

平成 12 年度
学士学位論文

動画像通信のための情報圧縮技術

**Compression technology for Moving picture
communications**

1010391 合田 仁

指導教員 福本 昌弘

2001 年 2 月 5 日

高知工科大学 情報システム工学科

要 旨

動画像通信のための情報圧縮技術

合田 仁

現在の携帯電話は電子メールなどのインターネットアクセスのサービスを提供できるまでになっており、さらに次世代の携帯電話では音声はもちろん動画像の伝送やその他のマルチメディアサービスを利用することも可能になる。携帯電話などで動画像を伝送する際に行う画像符号化の主な目的は情報量を削減することであり、いかに少ない情報量で画像情報を伝えられるかということが重要となる。

本研究ではテレビ電話のように、背景が変化しないような画像を符号化の対象として低ビットレートでの伝送に適した動画像圧縮を実現する。

キーワード 離散コサイン変換 (DCT) MPEG 符号化アルゴリズム

Abstract

Compression technology for Moving picture communications

Jin Gouda

The present cellular phone can offer service of Internet accesses, such as an email. Furthermore, in the cellular phone of the next generation, it also enables a sound to, use transmission of a motion image, and other multimedia services of course, and the main purposes of the picture coding performed in case a motion image is transmitted with a cellular phone are cutting down the amount of information. It becomes important how image information is told for the few amount of information.

In this research, motion image compression suitable for transmission by the low bit rate made into the object of coding of an image from which a background does not change like a TV phone is realized.

key words Motion compensation, Interframe coding, Discrete Cosine Transform(DCT), MPEG

目次

第 1 章	序論	1
1.1	本研究の背景と目的	1
1.2	本論文の概要	1
第 2 章	動画像圧縮技術	3
2.1	まえがき	3
2.2	変換符号化	5
2.2.1	変換符号化の特徴	6
2.2.2	アダマール変換	6
2.2.3	カルーネン・レーベ変換	7
2.2.4	離散コサイン変換 (DCT)	9
2.3	予測符号化	10
2.3.1	予測符号化の特徴	10
2.3.2	フレーム間予測符号化	10
2.3.3	動き補償符号化	11
2.4	その他の動画像圧縮に用いる技術	11
2.4.1	可変長符号化 (エントロピー符号化)	11
2.4.2	明るさの省略	11
2.4.3	間引き	12
2.5	JPEG について	12
2.6	MPEG について	13
2.6.1	MPEG の符号化アルゴリズム	14
第 3 章	符号化アルゴリズム	18

3.1	はじめに	18
3.2	符号化アルゴリズム	18
3.3	通信方法	22
第 4 章	結論	23
4.1	MPEG との違い	23
4.2	問題点	23
4.3	まとめ	23
	謝辞	24
	参考文献	25
付録 A	MPEG の基本構成	26
付録 B	DCT のプログラム	29

目次

2.1	冗長部分除去による情報圧縮	4
2.2	符号化方式の分類	5
2.3	変換符号化方式の構成図	6
2.4	予測符号化方式の構成図	10
2.5	画像の分解	14
2.6	動き補償・フレーム間予測	15
2.7	2次元離散コサイン変換 (DCT)	16
2.8	エントロピー符号化	16
2.9	MPEG の符号化アルゴリズム	17
3.1	画像の分解	18
3.2	分離	19
3.3	動き補償・フレーム間予測	20
3.4	エントロピー符号化	21
3.5	符号化アルゴリズム	21
A.1	P,B ピクチャ予測構造と I ピクチャ	27
A.2	MPEG 方式の基本構成	28

第 1 章

序論

1.1 本研究の背景と目的

数年前までの携帯電話はただ話すだけしかできなかった。現在の携帯電話はインターネットの閲覧や電子メールの送受信，着信音や静止画像のダウンロードが利用できるようになり，データ通信としての利用が多くなってきている。そして次世代の携帯電話はさらにデータ通信の利用が増えていくものと予想され，その中で特に注目すべき点は動画像通信である。しかし低ビットレートの移動体通信の環境では動画のファイルサイズが大きすぎるため不可能である。そこで，移動体では回線の高速化と同時に圧縮技術が重要になってくる。

また現在，企業の会議などにテレビ電話が利用されているが，この技術を応用し携帯電話で利用できないか考えた。これが実現できれば，どこにいても相手の顔が見れるというメリットが生きてと思われる。本研究ではテレビ電話のように，背景が変化しないような画像を符号化の対象として低ビットレートでの伝送に適した動画像圧縮を実現する。

1.2 本論文の概要

本論文の概要は以下の通りである。

まず第 2 章において，今までに提案されているさまざまな動画像圧縮技術について説明する。また本研究において提案する符号化アルゴリズムと比較するため，MPEG の符号化アルゴリズムについて詳しく説明する。

次に 3 章において，2 章で提案されている技術を使い，テレビ電話のように背景が変化し

ないような画像を符号化の対象とした符号化アルゴリズムを提案する。また低ビットレートでの伝送に適した動画像圧縮を実現する。

最後に、本研究により提案された符号化アルゴリズムと MPEG の符号化アルゴリズムとの比較を行ない、その結果と問題点、将来への展望を第 4 章に示す。

付録として MPEG の基本構成の詳しい説明を加える。

第 2 章

動画像圧縮技術

2.1 まえがき

近年のコンピュータやインターネットの普及はめざましく、ネットワークと情報処理がますます身近なものとなってきた。また、コンピュータは、文字データばかりでなく、画像や音声を扱うこともごく普通になっている。

特に画像信号は音声信号の 1000 倍の情報量を持っている [3]。だからと言って、画像を用いたサービスの料金を電話料金の 1000 倍にしたのでは、高すぎて誰も使用しない [3]。また、限られた伝送路容量では良好な画像のリアルタイム伝送は不可能であり、限られた記憶媒体容量では、長時間分の動画像の記録が困難である。これらの問題点を解決するためには、画像信号の情報量をできるだけ圧縮する必要がある。このための技術として画像符号化がある。

また、どこまで圧縮ができるのかというと、画像信号は、変化の少ない領域では、同じ明るさの信号、あるいは同じ色の信号が連続して発生するケースが多い [3]。これは画像信号が冗長性を持っていることを意味している [3]。このような信号に対しては、すべての信号を伝送するのではなく、たとえば、最初の信号のみを伝送し、受信側ではこの信号を必要なだけ繰り返し使用することにより、伝送する情報量を削減することができる。これは、同じ信号が連続して発生するという統計的性質を利用して、冗長性を除去するものである。冗長性の除去の図を図 2.1 示す。

また、人間の目は通常の明るさでは僅かな変化も識別できるが、暗い所や明るい所では細かな変化は識別しにくいという性質がある [1]。このような視覚特性を利用して、暗い画像

や明るい画像では、視覚特性に応じて少し粗く表現することにより、発生情報量を削減することができる。

このように、画像の持つ統計的性質や視覚特性を利用することにより、情報量を削減できるが、削減しすぎると受信側で良好な画像品質を保てなくなる。

画像符号化の目的は情報量の削減することであるが、いかに少ない情報量で高品質の画像を伝送できるかということが重要である。画像符号化の分類を図 2.2 に示すが、この章ではすでに提案されている変換符号化、予測符号化の技術と効率的な動画像圧縮のためのその他の技術について述べる。

また静止画像の圧縮符号化方式としての JPEG、動画像の圧縮符号化方式としての MPEG は、それぞれ世界レベルでの標準方式として広く普及している。特に MPEG の技術は本研究でも使用するため、後で説明を加えておく。

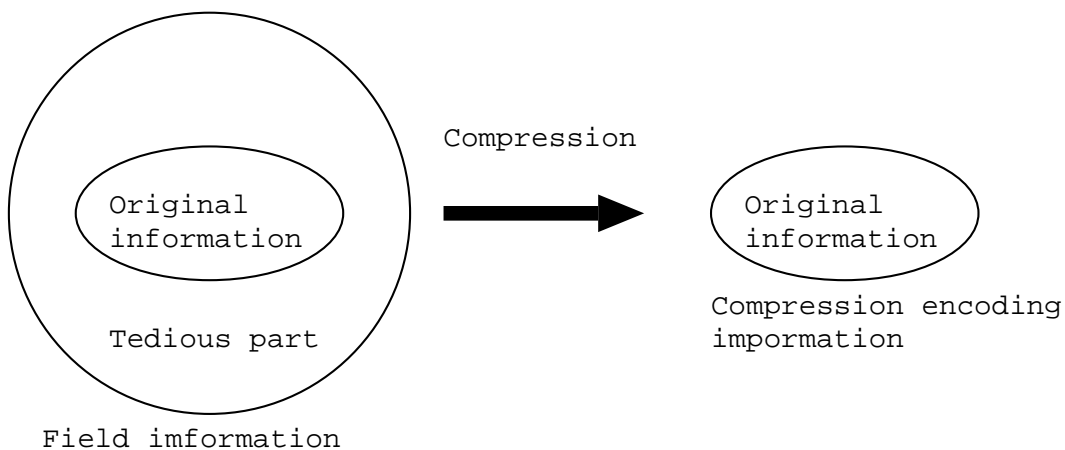


図 2.1 冗長部分除去による情報圧縮

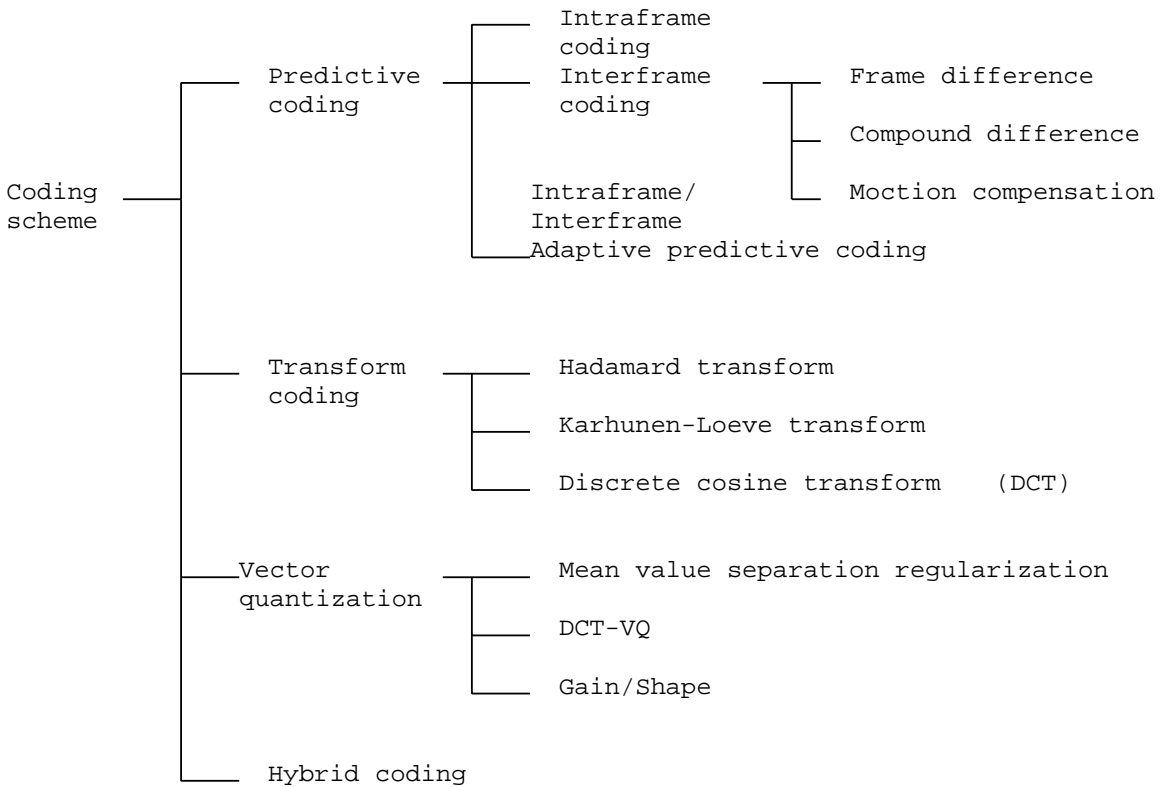


図 2.2 符号化方式の分類

2.2 変換符号化

ここでは、画像圧縮技術の方法の 1 つである変換符号化について説明し、また変換符号化の構成図を図 2.3 に示す。次にそれぞれの変換符号化の概要を述べる。

画像信号を小さな複数個のブロック (たとえば 8 ライン × 8 画素) に分割し、各ブロック毎に直交変換を施し、変換係数を伝送するものである [2]。変換係数領域では、低周波の空間周波数領域にエネルギーが集中する [2]。直交変換の方式により、アダマール変換、カルーネン・レーベ変換、離散コサイン変換 (DCT) 等がある。変換符号化では、分割されたブロック毎に符号化を行なうため、ブロック間で処理法が大きく異なると、ブロックの境界が目に見えるようなブロック状の雑音が生じることがある [2]。

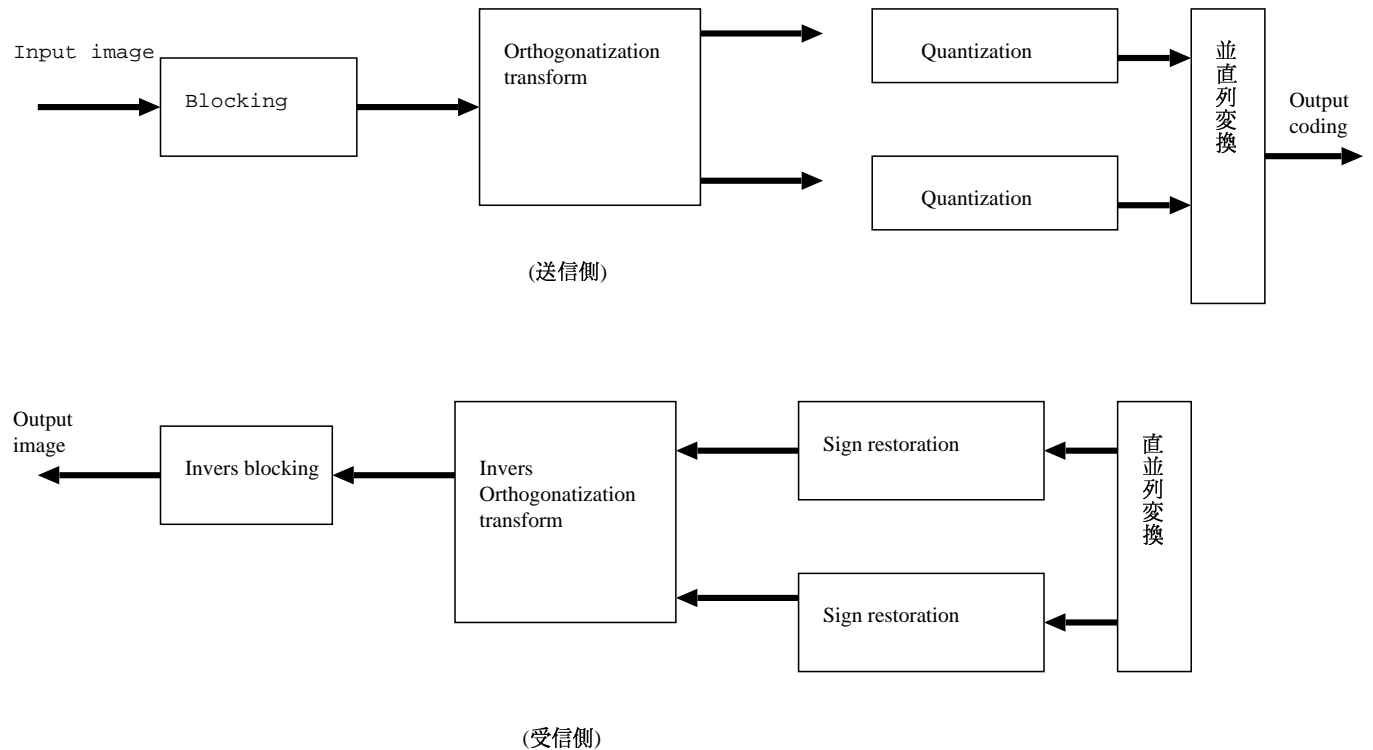


図 2.3 変換符号化方式の構成図

2.2.1 変換符号化の特徴

- 伝送路の誤りに強い
- 高い圧縮率
- 多くの適応処理を行うことができる
- その処理は複雑である

2.2.2 アダマール変換

直交行列にアダマール行列を使用するものが、アダマール変換である [2]。アダマール行列は、1 と -1 のみを要素とし、 N 行 N 列の正方行列で、かつ $1/\sqrt{N}$ で正規化した直交アダマール行列を使用する [2]。画像圧縮に利用されるのは 2 次元アダマール変換であり、これ

は、画像信号の2次元的な相関を利用するもので、 M ライン \times N 画素を単位としてブロック毎に符号化を行なう [2]。この2次元アダマール変換は、

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & \cdots & y_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ y_{M1} & y_{M2} & \cdots & y_{MN} \end{bmatrix} = [H_M] \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ x_{M1} & x_{M2} & \cdots & x_{MN} \end{bmatrix} \quad (2.1)$$

となる。これを一般化すると、次式になる。

$$[y_m n] = [H_M][x_m n][H_N] \quad (2.2)$$

逆変換は次のように表す。

$$[x_m n] = [H_M][y_m n][H_N] \quad (2.3)$$

しかし、アダマール変換では画像信号に関わらず基底ベクトルをあらかじめ決めておき、画像信号がその基底ベクトルの成分を含む割合を変換係数として構成していた。したがって、用意している基底ベクトルの成分を含まないような信号に対しては、良好な変換特性を得ることは困難である [2]。

2.2.3 カルーネン・レーベ変換

カルルーネン・レーベ変換は画像信号の持つ特性に応じた基底ベクトルにより変換しようとするものである [2]。すなわち、 N 画素からなる画像ブロックに対して、変換後の N 個の変換係数の内、 M 個 ($M < N$) で打ち切った時の現画像に対する2乗誤差の期待値を最小とするような直交固有関数を求めるために、原画像の相関関数を用いる [2]。

簡単のため1次元カルルーネン・レーベ変換について説明する。 N 次のカルルーネン・レーベ変換および逆変換をそれぞれ下式のように表す。

$$(y_n) = [K_{NN}](x_n) \quad (2.4)$$

$$(x_n) = [K_{NN}]^t(y_n) \quad (2.5)$$

$[K_{NN}]$ は現画像の相関関数 $[R_{NN}]$ に対する固有ベクトルであり, $[R_{NN}]$ は次式のように定義する.

$$[R_{NN}] = [E(x_{ij} - \bar{x})(x_{ij} - \bar{x})^t] \quad (2.6)$$

この N 次元正方行列である相関関数に関して, 正規化された固有ベクトルを, (K_i) , 固有値 λ_i を用いて次式が成立する.

$$[R_{NN}](K_i) = \lambda_i(K_i) \quad (2.7)$$

固有ベクトル (K_i) を用いて, 変換行列 $[K_{NN}]$ を作ると次式を得る.

$$[K_{NN}] = \begin{bmatrix} K_1^t \\ K_2^t \\ \dots \\ K_N^t \end{bmatrix} \quad (2.8)$$

(K_i) が固有ベクトルであるから, 次のような計算ができる.

$$\begin{aligned} [K][R][K]^t &= [K][\lambda_i(K_i)] \\ &= \begin{bmatrix} \lambda_1(k_1, k_1) & \lambda_2(k_1, k_2) & \dots & \lambda_N(k_1, K_N) \\ \lambda_1(k_2, k_1) & \lambda_2(k_2, k_2) & \dots & \lambda_N(k_2, K_N) \\ \dots & \dots & \dots & \dots \\ \lambda_1(k_1, k_1) & \lambda_2(k_1, k_2) & \dots & \lambda_N(k_1, K_N) \end{bmatrix} \\ &= \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix} \end{aligned} \quad (2.9)$$

上式のように, 変換係数 K_i は無関数になっており, もっとも効率の高い方法として知られている [2]. ただし基底ベクトル $[K]$ は, 画像信号の相関関数 $[R]$ に関係しているため, 画像が変わる毎にこれを求め直さなければならず, また受信側に変更内容を伝える必要があり, 処理が複雑である [2].

2.2.4 離散コサイン変換 (DCT)

離散コサイン変換は、国際標準の符号化方式に採用されている [2].

N 次の 1 次元離散コサイン変換、逆変換はそれぞれ下式のように表される.

$$F(u) = \sqrt{\frac{2}{N}} \sum_{x=0}^{N-1} f(x) \cos \frac{(2x+1)\pi}{2N} u \quad (2.10)$$

(u=0,1,...N-1)

$$f(x) = \sqrt{\frac{2}{N}} \sum_{u=0}^{N-1} D(x) \cos \frac{(2x+1)\pi}{2N} u \quad (2.11)$$

(x=0,1,...N-1)

ブロックサイズ ($N \times N$) の 2 次元離散コサイン変換、逆変換はそれぞれ下式のようになる.

$$F(u, v) = \frac{2C(u)C(v)}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)\pi}{2N} u\right] \cos\left[\frac{(2y+1)\pi}{2N} v\right] \quad (2.12)$$

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)\pi}{2N} u\right] \cos\left[\frac{(2y+1)\pi}{2N} v\right] \quad (2.13)$$

(x,y=0,1,...N-1)

但し,

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u=0, v=0 \\ 1 & u, v \neq 0 \end{cases}$$

2 次元離散コサイン変換は、水平方向、垂直方向それぞれに 1 次元離散コサイン変換を行うことにより、実現することができる。またその基底ベクトルの形がカルーネン レーベ変換に最もており、かつ処理方式が簡単である [3].

2.3 予測符号化

ここでは、画像圧縮技術の方法の1つである予測符号化説明し、その構成図を図 2.4 に示す。またその中で、フレーム間予測符号化と動き補償予測の概要を述べる [2]。

予測符号化は符号化済みの過去の画素値を用いて、これから符号化しようとする画素に対する予測信号を構成し、その予測誤差信号を伝送するものである。予測誤差信号は、画像信号そのものに比べ、エネルギーは零値近傍に集中する [2]。

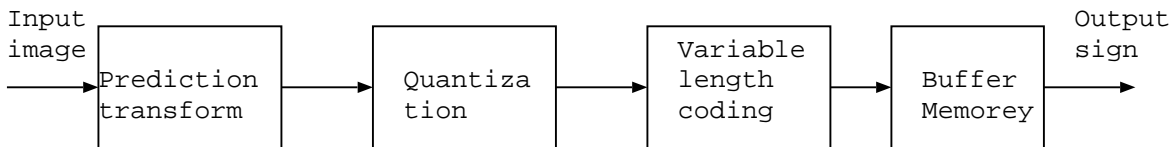


図 2.4 予測符号化方式の構成図

2.3.1 予測符号化の特徴

- 伝送路の誤りに弱い
- 圧縮率が低いときに有効
- その処理は簡単である

2.3.2 フレーム間予測符号化

1 フレーム以上前の画素を用いて予測を行なう方式である。この方式は、送信側、受信側両方に 1 フレーム前の画像信号を記憶するフレームメモリを持つことにより実現できる。現フレームの画像と直前に処理した前フレームとの差分をとることによって、静止している領域に対しては差が 0 となり、情報を伝送する必要がなく、動いている領域についてのみ信号を伝送すればよい。これは静止領域の多い画像に対して有効である [2]。

2.3.3 動き補償符号化

前フレームから現フレームまでの間に、被写体が動いた方向と距離を表す位置情報(動きベクトル)を調べ、この動き位置情報で表される位置の画素値を予測信号として用いる方法である。画像を複数の小さなブロックに分割し、そのブロック毎に一つの動き位置情報が伝送される。ブロックの大きさが大きいほどブロックの数が少なくなるため、伝送する動き位置情報の数少なくなり有利であるが、逆に予測誤差が0にならない画素の数が多くなる [2]。

2.4 その他の動画像圧縮に用いる技術

2.4.1 可変長符号化(エントロピー符号化)

エントロピー符号化は符号量を削減するための方法で、出現頻度の多いものに短い符号を割り当てる。動画像圧縮の場合、DCT係数を求め、これらの値を1,0の符号に置き換える。主にハフマン符号や、算術符号などが使用される [2]。

ハフマン符号は、事象の発生頻度をあらかじめ想定して符号表を作成し、それに対して符号化を行なう。しかし、出現確率に合わせた符号表を作成するのが難しいという難点がある。それに対し、算術符号化は、符号表を必要としない特徴があり、常に統計的性質を監視しながら、これに合わせて符号化を行なう。そのため、符号表をあらかじめ用意する必要がなく、符号化の途中で出現確率が変わっても確率表を書き換えることで容易に対応できる利点がある [3]。

2.4.2 明るさの省略

入力信号の明るさは、黒から白までの256通りで表されている。この情報を削減するために、目の特性を利用する。人間の目は、変化の激しい領域では、明るさの区別がつきにくいという特性がある。この特性を利用して、DCT係数のうち、周波数の高い係数の情報を省略する [3]。

2.4.3 間引き

複数のサンプリングレートを用いるマルチレート信号処理の基本的な操作であるレート変換技術として、レートを下げる操作のことである [3].

2.5 JPEG について

JPEG の特徴には

- 連続した階調をもつデジタル静止画を扱う広い応用に対応
- 利用し得る限りでベストの品質の圧縮技術
- ユーザの要求により圧縮率と品質が選択可能
- 計算があまり複雑でなく、高速で計算可能

などがある.

ITU-TS (国際電気通信連合:旧 CCITT) と ISO (国際標準化機構) で定めたカラー静止画像の圧縮, 展開を決める規格. 規格を制定した標準作成委員会 (Joint Photographic Expert Group) がそのまま規格の名前になっている. JPEG は, 人間の目が低周波成分については感度が良いが高周波成分で鈍いという視覚特性を利用している. フルカラーあるいはグレースケール画像を対象とした圧縮方式で, 2 値画像は対象としていない. 圧縮率が高いわりに画質の低下が少ないのが特徴. 圧縮や伸長の処理に時間がかかる欠点があるが, 圧縮率が変更できるので, 圧縮での画質の低下の程度を変えることで, 画質の品質とファイルサイズ, および処理時間の組み合わせの中から最適なものを選ぶことができる. 一般的に JPEG は, 非可逆性圧縮を指すことが多いが, 可逆性圧縮方式の標準化も提案されている. QuickTime の標準圧縮方式としても採用されている [1].

2.6 MPEG について

MPEG の特徴には

- 標準的な動画を 1.0 から 1.5Mbps で供給 (MPEG-1)
- 対称的／非対称的 圧縮／解凍の応用を支援
- 圧縮した時にも，ランダムアクセスプレイバックが可能
- 圧縮時に，早送り，早巻き戻し，巻き戻しも可能
- 音声と動画が同期
- 破局的なデータエラーを回避
- 必要に応じ，圧縮，解凍の遅れが制御可能
- 必要なら，編集も可能
- 画面（ウィンドウ）上で動画がプレイできる
- 低価格のハードウェア（チップ）による圧縮が可能

などがある。

ITU-TS（国際電気通信連合：旧 CCITT）と ISO（国際標準化機構）で制定されたデジタル動画と音声の圧縮・伸長に関する規格。基本的には，動画の 1 コマ目のデータをもとに，2 コマ目はそれと違う部分だけ，3 コマ目は 2 コマ目と違う部分だけと，差分を記録することで圧縮を実現する。このため動きの大きな動画の場合は，圧縮率が落ちる。実際に規格の作成にあたった ISO の下部組織である Motion Picture Experts Group の略称がそのまま規格の名前になっている。MPEG-1 は，データ転送レート 1.5Mbps までの CD，DAT，ハードディスク等のデジタルメディアの動画と関連する音響を対象とした規格で，Video CD に採用されている。MPEG-2 はデータ転送レート数 M～数十 M で，放送メディアが対象，DVD-Video でも採用されている。MPEG-1 は QuickTime で再生可能。1999 年 8 月に標準化された MPEG-4 は，符号化の向上に加え，画像を物体毎に符号化できるオブジェクトベース符号化や誤り耐性符号化などを大きな特徴としている [1]。

2.6.1 MPEG の符号化アルゴリズム

MPEG の符号化アルゴリズムについて説明する。

1. 画像分解

入力される画像をそのまま圧縮すると、非常に多くの演算処理が必要となる。高速処理のために入力画像を縦横 8×8 画素に分解する。

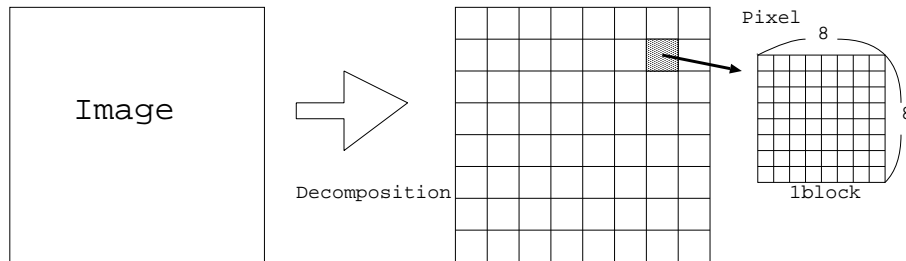


図 2.5 画像の分解

2. 動き補償・フレーム間予測

動画は複数のフレーム画像により構成されており、直前のフレームをもとに、現フレームとの差のみを抽出して符号化すれば、情報量を削減することができる。また画面内で動く物体を検出し、その動きを予測してその結果と現フレームとの差を抽出するとさらに情報量を削減することができる。そこで得られた予測誤差を離散コサイン変換 (DCT) する。

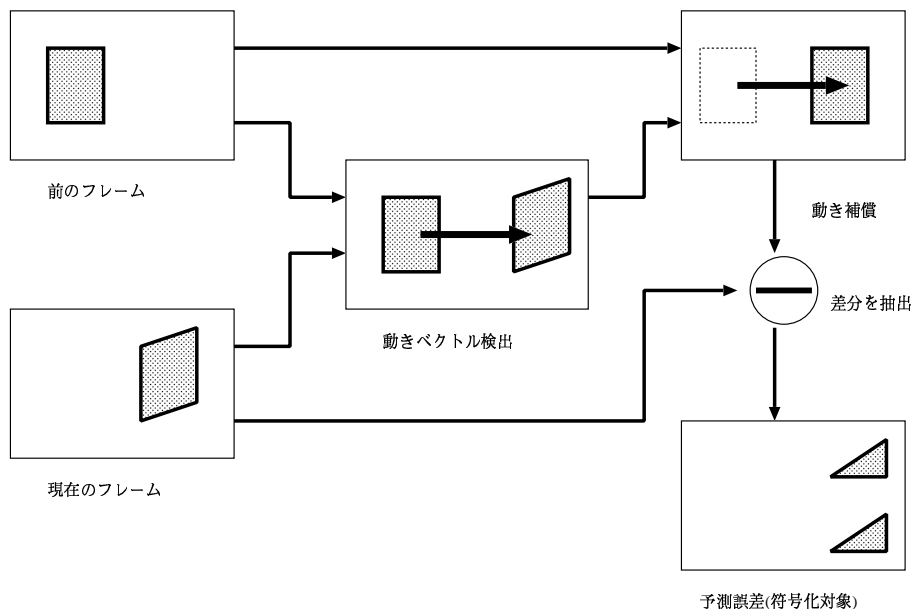


図 2.6 動き補償・フレーム間予測

3. 離散コサイン変換 (DCT)

2 で得られた予測誤差を MPEG では 8×8 の 64 画素を 1 ブロックとして, DCT を行ない DCT 係数を求める. $N = 8$ となるので, 式 2.17 と式 2.18 はそれぞれ次のようになる.

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left[\frac{(2x+1)\pi}{16}u\right] \cos\left[\frac{(2y+1)\pi}{16}v\right] \quad (2.14)$$

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)\pi}{16}u\right] \cos\left[\frac{(2y+1)\pi}{16}v\right] \quad (2.15)$$

$$\text{但し } C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u=0, v=0 \\ 1 & u, v \neq 0 \end{cases}$$

$u=v=0$ の場合, コサイン関数の角度が 0 なので, コサイン関数の値がいずれも 1 になり,

$$F(0, 0) = \frac{1}{8} \sum_{u=0}^7 \sum_{v=0}^7 f(u, v) \quad (2.16)$$

となる。この値 $F(0,0)$ は縦横の周波数成分がいずれも 0 であり、DC 係数の値を表す。式 3.3 から分かるように、DC 係数 $f(0,0)$ は、ブロック内の画素値 $f(x,y)$ の総和を 8 で除した値になるので、ブロック内の画素値の平均値に比例する。

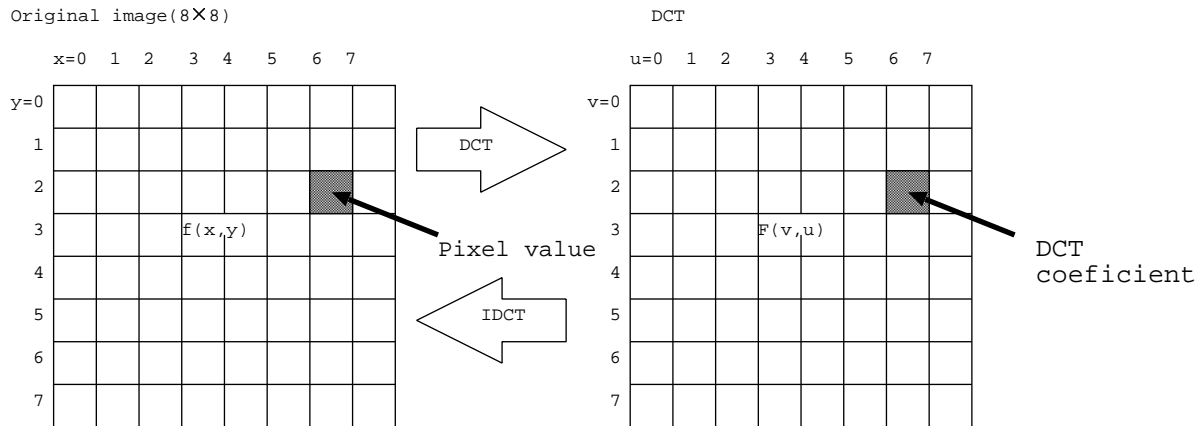


図 2.7 2次元離散コサイン変換 (DCT)

4. エントロピー符号化

エントロピー符号化は符号量を削減するための方法で、出現頻度の多いものに短い符号を割り当てる。MPEG の場合、ハフマン符号を使用し、DCT 係数を 1,0 の符号に置き換える。

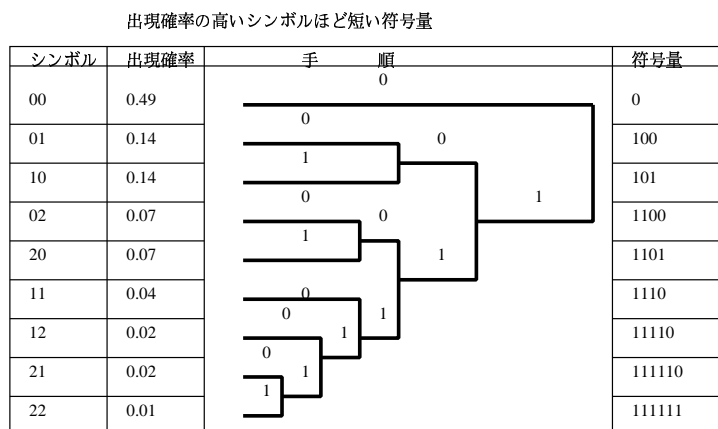


図 2.8 エントロピー符号化

5. MPEG の符号化アルゴリズム

手順 1~4 までが MPEG でのおおまかな符号化アルゴリズムである。図 2.9 に MPEG の符号化アルゴリズムの図を示す。

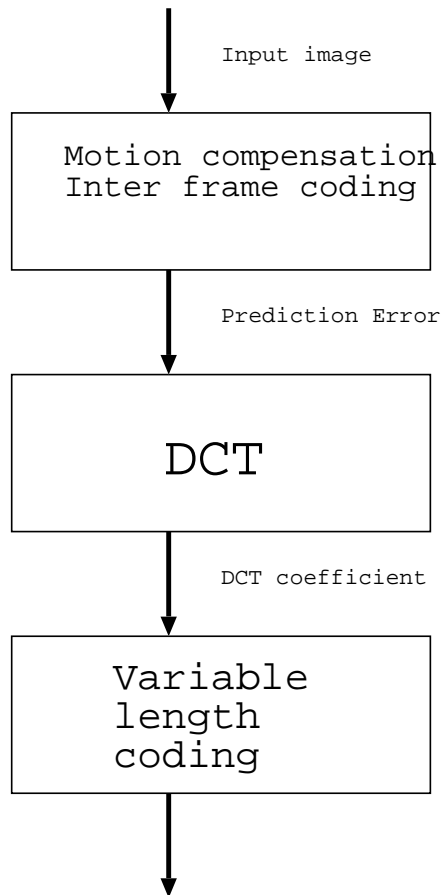


図 2.9 MPEG の符号化アルゴリズム

第3章

符号化アルゴリズム

3.1 はじめに

本研究では、できるだけ発生情報量を削減できる符号化アルゴリズムを提案する。2章で色々な圧縮符号化の方法を説明したが、変換符号化、予測符号化の両方の利点を生かし、またその他の圧縮技術を使用した符号化アルゴリズムを提案する。提案する符号化処理のアルゴリズムの構成図を図3.1に示す。また圧縮された動画像を携帯電話に見れるように伝送するための方法も提案する。

3.2 符号化アルゴリズム

1. 画像分解

まず、入力画像を分解する。この場合、サイズを小さくすると、圧縮効果が少なくなってしまうので [3], MPEG などでも使用されている 8×8 の 64 画素を 1 ブロックとして分解する方法を使用する。

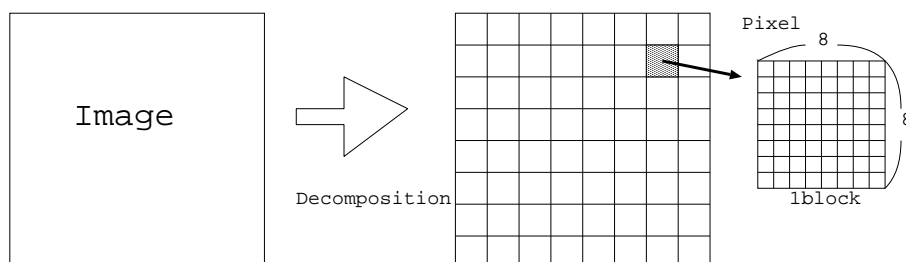


図3.1 画像の分解

2. 動き補償, フレーム間予測

次に動き補償により, 動いている領域については, 予測誤差を小さくできるが [2], 被写体の背景領域を分離することにより, さらに効率的な符号化が望める. そこで, 背景があまり変化しない場合, あらかじめ背景メモリを設置し, これに背景画像を記憶させておく. 以後, 背景はメモリに保存されているものを使用する. 被写体の領域については, 動き補償, フレーム間予測を行う.

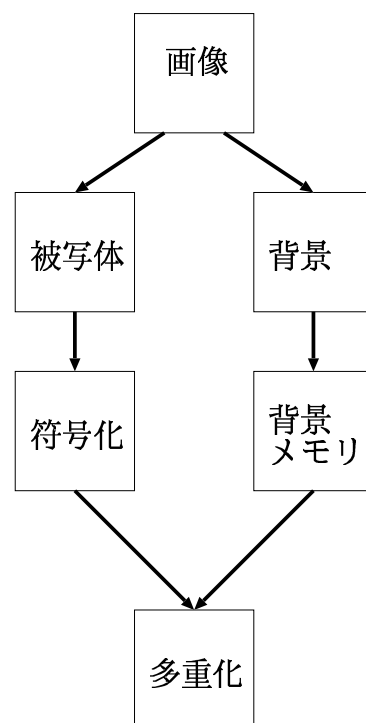


図 3.2 分離

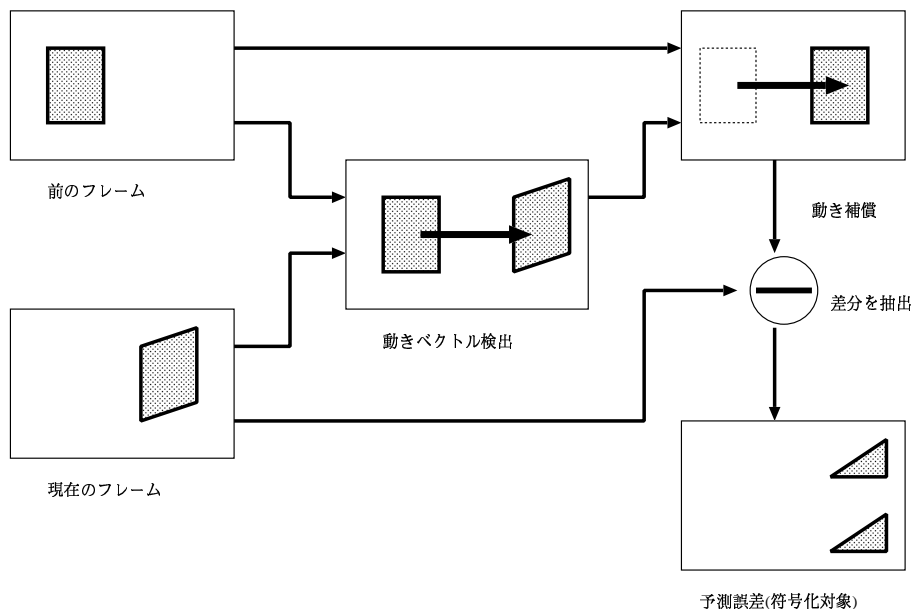


図 3.3 動き補償・フレーム間予測

3. 離散コサイン変換

これから得られた予測誤差を離散コサイン変換する。動き補償により得られた予測誤差で構成される画像においては、動いている被写体の画像信号はもともと輪郭がぼける傾向にあり、高周波成分の係数の値は小さくなる [2]。また動いている被写体に対しては、高周波成分に対する視覚感度が弱いという特徴があるので [2]、動き補償予測誤差を対象とした離散コサイン変換は、高周波成分の DCT 係数をより大幅に省略できることが分かる。

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left[\frac{(2x+1)\pi}{16}u\right] \cos\left[\frac{(2y+1)\pi}{16}v\right] \quad (3.1)$$

$$f(x, y) = \frac{1}{4} \sum_{u=0}^7 \sum_{v=0}^7 C(u)C(v)F(u, v) \cos\left[\frac{(2x+1)\pi}{16}u\right] \cos\left[\frac{(2y+1)\pi}{16}v\right] \quad (3.2)$$

4. エントロピー符号化

そこから得られた DCT 係数をエントロピー符号化する。

出現確率の高いシンボルほど短い符号量

シンボル	出現確率	手 順	符号量
00	0.49	0	0
01	0.14	0 0	100
10	0.14	1 0	101
02	0.07	0 1 0	1100
20	0.07	1 1 0	1101
11	0.04	0 1 1	1110
12	0.02	0 1 1 0	11110
21	0.02	1 1 1 0	111110
22	0.01	1 1 1 1 0	111111

図 3.4 エントロピー符号化

これら 1~4 の技術を使って作成した符号化アルゴリズムの図を下に示す。

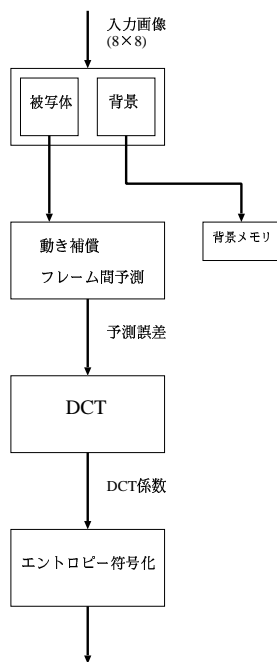


図 3.5 符号化アルゴリズム

3.3 通信方法

動画像を圧縮することができても、それを伝送する方法がなければ、実際に見ることはできない。また携帯電話という狭い帯域で伝送することを考慮した通信方式を提案する。

上記に示した方法で圧縮された動画像を低ビットレートで伝送し、携帯端末で見るようにするには、携帯端末の画像サイズを小さくする必要がある。この場合、 160×120 ドットに指定する。また、フレームレートは、テレビや MPEG などは 30 フレーム/秒であるが、動きの少ない被写体を想定するため 15 フレーム/秒まで間引く。

この結果、動画像圧縮における符号化アルゴリズムと低ビットレートで動画を伝送することが実現できる。

第4章

結論

4.1 MPEG との違い

3章で提案した符号化アルゴリズムと2章ですでに提案されているMPEGのアルゴリズムを比較してみると、やはりMPEGと近い形になっているのが分かる。違う点は、動き補償、フレーム間予測の処理を行なう前に、被写体と背景を分離している部分である。MPEGなどの技術で動画像をそのまま処理していくよりも、効率的な圧縮が実現できる。

4.2 今後の課題

問題点としては、ブロック歪みの発生と動きベクトルの誤検出がある。隣接したブロックの動きベクトル量が等しい場合は、動き補償予測画像のブロック境界に歪みがなく正常に接続される。ところが、実際の動画像では、平行移動をするような場面が少ないため、隣接した画素の動きベクトル量が等しくならないことが多い。すると、予測画像の境界部分には、重なり、ずれ、飛びなどの不連続性からなる歪みが発生してしまう。重なり、ずれ、飛びなどによる動き補償予測画像の歪みはブロックの境界、特に四隅に生じやすい。したがって、このような問題点を改善していく必要がある。

4.3 まとめ

本研究では条件付きではあるが、被写体と背景を分離して圧縮し、それを低ビットレートで伝送できる圧縮方法を示した。

謝辞

本研究を行なうにあたり，ご指導，ご助言を頂いた福本昌弘に深く感謝いたします。また本論文のご審査をしていただき島村和典教授，菊地豊助教授に深く感謝いたします。最後に，日頃から多面にわたってご協力を頂いた福本研究室の諸氏に感謝いたします。

参考文献

- [1] 谷萩 隆嗣, 音声と画像のデジタル信号処理, コロナ社, 1996
- [2] 黒田 英夫, 画像符号化技術, 昭晃堂, Dec.1996
- [3] 越知 宏, 黒田 英夫 画像圧縮技術, 日本実業出版社, 1999
- [4] 半谷 精一郎, デジタル信号処理, コロナ社, 2000
- [5] 笠野 英松, 入門 MPEG-4, 技術評論社, 2000
- [6] 安田 浩, 藤原 洋, デジタル放送・インターネットのための情報圧縮技術, 共立出版, 1998
- [7] 佐藤 政次, 電子情報通信ハンドブック, 1998
- [8] 江戸川 + 次世代通信研究会, 次世代携帯電話がわかる, 2000
- [9] 寺田 文行, 中村直人, 釈氏孝浩, 松井辰則, 情報数学の基礎, サイエンス社, 1999

付録 A

MPEG の基本構成

MPEG のもっとも重要な点は、動画像圧縮と早送り・逆送り機能の実現である。動画像圧縮のために、前方向予測を行なう P ピクチャと、双方向予測を行なう B ピクチャがあり、早送り・逆送り機能を実現するためにフレーム内符号化を行なう I ピクチャがある。

これら 3 種類のピクチャの構造図 A.1 に示す。予測用画像メモリの中には、既に符号化された過去の複数フレーム分の画像信号が記憶される。また、入力画像は一旦時間調整用画像メモリに数フレームにわたって記憶される。

これによって、P ピクチャを符号化する場合には、予測用画像メモリの出力を入力画像に対する予測信号として、前方向予測を行ない、予測誤差を得る。

B ピクチャを符号化する場合には、時間調整用画像メモリから符号化すべき画像を読み出し、この画像に対して過去および未来のフレームの受信画像に相当する画像を予測用画像メモリの中から読み出し、これらを予測画像として双方向予測を行ない、予測誤差を得る。時間調整用画像メモリに記録されている数フレーム前の画像を基準に処理すれば、それより後のフレームは未来の画像になるので、予測画像メモリ中の未来の画像との比較が可能となる。

I ピクチャを符号化する場合には、入力画像をそのまま使用する。切替回路は、I、P、B ピクチャに応じてそれぞれの信号を切り替える。切り替えられたそれぞれの信号は DCT により DCT 係数に変換される。この DCT 係数と動き位置情報がエントロピー符号化により、発生確率の高い信号には短い符号を、低い信号には長い符号を割り当てられ、符号化信号として蓄積/伝送される。動き位置情報は、前方向予測および双方向予測により得られた信号が使用される。

また復元側と同じ信号を作るために、符号化側でも DCT 係数を逆 DCT により予測誤差信号に変換し、得られた予測誤差と、動き位置情報を用いて、I, P, B ピクチャを復元し、続く入力画像のための予測信号として予測用画像メモリに蓄積する。これらすべてを符号化するための基本構成を図 A.2 示す。

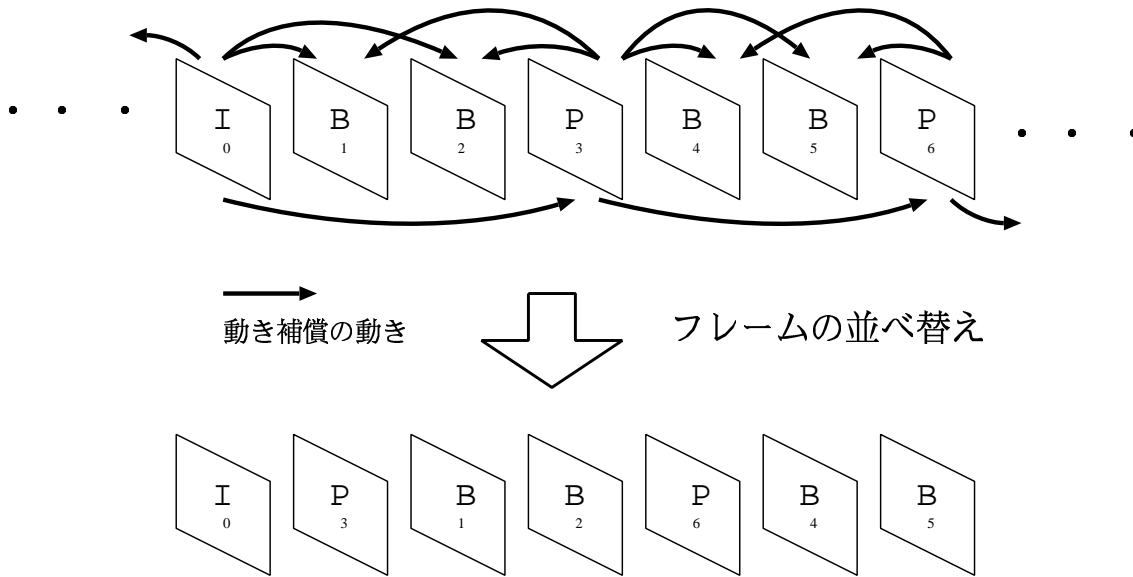


図 A.1 P,B ピクチャ予測構造と I ピクチャ

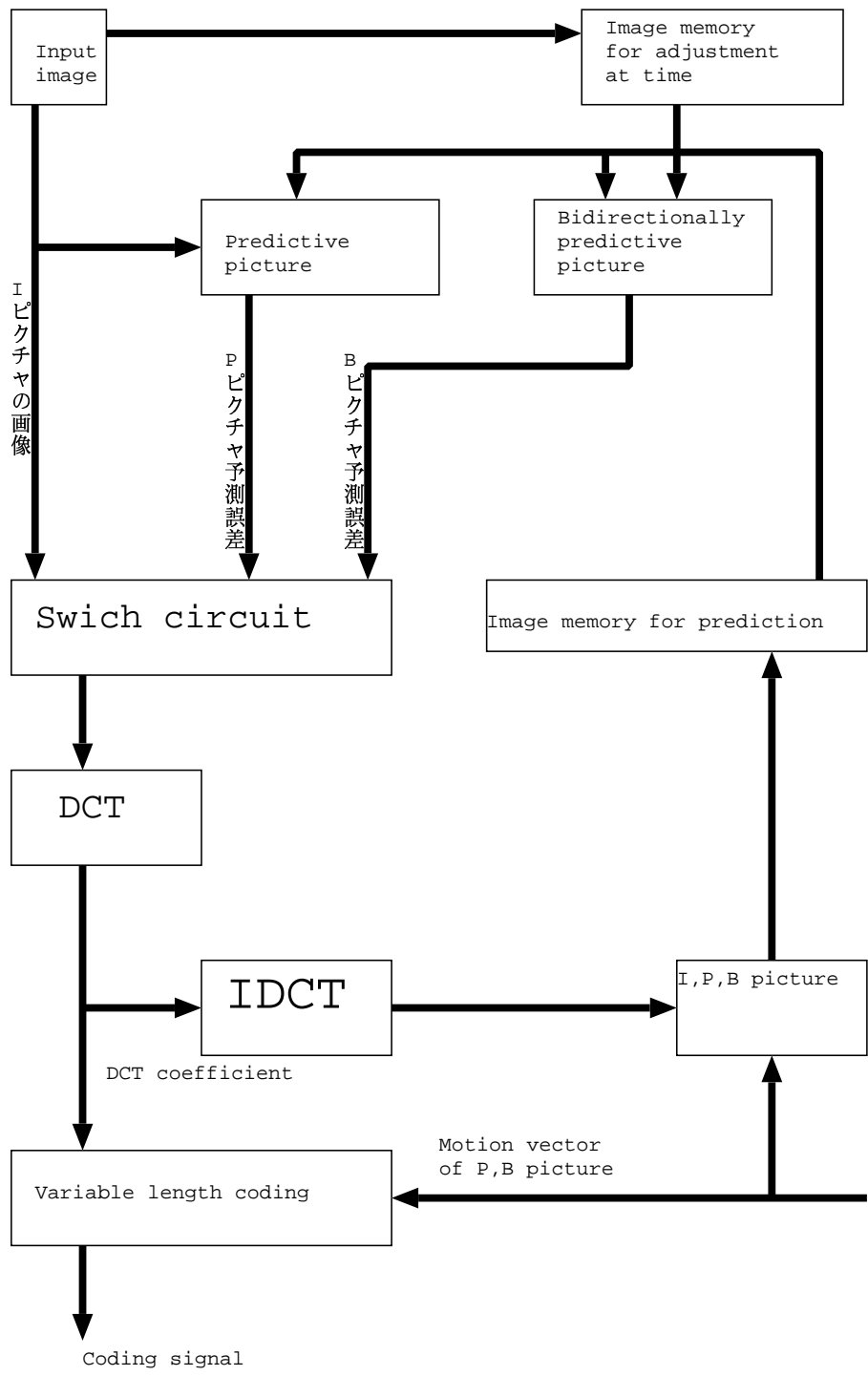


図 A.2 MPEG 方式の基本構成

付録 B

DCT のプログラム

```
#define L 64
#define L2 32
#define N 64
#define N2 32
#define NN 4096
#define K 4
#define PI 3.14159265358979
#ifndef max
#define max(A,B) ((A) < (B) ? (A) : (B))
#endif

void dct2d (
    double s0[N][N],
        unsigned int L,
        double p[],
        double q[],
        unsigned int sup,
        double s1[N2][N2],
```

```

        double w1v[N2][N2],
        double w1h[N2][N2],
        double w1d[N2][N2]);

void idct2d (
    double s1[N2][N2],
    double w1v[N2][N2],
    double w1h[N2][N2],
    double w1d[N2][N2],
    unsigned int s'len,
    double p[],
    double q[],
    unsigned int sup,
    double s0[N][N]);

void main(void);

#include <stdio.h>
#include <math.h>
#include "fwt.h"

void dct2d();
void main()–

```

```

double s0[N][N];

double s0r[N][N];

double p[K] = -0.482962913145, 0.836516303738,
0.224143868042, -0.129409522551";

double q[K];

double s1[N2][N2], w1h[N2][N2], w1v[N2][N2], w1d[N2][N2];

int i,j;

int sup = K;

for(i = 0; i < sup; i++)-
    q[i] = pow(-1,i)*p[sup-i-1];
"

for(j = 0; j < N; j++)-
    for(i = 0; i < N; i++)-
        s0[j][i] = 0.2;
"
"

for(i = 5; i < N-5; i++)-
    s0[5][i+1] = 1.0;
    s0[N-6][i+1] = 1.0;
    s0[i][6] = 1.0;
    s0[i][N-5] = 1.0;

```

```

    s0[i][i+1] = 1.0;
    s0[i][N-i] = 1.0;
    "
printf("入力信号 "n");
for(j = 0; j < N; j++)
    for(i = 0; i < N; i++)
        printf("%7.4f ",s0[j][i]);
    "
    printf("n");
    "

dct2d(s0,N,p,q,sup,s1,w1v,w1h,w1d);
/* idct2d(s1,w1v,w1h,w1d,N/2,p,q,sup,s0r);

printf("再構成信号 "n");
for(j = 0; j < N; j++)
    for(i = 0; i < N; i++)
        printf("%7.4f ",s0r[j][i]);
    "
    printf("n");
    "
*/

```

```

void dct2d (
    double X ;
    int u,v,x,y;

    for (u=0 ; u<L ; u++)-
for(v=0 ; v<L ; v++)-
    if(u==0 && v==0)
        c(u)=1/sqrt(2) , c(v)=1/sqrt(2);
else
    c(u)=1 , c(v)=1;
    for(x=0; x<L ; x++)-
        for(y=0 ; y<L ; y++)
            X+=(cos((2*x+1)*u*PI/(2*L)))*
(cos((2*y+1)*v*PI/(2*L)))*s0[N][N]
sum=2*c(u)*c(v)*X/N

```

