \cos \left[\frac{(2n+1)k\pi}{2N} \right] \\ \vdots \\ \vdots \end{bmatrix} \\ (N \times N) \\ \text{(IDCT 行列)} \end{matrix} \begin{matrix} X^{e2}(k) \\ \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \end{bmatrix} \\ (N \times 1) \\ \left(\begin{matrix} \text{変換} \\ \text{ベクトル} \end{matrix} \right) \end{matrix} \quad (1.25)
 \end{aligned}$$

ここで DCT は直交行列であるため式 (1.26) が成立する。

$$\begin{aligned}
 & \frac{2}{N} [\text{DCT 行列}] [\text{IDCT 行列}] = I_N \\
 & \quad \quad \quad (N \times N) \\
 & [\text{DCT}] [\text{IDCT}]^{-1} = I_N \\
 & \therefore [\text{DCT}]^{-1} = \frac{2}{N} [\text{IDCT}] \quad (1.26) \\
 & \quad \quad \quad = \frac{2}{N} [\text{DCT}]^T \\
 & [\text{IDCT}] = [\text{DCT}]^T \\
 & \frac{2}{N} [\text{DCT}] [\text{DCT}]^T = I_N
 \end{aligned}$$

また正規化 DCT の基底ベクトルは、式 (1.27) で表される、

$$\begin{aligned}
 c(k, n) &= 1/\sqrt{N}, \text{ for } k = 0, 0 \leq n \leq N - 1 \\
 c(k, n) &= \sqrt{\frac{2}{N}} \cos \left[\frac{(2n + 1)k\pi}{2N} \right], \text{ for } \begin{matrix} 0 \leq n \leq N - 1 \\ 1 \leq k \leq N - 1 \end{matrix} \quad (1.27) \\
 &= \sqrt{\frac{2}{N}} C_{2N}^{k(2n+1)}, \text{ ここで } C_a^b = \cos \left(\frac{b\pi}{a} \right)
 \end{aligned}$$

さらに正規化 DCT 行列と正規化 IDCT 行列は、各々式 (1.28) と (1.29) とで表される、

$$\begin{array}{c}
 \text{行} \downarrow \\
 \begin{array}{c}
 0 \\
 1 \\
 2 \\
 3 \\
 \vdots \\
 N-1
 \end{array}
 \end{array}
 \begin{array}{c}
 \text{列} \rightarrow n \\
 \begin{array}{c}
 0 \quad 1 \quad 2 \quad 3 \quad \dots \quad N-1 \\
 \left[\begin{array}{c}
 (1/\sqrt{N})(1/\sqrt{N}) \dots (1/\sqrt{N}) \\
 \sqrt{\frac{2}{N}} \cos \left[\frac{(2n+1)k\pi}{2N} \right] \\
 0 \leq n \leq N-1 \\
 1 \leq k \leq N-1
 \end{array} \right]
 \end{array}
 \end{array}
 \quad (1.28)$$

$$\begin{array}{c}
 \text{行} \downarrow \\
 \begin{array}{c}
 0 \\
 1 \\
 2 \\
 3 \\
 \vdots \\
 N-1
 \end{array}
 \end{array}
 \begin{array}{c}
 \text{列} \rightarrow k \\
 \begin{array}{c}
 0 \quad 1 \quad 2 \quad 3 \quad \dots \quad N-1 \\
 \left[\begin{array}{c}
 1 \\
 \sqrt{N} \\
 1 \\
 \sqrt{N} \\
 \vdots \\
 1 \\
 \sqrt{N}
 \end{array} \right]
 \sqrt{\frac{2}{N}} \cos \left[\frac{(2n+1)k\pi}{2N} \right] \\
 0 \leq n \leq N-1 \\
 1 \leq k \leq N-1
 \end{array}
 \end{array}
 \quad (1.29)$$

ここで DCT は三角対角行列 $[Q_c]$ を対角化するため式 (1.30) と (1.31) が成り立つ。

$$\begin{matrix} [\hat{C}_N^{\text{II}}]^T & [Q_c] & [\hat{C}_N^{\text{II}}] & = & \Lambda(\text{対角行列}) \\ (N \times N) & (N \times N) & (N \times N) & & \end{matrix} \quad (1.30)$$

$$\begin{matrix} Q_c & = & \begin{bmatrix} 1 - \alpha & -\alpha & 0 & 0 & 0 \\ -\alpha & 1 & -\alpha & 0 & 0 \\ 0 & -\alpha & 1 & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & -\alpha \\ 0 & 0 & 0 & -\alpha & 1 - \alpha \end{bmatrix} \\ (N \times N) & & \end{matrix} \quad (1.31)$$

ところで一次マルコフ過程 R は、 ρ を隣接相関関数として式 (1.32) と (1.33) とで表される。

$$\begin{aligned}
 R &= \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{bmatrix} \\
 (N \times N) & \\
 &= \text{相関係数行列}
 \end{aligned} \tag{1.32}$$

$$\begin{aligned}
 \beta^2 R^{-1} &= \begin{bmatrix} 1 - \rho\alpha & -\alpha & 0 & 0 & 0 \\ -\alpha & 1 & -\alpha & 0 & 0 \\ 0 & -\alpha & 1 & \ddots & 0 \\ 0 & 0 & \ddots & \ddots & -\alpha \\ 0 & 0 & 0 & -\alpha & 1 - \rho\alpha \end{bmatrix} \\
 (N \times N) & \\
 & \tag{1.33}
 \end{aligned}$$

ここで $\beta^2 = (1 - \rho^2)/(1 + \rho^2)$, $\alpha = \rho/(1 + \rho^2)$.

$\rho \cong 1$, $(1 - \rho\alpha) \cong (1 - \alpha)$ のとき, $\beta^2 R^{-1} \cong \begin{matrix} Q_c \\ (N \times N) \end{matrix}$ となる.

次に DCT が直交行列であることを示すのに DCT (式 (1.19)) に IDCT (式 (1.20)) を代入すると次式となる。

$$\begin{aligned}
 x_n &= \frac{2}{N} \sum_{k=0}^{N-1} c_k^2 \sum_{m=0}^{N-1} x_m \cos \left[\frac{(2m+1)k\pi}{2N} \right] \cos \left[\frac{(2n+1)k\pi}{2N} \right] \\
 &= \frac{2}{N} \sum_{m=0}^{N-1} x_m \sum_{k=0}^{N-1} c_k^2 \cos \theta_m \cos \theta_n
 \end{aligned} \tag{1.34}$$

ただし、

$$\begin{aligned}
 \theta_m &= \frac{(2m+1)k\pi}{2N}, \quad \theta_n = \frac{(2n+1)k\pi}{2N} \\
 \cos \theta_m \cos \theta_n &= \frac{1}{2} [\cos(\theta_m + \theta_n) + \cos(\theta_m - \theta_n)] \\
 &= \frac{1}{2} \left[\frac{e^{j(\theta_m + \theta_n)} + e^{-j(\theta_m + \theta_n)}}{2} \right] + \frac{1}{2} \left[\frac{e^{j(\theta_m - \theta_n)} + e^{-j(\theta_m - \theta_n)}}{2} \right]
 \end{aligned}$$

ここで $\sqrt{-1} = j$ とおくと式 (1.35) となる.

$$\begin{aligned}
 x_n &= \frac{1}{2N} \sum_{m=0}^{N-1} x_m \sum_{k=0}^{N-1} c_k^2 \left(\exp \left[j \frac{2k\pi}{2N} (m+n+1) \right] + \exp \left[-j \frac{2k\pi}{2N} (m+n+1) \right] \right. \\
 &\quad \left. + \exp \left[j \frac{2k\pi}{2N} (m-n) \right] + \exp \left[-j \frac{2k\pi}{2N} (m-n) \right] \right) \\
 c_k &= \begin{cases} 1, & k \neq 0 \\ 1/\sqrt{2}, & k = 0 \end{cases} \\
 &= \frac{1}{2N} \sum_{m=0}^{N-1} x_m \sum_{k=0}^{N-1} c_k^2 \left(W_N^{-(m+n+1)\frac{k}{N}} + W_N^{(m+n+1)\frac{k}{N}} \right. \\
 &\quad \left. + W_N^{-(m-n)\frac{k}{N}} + W_N^{(m-n)\frac{k}{N}} \right) \tag{1.35}
 \end{aligned}$$

ただし, $W_N = \exp \left(\frac{-j2\pi}{N} \right) =$ ユニタリ値の N 乗根. である.

この式はさらに次式のように変形可能である。

$$\begin{aligned}
 & m = n \text{ and } k \neq 0 \text{ のとき } \left(\frac{1}{2N} x_n 2N \right) = x_n \\
 & m = n \text{ and } k \neq 0 \text{ のとき } \left(\frac{1}{2N} x_n \left(\frac{1}{\sqrt{2}} \right)^2 4N \right) = x_n \quad (1.36)
 \end{aligned}$$

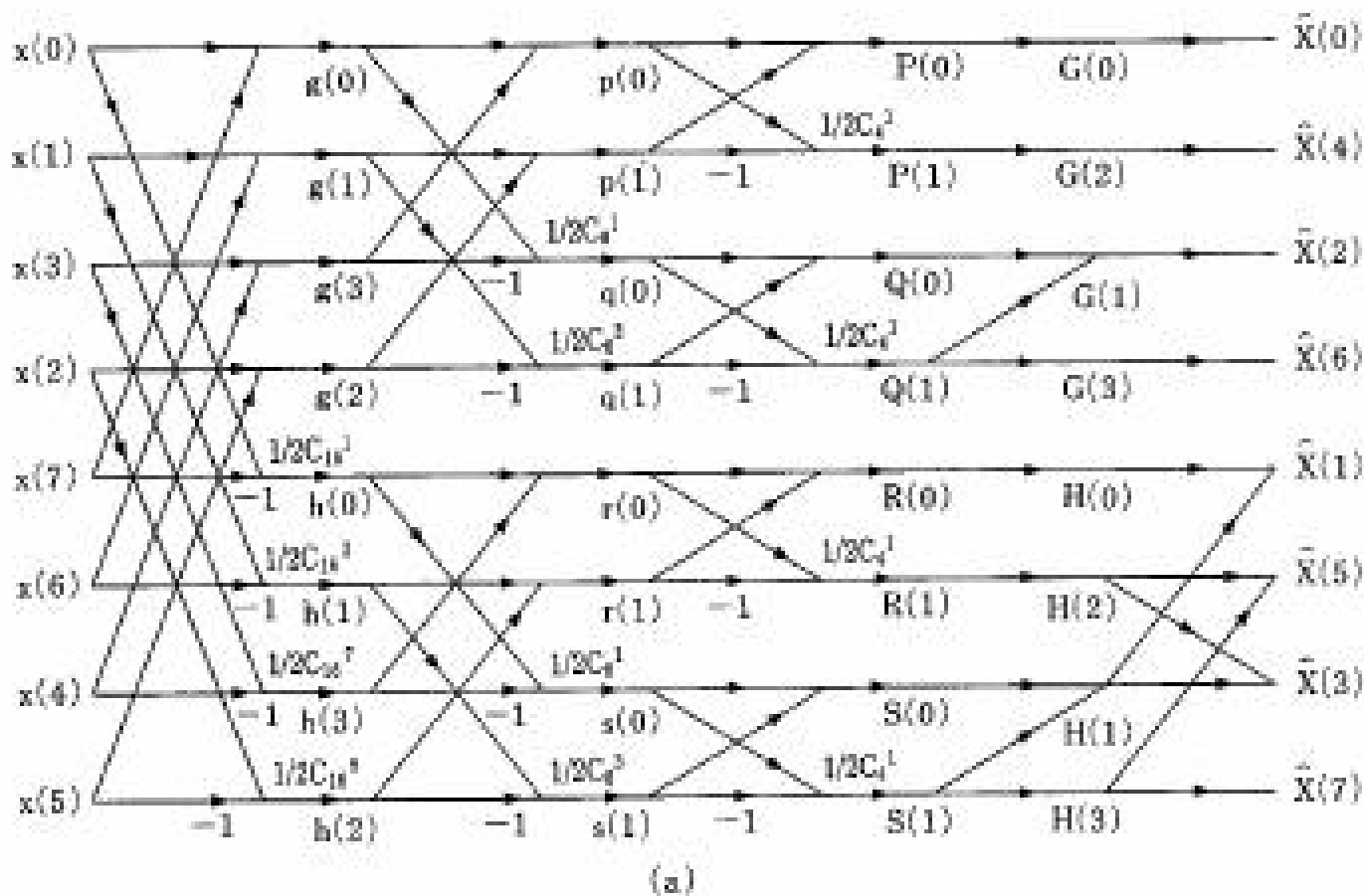
$$\text{ただし } \sum_{k=0}^{N-1} W_N^{kl} = N\delta(l), \text{ ここで } \delta(l) = \begin{cases} 0, & l \neq 0 \\ 1, & l = 0 \end{cases}$$

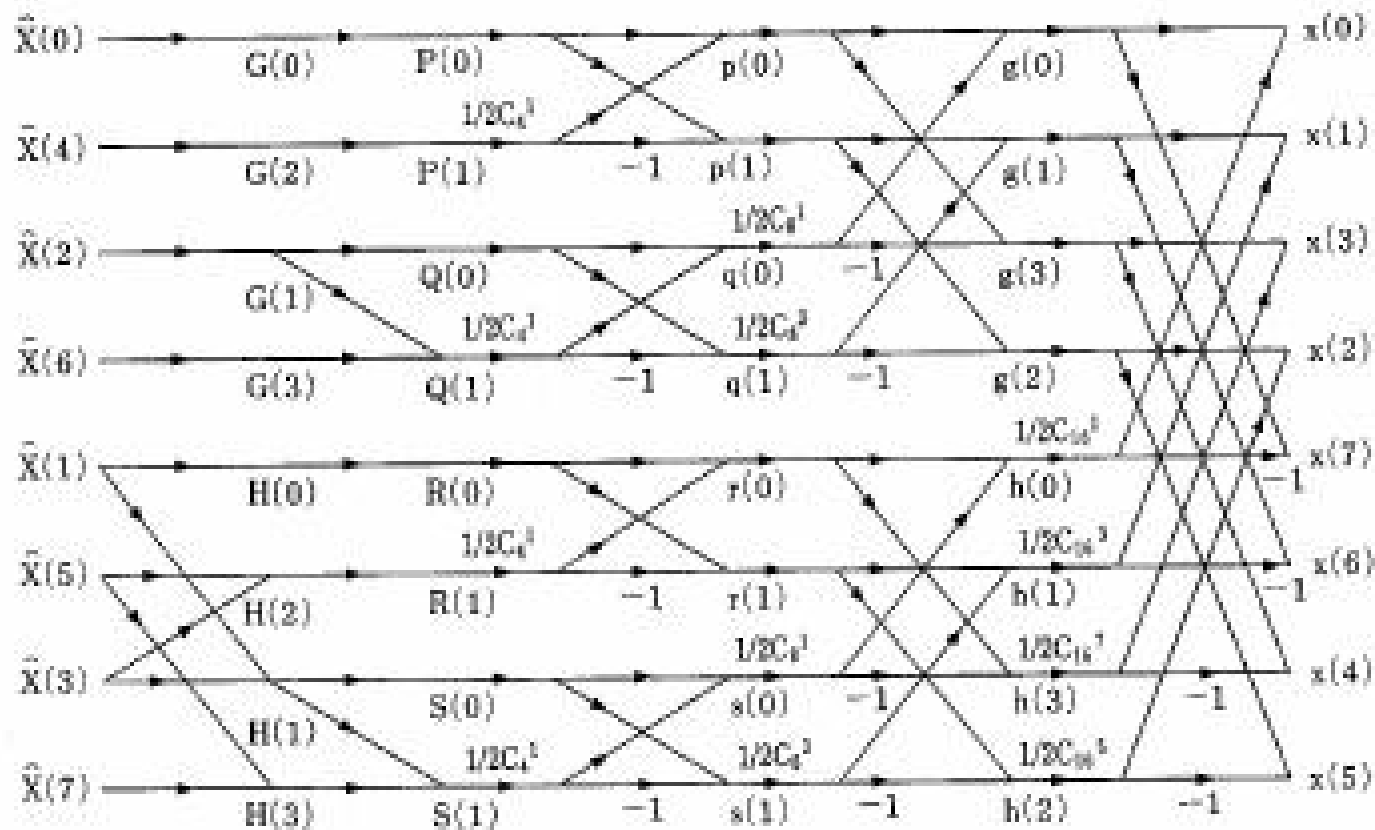
(2) DCT の高速演算アルゴリズム

ところで基本式 (1.19) と (1.20) によれば、 N 点 DCT と IDCT には各々 $N \times N$ 回の乗算と加算が必要となるため、非効率的である。これは、DCT の最大のメリットであるが、様々な高速演算アルゴリズムが開発されてきた。一例をあげると図 1.11 は、Lee のアルゴリズムと呼ばれるものである。この他にも多くの高速アルゴリズムがあるので、詳細は参考文献を参照されたい。

(3) 多次元 DCT

二次元 DCT と IDCT は、一次元の各式 (1.19) と式 (1.20) とを拡張して 2D-DCT, 2D-IDCT, 正規化 2D-DCT, および正規化 2D-IDCT は、各々式 (1.37) から (1.40) となる。





(b)

図 1.11 Lee のアルゴリズム

2D-DCT

$$X_{u,v}^{c2} = \frac{4}{NM} c_u c_v \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{n,m} \cos \left[\frac{(2n+1)u\pi}{2N} \right] \cos \left[\frac{(2m+1)v\pi}{2M} \right], \quad (1.37)$$

$$u = 0, 1, \dots, N-1, \quad c_l = \begin{cases} 1/\sqrt{2}, & l = 0 \\ 1, & l \neq 0 \end{cases}$$

$$v = 0, 1, \dots, M-1,$$

2D-IDCT

$$x_{n,m} = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} c_u c_v X_{u,v}^{c2} \cos \left[\frac{(2n+1)u\pi}{2N} \right] \cos \left[\frac{(2m+1)v\pi}{2M} \right], \quad (1.38)$$

$$n = 0, 1, \dots, N-1, \quad m = 0, 1, \dots, M-1$$

正規化 2D-DCT

$$\begin{aligned}
 X_{u,v}^{c2} &= c_u c_v \frac{2}{\sqrt{NM}} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} x_{n,m} \cos \left[\frac{(2n+1)u\pi}{2N} \right] \cos \left[\frac{(2m+1)v\pi}{2M} \right] \\
 &= \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} c_u \left[\sqrt{\frac{2}{M}} c_v \sum_{m=0}^{M-1} x_{n,m} \cos \frac{(2m+1)v\pi}{2M} \right] \cos \frac{(2n+1)u\pi}{2N}, \quad (1.39)
 \end{aligned}$$

$$\begin{aligned}
 u &= 0, 1, \dots, N-1, \\
 v &= 0, 1, \dots, M-1,
 \end{aligned}
 \quad c_l = \begin{cases} 1/\sqrt{2}, & l = 0 \\ 1, & l \neq 0 \end{cases}$$

正規化 2D-IDCT

$$X_{n,m} = \frac{2}{\sqrt{NM}} \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} c_u c_v X_{u,v}^{c2} \cos \left[\frac{(2n+1)u\pi}{2N} \right] \cos \left[\frac{(2m+1)v\pi}{2M} \right], \quad (1.40)$$

$$n = 0, 1, \dots, N-1, \quad m = 0, 1, \dots, M-1$$

当然ながら分離可能であるため、行列表現は、各々式 (1.41) と (1.42) となる。

2D-DCT

$$\begin{matrix}
 [X^{c2}] & = & \frac{2}{N} [C_N^{\text{II}}] & [x] & \frac{2}{N} [C_N^{\text{II}}]^T \\
 (N \times N) & & (N \times N)(N \times N) & & (N \times N)
 \end{matrix} \quad (1.41)$$

2D-IDCT

$$\begin{matrix}
 [x] & = & [C_N^{\text{II}}]^T [X^{c2}] [C_N^{\text{II}}] \\
 (N \times N) & & (N \times N)(N \times N)(N \times N)
 \end{matrix}$$

$$\begin{matrix}
 \frac{2}{N} [C_N^{\text{II}}] [C_N^{\text{II}}]^T & = & \frac{2}{N} [C_N^{\text{II}}]^T [C_N^{\text{II}}] \\
 (N \times N)(N \times N) & & (N \times N)(N \times N)
 \end{matrix} \quad (1.42)$$

$$\begin{matrix}
 = \mathbf{I}_N \\
 (N \times N)
 \end{matrix}$$

なお、分離可能性の証明は、式 (1.43) に示すことができる。

$$\begin{aligned}
 X_{u,v}^{c2} &= \frac{2}{N} \sum_{n=0}^{N-1} c_v \left[\frac{2}{M} \sum_{m=0}^{M-1} c_u x_{n,m} \cos \frac{(2m+1)v\pi}{2M} \right] \cos \frac{(2n+1)u\pi}{2N} \\
 &= \frac{2}{M} \sum_{m=0}^{M-1} c_v \left[\frac{2}{N} \sum_{n=0}^{N-1} c_u x_{n,m} \cos \frac{(2n+1)u\pi}{2N} \right] \cos \frac{(2m+1)v\pi}{2M}, \quad (1.43)
 \end{aligned}$$

$$u = 0, 1, \dots, N-1, \quad v = 0, 1, \dots, M-1$$

また、三次元への拡張も容易にでき式 (1.44) と式 (1.45) が得られる。

3D-DCT

$$\begin{aligned}
 X_{u,v,p}^{c2} &= \frac{8}{NML} c_u c_v c_p \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x_{n,m,l} \cos \left[\frac{(2n+1)u\pi}{2N} \right] \\
 &\quad \cos \left[\frac{(2m+1)v\pi}{2M} \right] \cos \left[\frac{(2l+1)p\pi}{2L} \right], \quad (1.44)
 \end{aligned}$$

$$\begin{aligned}
 &u = 0, 1, \dots, N-1, \\
 &v = 0, 1, \dots, M-1, \\
 &p = 0, 1, \dots, L-1,
 \end{aligned}
 \quad c_k = \begin{cases} 1/\sqrt{2}, & k = 0 \\ 1, & k \neq 0 \end{cases}$$

3D-IDCT

$$X_{n,m,l} = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} \sum_{p=0}^{L-1} X_{u,v,p}^2 c_u c_v c_p \cos \left[\frac{(2n+1)u\pi}{2N} \right] \cos \left[\frac{(2m+1)v\pi}{2M} \right] \cos \left[\frac{(2l+1)p\pi}{2L} \right] \quad (1.45)$$

このようにして得られる DCT の特性を理解するために、図 1.12 に二次元 8×8DCT の基底画像を示す。

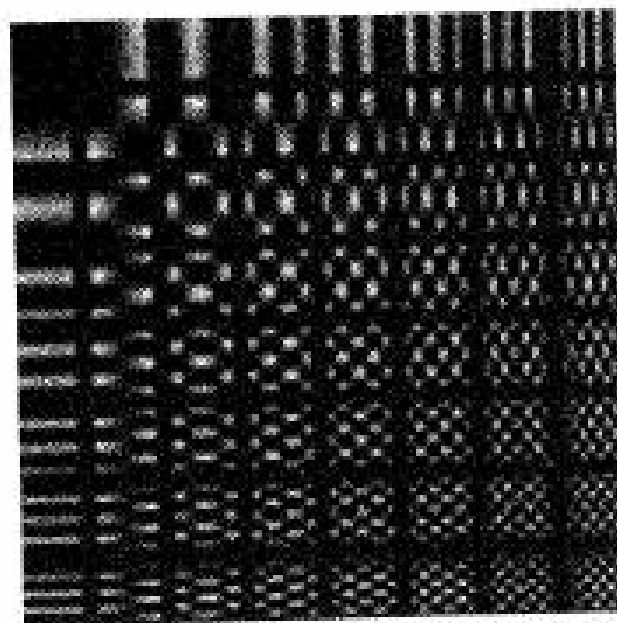


図 1.12 二次元 (8×8) DCT の基底画像

5. 予測符号化 ~時間的圧縮のために~

1.2.4 予測符号化

予測符号化とは、予測値、すなわち予測誤差（差分）を符号化することである。このことから DPCM(Differential Pulse Code Modulation) とも呼ばれる。予測誤差を量子化するメリットは大きい。それは、原信号を量子化するよりかなり少ない量子化レベルで量子化できることである。図 1.13 に DPCM のブロック図を示す。DPCM は、JPEG など多くの情報圧縮方式の要素として利用されている。

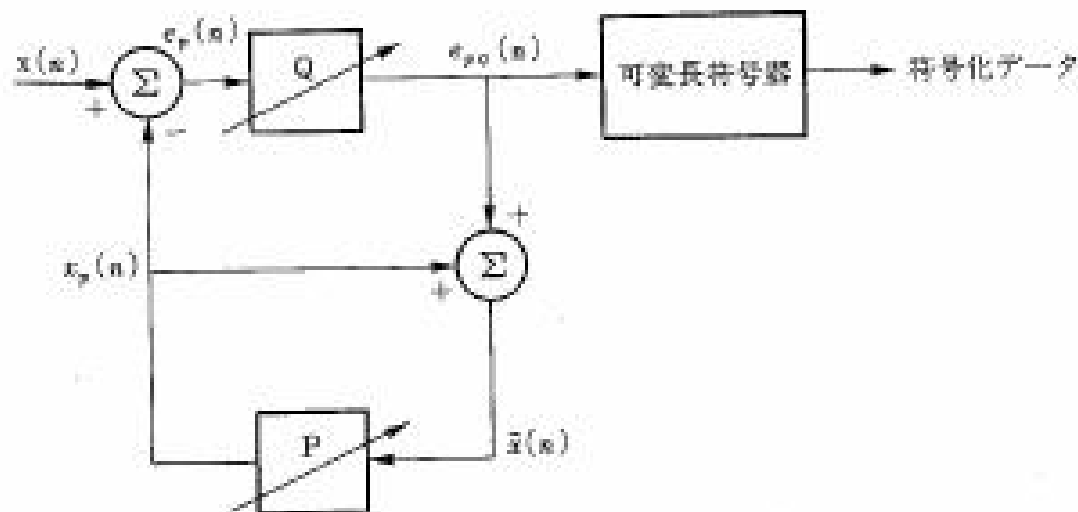


図 1.13 DPCM のブロック図

1.2.7 動き補償

動き補償とは、動画像圧縮だけに有効な手法である。これは、1画面中にある対象物体が異なる時刻にどの位置にあるかを探索し、移動量を求めることで情報量を圧縮する考え方である。アイデアとしては、画素ごとに追跡するPRA (Pel-Recursive Algorithm, 画素漸化型アルゴリズム) と画素ブロックごとに追跡するBMA (Block Matching Algorithm, ブロック・マッチング・アルゴリズム) とが知られているが、PRAは実用的でないためもっぱらBMA (図1.6参照) が用いられている。

図1.16は、現フレームの $M \times N$ 画素ブロックが、前フレーム中の探索範囲 (サーチウィンドウ

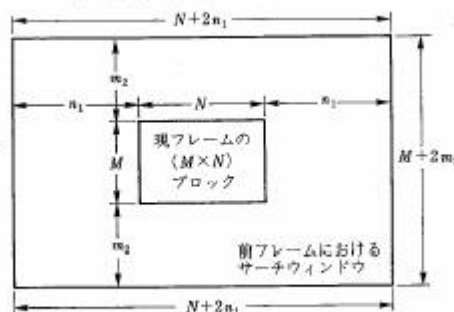


図 1.16 現フレームの $M \times N$ 画素ブロックと前フレーム中の探索範囲の関係

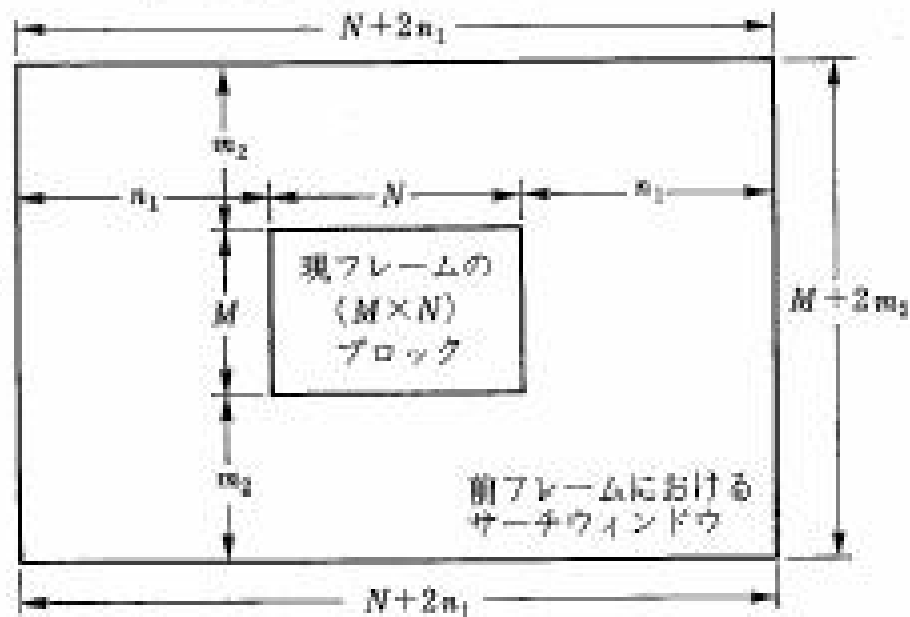


図 1.16 現フレームの $M \times N$ 画素ブロックと前フレーム中の探索範囲の関係

この対象ブロックの画素値と比較ブロックの画素値間の一致度を測る評価測度としては、以下に示す3つが主として用いられている。各々一長一短があるが、実装の容易な MAE が多く用いられている。

① 平均二乗誤差 (MSE)

$$M_1(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N (X_{m,n} - X_{m+i,n+j}^R)^2, \quad (1.52)$$

$$|i| \leq m_2, |j| \leq n_2$$

② 平均絶対値誤差 (MAE)

$$M_2(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |X_{m,n} - X_{m+i,n+j}^R|, \quad (1.53)$$

$$|i| \leq m_2, |j| \leq n_2$$

③ 相互相関関数

$$M_3(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N X_{m,n} X_{m+i,n+j}^R}{\left[\sum_{m=1}^M \sum_{n=1}^N X_{m,n}^2 \right]^{1/2} \left[\sum_{m=1}^M \sum_{n=1}^N (X_{m+i,n+j}^R)^2 \right]^{1/2}}, \quad (1.54)$$

$$|i| \leq m_2, |j| \leq n_2$$

さて、式 (1.52) から (1.54) を全探索領域にわたって演算するには膨大な演算量となる。例えば、 16×16 画素を X, Y 軸ともに ± 8 の範囲で探索すると、1つの動きベクトルを得るのに 256 回の演算が必要になってしまう。このため、ソフトウェア処理のために多くの簡略化演算手法が提案されている。例をあげると、対数探索、共役方向探索、三段探索、および階層探索などが知られている。

6. その他の圧縮手法 ~VQとサブバンド符号化~

1.2.5 ベクトル量子化

ベクトル量子化は、体系的理論として整理されたのは、1980年のLinde, Buzo, Grayによる論文が最初である。また、VQ(Vector Quantization)とも略されて親しまれてきた情報圧縮手法で、MPEGなどの国際標準においては、DCTに主役の座を渡した感があるが、極めてシンプルな考え方でDCTに匹敵する有効な方式である。

これは、図1.14に示したように入力値の集合を入力ベクトルとして定義し、あらかじめ作成しておいたエンコーダ・コードブックとパターンマッチング処理を行い、一致度の最も高いコードをインデックスとして送るものである。インデックスを受信したデコーダでは、テーブル検索を行って出力ベクトルを得るようになっている。

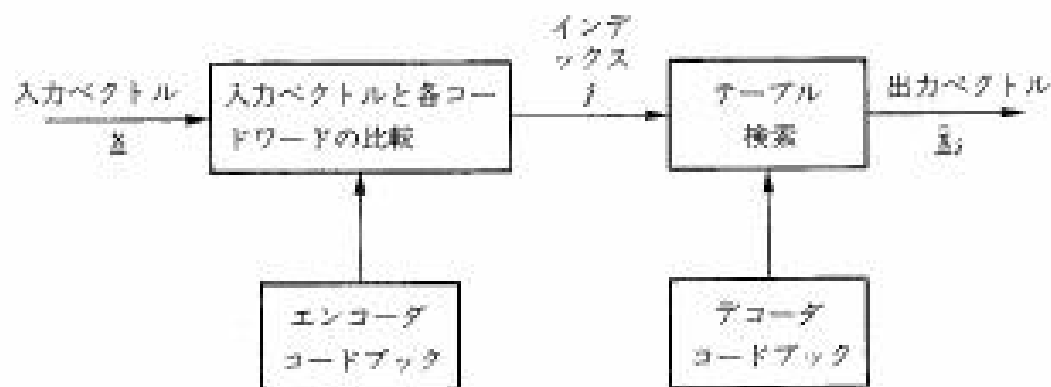


図 1.14 ベクトル量子化の処理ブロック図

図 1.14 において重要なのは、エンコーダ・コードブックから得られたベクトル値と実際の入力値との間の歪みを最小に抑えることである。この歪みの測度としては、主として以下の 6 つが知られている。

① 平均二乗誤差 (MSE)

$$d_1(\underline{x}, \hat{\underline{x}}_i) = \frac{1}{K} \sum_{m=1}^K [x(m) - \hat{x}_i(m)]^2 \quad (1.46)$$

② 平均絶対値誤差 (MAE)

$$d_2(\underline{x}, \hat{\underline{x}}_i) = \frac{1}{K} \sum_{m=1}^K |x(m) - \hat{x}_i(m)| \quad (1.47)$$

③ L_n ノルム

$$d_3(\underline{x}, \hat{\underline{x}}_i) = \left[\sum_{m=1}^K |x(m) - \hat{x}_i(m)|^n \right]^{\frac{1}{n}} \quad (1.48)$$

$$d_4(\underline{x}, \hat{\underline{x}}_i) = [d_3(\underline{x}, \hat{\underline{x}}_i)]^n$$

④ 重み付け歪み

$$d_5(\underline{x}, \hat{\underline{x}}_i) = \sum_{m=1}^K w_m [x(m) - \hat{x}_i(m)]^2 \text{ or } \sum_{m=1}^K w_m |x(m) - \hat{x}_i(m)| \quad (1.49)$$

⑤ 一般二次歪み

$$d_6(\underline{x}, \hat{\underline{x}}_i) = (\underline{x} - \hat{\underline{x}}_i)[W](\underline{x} - \hat{\underline{x}}_i)^T \quad (1.50)$$

⑥ 最大歪み

$$d_7(\underline{x}, \hat{\underline{x}}_i) = \max_m |x(m) - \hat{x}_i(m)| \quad (1.51)$$

ベクトル量子化器の性能は、コードブックの設計にかかっている。前述の歪み測度を効率的に適用することで、これが可能となる。

VQには、様々な実装法が提案されている。一例をあげると、木探索 VQ (Tree-search VQ, 木構造の中から枝へ向かって順次探索する)、階層型 VQ (Hierarchical VQ, $N \times N$ を親のブロックとし内部に $1/2$ ずつ精度をあげていく)、多段 VQ (Multistage VQ, 複数の VQ 処理を縦列接続する) などの他、平均値分離 VQ, ゲイン・シェーブ VQ, 補間 VQ, エントロピー制約 VQ など多くの方式がある。

さらに、予測符号化や変換符号化と組み合わせた複合型 VQ なども提案されている。

1.2.6 サブバンド符号化

サブバンド符号化は、変換符号化のように直交変換による周波数分解ではなく、入力信号の周波数成分をストレートに分割する方式である。この周波数分割には、フィルタバンクを用いる。Part3で詳しく述べる MPEG オーディオでは、このサブバンドの分析/合成処理が行われている。

サブバンド・エンコーダの例を図 1.15 に示す。

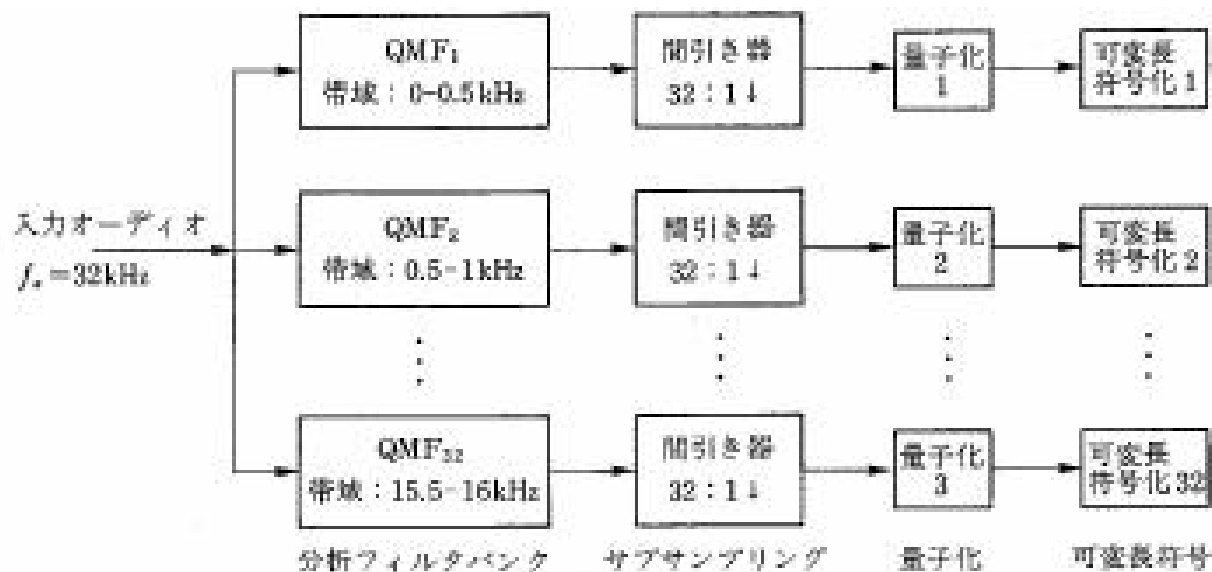


図 1.15 オーディオ・サブバンド・エンコーダの例

7. エントロピー符号化

1.2.8 エントロピー符号化

エントロピー符号化は、これまで述べてきた様々な情報圧縮符号化の最終段階で適用するのが常である。これは、前にも述べたように符号化データの発生確率の偏りを利用した可変長符号化である。例えば、動きベクトルの移動量は、小さい値をとる確率が高い。このため、符号化する際には近傍での

動きベクトルにより短い符号を、遠隔の動きベクトルにはより長い符号を割り当てることで、全体の情報量を抑えることができる。このような発生確率を基本とした可変長符号化をエントロピー符号化と呼んでいる。

実際例として Huffman 符号と二次元 Huffman 符号とについて示す。

(1) Huffman 符号

以下に示すように 4 値をとる例では、固定長 2 進データでは 2 ビット長となるが、可変長 Huffman 符号では 3 ビット長となる。しかし、ビット長の短い 0 の発生確率が極端に高いと統計的なデータ長は、2 ビット以下にすることができるとはならない。

元データ (固定長)	: 10 進数で 0 から 3 とする
2 進データ (固定長)	: 2 進数で 00, 01, 10, 11 とする
Huffman 符号 (可変長)	: 発生確率順位
	1 位: 0 → 0 (1 ビット)
	2 位: 1 → 10 (2 ビット)
	3 位: 2 → 110 (3 ビット)
	4 位: 3 → 111 (3 ビット)

(2) 二次元 Huffman 符号

上記の (1) に加えてランレングス符号化を併用するものである。

この場合は、ゼロラン（連続するゼロの個数）とその次に現れる値のマトリクスに対して Huffman 符号を割り当てるため、二次元 Huffman 符号と呼ばれる。例を表 1.3 に示す。

表 1.3 二次元 Huffman 符号

元データ値	0	1	2	3
ゼロラン	0	00	110	1110
	1	010	1010	01111
	2	0110	01110	111101
	3	110	1011	111100

10 進元データ : 0 2 0 1 0 0 0 1 1 0 0 0 0 3

2 進元データ : 00 10 00 01 00 00 00 01 01 00 00 00 00 11 = 28 ビット

Huffman 符号 : 0 110 0 10 0 0 0 10 10 0 0 0 0 111 = 21 ビット

二次元 Huffman : 1010 010 1011 00 110 1110 = 20 ビット

以上の例においてわかるように 2 進元データに対して Huffman 符号と 二次元 Huffman 符号で符号化したデータは、全体のデータ長を圧縮することができる。

(3) Huffman 符号と二次元 Huffman 符号で符号化したデータの生成法

以下のような手順で符号化する。

- ① Huffman 符号の場合は、符号化対象とする元データの、また二次元 Huffman 符号の場合は、マトリクスの各値の発生確率を求める。
- ② 対象値の数が奇数の場合は、発生確率の最低のものに 0 か 1 を割り当てる。偶数の場合は、最低の 2 つを組み合わせると 0 と 1 を割り当てる。
- ③ 上記のペアと次に低確率の符号と組み合わせると 0 と 1 を割り当てる。
- ④ 順次組み合わせると符号化の木を生成する。

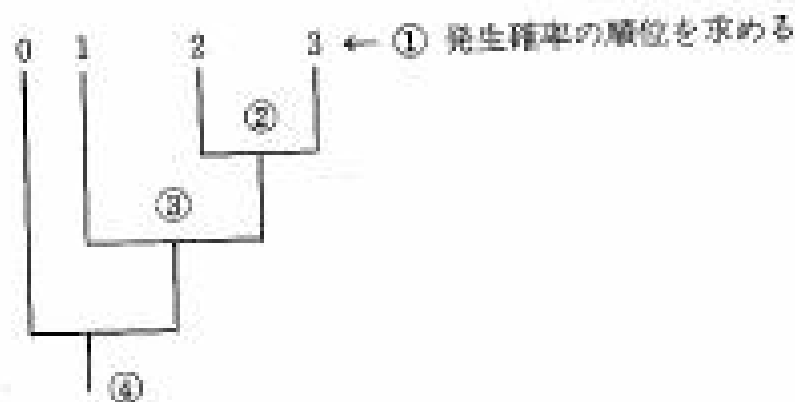


図 1.17 Huffman 符号の生成法

1.3 データ圧縮の基礎

これまでは、画像・音声情報圧縮とインターネットとの関係、および圧縮アルゴリズムの基本原理について解説してきた。読者にはどのような適用条件と原理が用いられているのかが、おおむね理解頂けたと思うが、画像・音声情報圧縮に用いられているのはデジタル信号処理と呼ばれるアナログ信号をデジタル化して冗長成分を削減するものである。これに対して、本節で述べるデータ圧縮は、もともとデジタルデータのファイル容量を削減し、記憶と伝送のコストを低減することを目的としている。したがって、これらの背景にある数学的基礎は全く異なっている。以下にその概要について述べる。

1.3.1 データ圧縮の基本原理

データ圧縮は、これまで述べてきた画像・音声情報圧縮と異なり文字列の圧縮を主な目的としている。大別するとファクシミリやその他の画像・音声情報圧縮にも適用されている Huffman 符号化と、LZ(Lempel-Ziv) 方式との 2 方式が主要なものである。

Huffman 符号化 [1] は、文字の発生確率にバラツキがあることを利用している。このためエントロピー符号化とも呼ばれている。情報科学におけるエントロピー論的には、最適であることが証明済みであるが、前もって文字の発生確率を求める処理が必要となるため、データ列をあらかじめ走査することが要求される。この走査処理後、圧縮処理するという 2 パス方式であることが欠点である。この 2 パスの欠点を解消する方式として、その後、動的 Huffman 符号化方式が発表されている [2]。

これに対して、LZ 方式は、1 パス処理を特徴としている。本方式は、文字列の反復性に着目し、これを冗長成分として除去するものである。過去に記述された文字列と現在の文字列との比較処理を行って符号化する [3]。

表 1.4 にデータ圧縮の基本となる 2 方式の比較を示す。

表 1.4 データ圧縮の基本となる 2 方式の比較

方式	発明	処理過程	圧縮原理	圧縮率
Huffman 符号化	1952 年	2 パス	発生確率	0.7~0.8
LZ(Lempel-Ziv)	1977 年	1 パス	反復性	0.4~0.5

1.3.2 Huffman 符号化

詳細は、すでに 1.2.8 節エントロピー符号化で述べたように符号や文字列の発生確率をまず求めて、それに対応して全体のデータ長を短くする方式である。

1.3.3 LZ(Lempel-Ziv) 方式

LZ(Lempel-Ziv) 方式には大きく分けると 2 つの方式がある。それぞれ発明された年をとって LZ77 と LZ78 と呼ばれている。

(1) LZ77

LZ77 は、当時 Sperry 社に在籍していた Abraham Lempel, Jacob Ziv らが発明し、Stac Electronics 社の保有する特許の基本原理となっている。図 1.18 に示すようにスライド辞書法を用いており、処理時間はかかるが、圧縮率が高くソフトウェアによって実現可能である。LZ77 によって高い圧縮率を実現するには、十分大きなバッファメモリを用意する必要がある。バッファメモリには過去に出現した文字列を記憶し、一致する最長の文字列を求めるものである。このため入力文字列をシフトしていくことになる。

LZ77 を高速化する方法として、Stac 社は Hash 関数を用いている。Hash 関数の適用により、比較対象文字列を短くとることができる。例えば図 1.19 に示すように 2 バイト文字を Hash 関数によって 12 ビットにしたり、3 バイト文字を 1 バイトにしたりすることが可能である。本図においては、C1 と C2 の 2 文字 (2 バイト) を Hash 関数 $[HVAL=F(C1,C2)]$ に入力し、出力である C1 を左 4 ビットシフトし、C2 との排他的論理和をとった値 (12 ビット) としている。

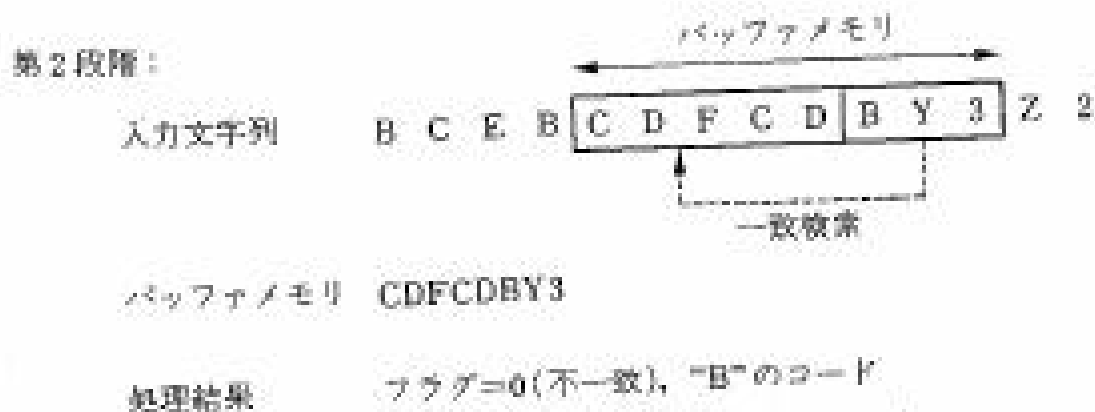
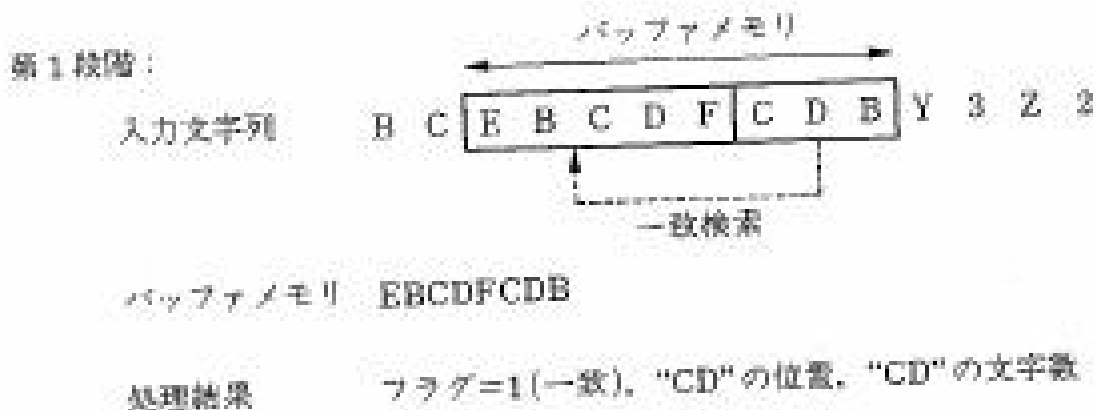


図 1.18 LZ77 の基本となるスライド辞書法



図 1.19 2 バイト文字の Hash 関数による 12 ビット化 [4]

第1段階：入力文字列 ababc

符号化処理結果(1, "b" の2進コード[次の1文字])

辞	文字列	a	b	c	...	z
書	コード	1	2	3	...	26

第2段階：入力文字列 babc

符号化処理結果(2, "a" の2進コード)

辞	文字列	a	b	c	...	z	ab
書	コード	1	2	3	...	26	27

第3段階：入力文字列 abc

符号化処理結果(27, "c" の2進コード)

辞	文字列	a	b	c	...	z	ab	ba
書	コード	1	2	3	...	26	27	28

図 1.20 LZ78 の基本となる動的辞書法

(2) LZ78

LZ78 は、Ferranti,plc 社の John Storer が発明し IBM 社保有の特許の基本原則となっている。図 1.20 に示すように動的辞書法を用いており、処理時間は短いが高圧縮率は低く、ハードウェア処理に向くのが特徴である。LZ78 では過去に出現した文字列に対して専用コードを割り当て、辞書形式で記憶する。

入力文字列と辞書とを比較し、一致した時に辞書コードで表現する。不一致の文字列は辞書に逐次登録する。このように辞書は動的に変化する。当然ながら処理が進むと辞書容量は増大する。

1.3.4 データ圧縮の応用分野

以上に述べたように、データ圧縮の基本原則は、Huffman 符号化と LZ 方式であるが、応用分野としては様々な分野がある。図 1.21 は、Addstor 社による実験例であるが、スプレッドシートファイルやビットマップファイルの圧縮には、極めて有効であることがわかっている。これに対して、実行ファイルの圧縮効率はさほどよくない。

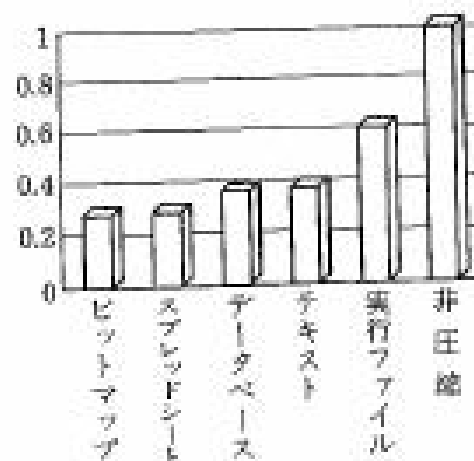


図 1.21 Addstor 社による実験例

現在、インターネット上で多くのデータ交換が行われるようになってきたが、これらは上記の基本原理に基づいて、OS 種別に応じて多様なファイル圧縮形式が用いられている（表 1.5 参照）。

表 1.5 OS 種別に応じて多様なファイル圧縮形式

OS 種別	圧縮形式 (拡張子)
UNIX	compress(Z) 他
Windows	Zip (zip, 32 ビット), LZH (lzh, 16 ビット) 他
Macintosh	Stuffit (sit,sea), Compact Pro (cpt) 他

ところで、インターネットを用いて圧縮ファイルを交換する場合は、一定規則で文字列に変換する必要があるが、これをエンコードと呼ぶことが多い。現在主要なエンコード形式としては、次の3つがある。なお、これらの変換自動化については、MIME(Multipurpose Internet Mail Extensions) というデコード規格によって実現される。

また実際の圧縮・解凍ソフトウェアとしては、フリーウェアでは、DropEx (LZH, ZIP, ARJ, GZ, TAR, ISH 形式の解凍), ChkUNARJ (ARJ, ZIP 形式の解凍と LZH 形式の圧縮・解凍), LHA32.EXE (UNLHA32.DLL を使った LHA 解凍・圧縮), UNARJ32.DLL (ARJ 形式の解凍 DLL), UNLHA32.DLL (LZH 形式の圧縮・解凍 DLL) などが多く用いられている。またシェアウェアとしては、Explzh, Lhasa, Press1 for Windows, WinLm, Winzip などが知られている。

(1) Uuencode

UNIX で用いられている標準形式。

(2) BASE64

Windows で用いられている標準形式。

(3) BinHex

Macintosh で用いられている標準形式。

空間的圧縮=DCT

時間的圧縮=動き補償フレーム間符号化

エントロピー符号化=二次元ハフマン

H.261を元に
MPEG, H.264
へ進化

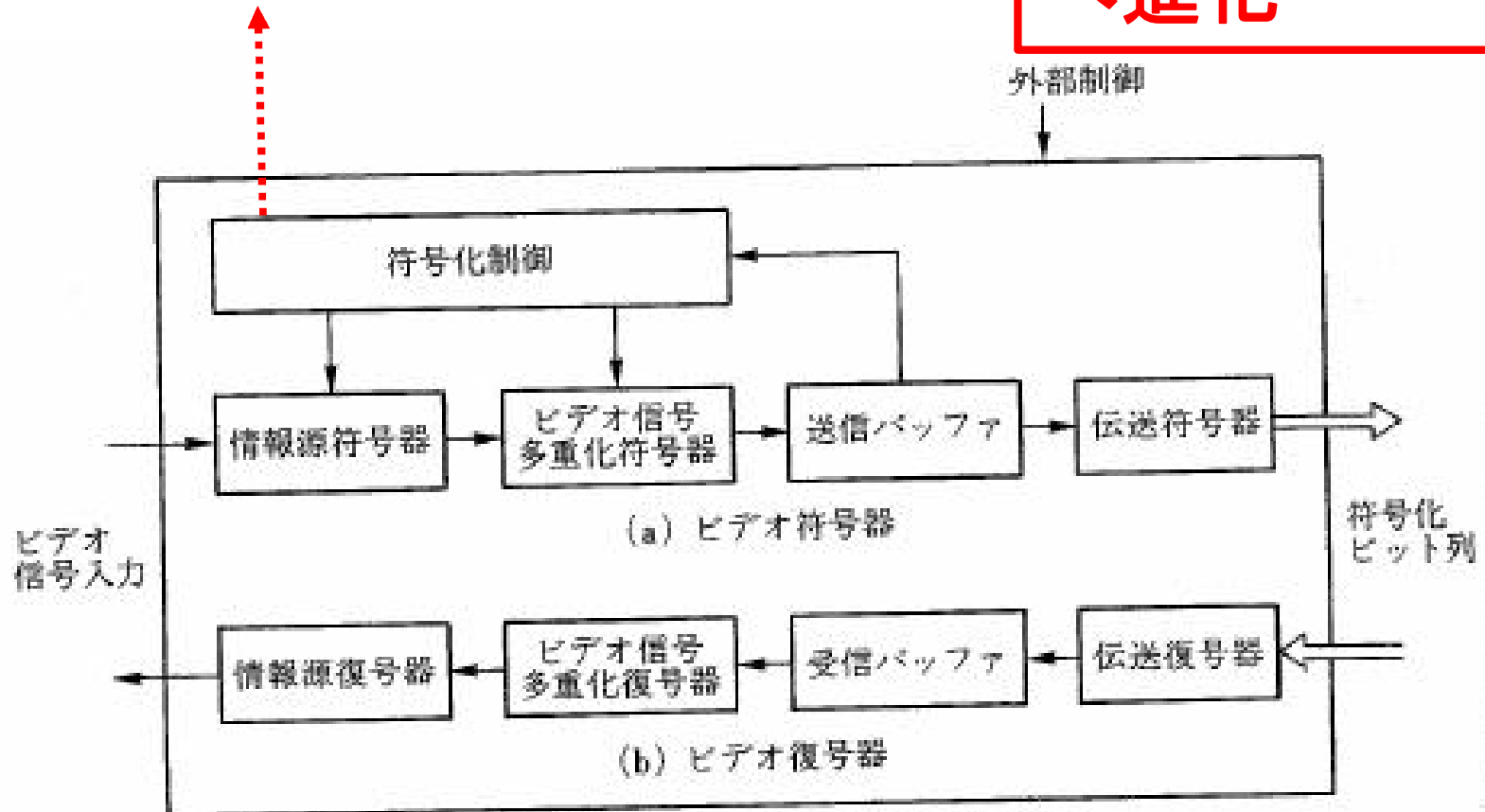


図 2.1 H.261 コーデックのブロック図

ご静聴ありがとうございました