



Title	動画像符号化のための高能率フレーム間動き補償予測に関する研究
Author(s)	浅水, 仁
Citation	北海道大学. 博士(工学) 甲第4753号
Issue Date	1999-03-25
DOI	10.11501/3151435
Doc URL	<a href="http://hdl.handle.net/2115/51598">http://hdl.handle.net/2115/51598</a>
Type	theses (doctoral)
File Information	000000336625.pdf



[Instructions for use](#)

動画様符号化のための高能率  
フレーム間動き補償予測に関する研究

浅水 仁

北海道大学大学院 工学研究科 電子情報工学専攻

動画像符号化のための高能率  
フレーム間動き補償予測に関する研究

Efficient Motion Compensated Interframe Prediction  
for Video Coding

浅水 仁

Satoshi Asamizu

北海道大学大学院 工学研究科 電子情報工学専攻

Division of Electronics and Information Engineering,  
Graduate School of Engineering, Hokkaido University

The document is submitted to Hokkaido University  
in February 1999, as doctoral dissertation.

# 目次

1 序論	1
1.1 本研究の背景	1
1.2 本研究の目的	3
1.3 論文の構成	3
2 国際標準動画像符号化	6
2.1 はじめに	6
2.2 動き補償	7
2.2.1 フレーム間相関の利用	7
2.2.2 動き補償	7
2.3 通信用符号化	8
2.3.1 $p \times 64$ kb/s 標準化	9
2.3.2 映像フォーマット	9
2.3.3 ソース符号化	10
2.4 MPEG1	14
2.4.1 符号化システム	15
2.4.2 符号化アルゴリズム	15
2.4.3 データ構造	16
2.4.4 復号化方式	18
2.5 MPEG2	25
2.5.1 標準アルゴリズム	26

2.5.2	スケーラビリティ機能	29
2.5.3	復号化アルゴリズム	30
2.6	MPEG4	31
2.6.1	MPEG4の概要と特徴	31
2.7	まとめ	32
<b>3</b>	<b>動きベクトル検出法</b>	<b>33</b>
3.1	はじめに	33
3.2	勾配法	33
3.3	勾配法：画素反復法	36
3.4	ブロック・マッチング法	38
3.5	まとめ	41
<b>4</b>	<b>遺伝的アルゴリズム</b>	<b>43</b>
4.1	遺伝的アルゴリズムの概要	43
4.1.1	遺伝的アルゴリズムとは	43
4.1.2	遺伝的アルゴリズムの基本的な考え方	43
4.2	単純遺伝的アルゴリズム	44
4.2.1	仮想生物と環境の設定	44
4.2.2	GAの処理手順	46
4.2.3	単純GAの特徴	52
4.3	スキーマ定理	53
4.4	まとめ	55
<b>5</b>	<b>テーブル参照による動き補償</b>	<b>56</b>
5.1	はじめに	56
5.2	動きテーブル参照による動き補償	57
5.2.1	動きテーブル参照による動き補償の考え方	57
5.2.2	動きベクトルの出現頻度を用いた動きテーブル作成法	58
5.2.3	GAを用いた動きテーブル作成法	60
5.3	実験および考察	63
5.3.1	動き補償予測誤差による予測性能の比較	63

5.3.2	Freq と GA の比較 . . . . .	65
5.4	むすび . . . . .	69
6	テーブル参照による動き補償のための動きモデルの検討 . . . . .	80
6.1	はじめに . . . . .	80
6.2	アフィン動きモデル . . . . .	81
6.2.1	動きのパラメータ表現 . . . . .	81
6.2.2	アフィン変換 . . . . .	81
6.2.3	パラメータの推定 . . . . .	83
6.3	動きテーブル参照による動き補償 . . . . .	84
6.3.1	画素ごとの動きを用いた動きテーブル作成法 . . . . .	84
6.3.2	動きベクトルの階層的検出法 . . . . .	85
6.3.3	動きパラメータ推定 . . . . .	85
6.3.4	ベクトル量子化 . . . . .	85
6.4	GA を用いた動きテーブル作成法 . . . . .	89
6.4.1	提案手法のための GA の設計 . . . . .	89
6.5	実験および考察 . . . . .	91
6.5.1	動き補償予測誤差による予測性能の比較 . . . . .	91
6.5.2	GA による動きテーブルの特色 . . . . .	92
6.6	DCT を用いた動きモデル . . . . .	99
6.6.1	階層画像を用いた画素毎の動きベクトル検出 . . . . .	99
6.6.2	DCT 変換係数を用いた動きモデル . . . . .	99
6.6.3	実験 . . . . .	102
6.7	テーブル参照による動き補償 . . . . .	102
6.7.1	画素毎の動きを用いたテーブル作成法 . . . . .	102
6.8	予測性能の比較 . . . . .	103
6.8.1	動きモデルの違いによる比較 . . . . .	103
6.8.2	実験 . . . . .	103
6.9	性質の異なる変換係数毎にテーブル作成 . . . . .	104
6.10	まとめ . . . . .	106

7 結論	108
謝辞	112
文献	113
著者の研究業績	118

# 第 1 章 序論

## 1.1 本研究の背景

インターネット，移動体通信，デジタル放送等で現在画像通信の分野は急速な発展を遂げている．画像のデジタル化の応用は，テレビ局のスタジオ装置や画像伝送装置に始まり，テレビ会議，テレビ電話用の符号化装置では必須のものとなった．特にテレビ電話やテレビ会議では，伝送チャンネルのビットレートが限られているために，圧縮技術を駆使して動画像データの冗長性を削減することが重要である．

PC(Personal Computer)の発展とともにデジタル化された画像が急速に広まった．PCの限られたハードディスクやCD-ROM(Compact Disk-Read Only Memory)にデジタル化された動画像を収納できるようにするためにビデオ符号化の研究と実用化が進んでいる．

さらに，デジタル変調技術の進歩によって，データ圧縮した動画像によるデジタルテレビの衛星放送の方が，アナログ衛星放送よりも有効に電波資源を活用できることが確認されている．そして，画像はいったんデジタル化されると，放送や通信やパッケージメディアを問わずにデータ交換できることから，メディアの枠を越えたアプリケーションが生まれている．そのため，不足しつつある伝送路の大容量化と同時に経済化を図る必要がある．このように伝送路の経済化を図るためには，高能率符号化による伝送レートの削減が強く要求され，様々な所で研究がなされている．

実際の動画像符号化では，入力種別並びに適用形態から求められる特性（符号化速度，符号化品質）を考慮して，様々なアルゴリズムを選択あるいは組み合わせることによって符号化方式が構成される．構成される符号化方式は，ISDN(Integrated Services Digital Network)における伝送を考える場合，何らかの標準化が必要となってくる．よって，ITU-T（国際無線通信連合電気通信標準化部門）等で標準化が行われ，勧告 H.261 や MPEG（



Moving Picture Experts Group) という映像符号化標準が登場した。

動画像データの圧縮のためにはまず、時間方向の冗長度の削減が重要である。従って、2枚の連続する画像において、高精度で動き情報を抽出することは全体の符号化効率に大きな影響を及ぼす。動き情報がどのように抽出されるかで、その後の動き補償予測誤差信号の性質が異なる。このことにより、予測誤差信号の高周波領域及び低周波領域へのビットの割り当て方が変わって来るので、画質にも影響を及ぼす。よって、動画像データの圧縮のための動きベクトル検出法が色々検討されている。これまでに提案されている動きベクトルの検出手法で代表的なものとしてブロック・マッチング法と勾配法がある。ブロック・マッチング法は動きを高精度で検出することができ比較的大きな動きも検出できるが、膨大な処理量を要するという欠点がある。また、勾配法は処理量が少ないという利点があるが、大きな動きに対する検出精度が低く雑音に弱いという欠点がある。

動き補償の問題点としては、ブロック歪が挙げられる。離散余弦変換 (DCT) のようなブロックに閉じた変換であればブロック内では問題はない。しかし、最近注目されてきたウェーブレット変換のような非ブロック変換においては、ブロック境界部分の歪が全体の符号化効率の低下を招く。

予測誤差信号の符号化技術は大別すると、時間軸上で処理を行なう予測符号化と周波数軸上で処理を行なう変換符号化とに分けられる。これらの技術は静止画像の符号化技術に大きく貢献したが、従来予測符号化は高速性、変換符号化は高能率というように目的により使い分けられた。ところが、現在の VLSI 技術の急速な発展により、変換符号化の処理速度が予測符号化に引けを取らないようになった。従って、現在では変換符号化の方が主流である。動画像符号化に対して現在のところ主流となっている技術は、動き補償フレーム間予測と変換符号化の一種である離散余弦変換 (DCT) のハイブリッド符号化である。

予測誤差信号の符号化における問題点は、静止画像の符号化に大きな威力を発揮した変換符号化が、動画像のフレーム間予測誤差信号の符号化にはその特性を十分に発揮できないということである。フレーム間予測誤差信号には高周波成分が多い。上で述べた動き補償によるブロック境界部分の歪もその一因となっている。従って、変換を行っても低周波数領域に信号が集中せず、効果的なビット割り当てができない。

現在では、家庭の PC でも手軽に動画像を見ることができるようになっている。また、国際標準として MPEG4 の検討が開始され、移動体通信の分野にまで動画像通信技術が入り込もうとしている。以上のように、動画像の符号化はますます重要な技術となっており、

これからの時代に必要な技術となっている。そして、この技術の大いなる発展が現在必要とされている。

## 1.2 本研究の目的

本論文は、動画像符号化において重要な要素技術である、フレーム間動き補償予測に関する研究をまとめたものである。

多くの情報媒体の中で、言うまでもなく動画像は重要な位置を占める。コンピュータ内部及びネットワーク上で、大量の画像データが扱われている。なかでも動画像表現には、通常膨大なデータ量が関与し、高性能な画像圧縮アルゴリズムが常に求められている。本論文はこのような背景のもとで、動画像符号化において重要な要素技術である、フレーム間動き補償予測を効率良く行なうため、動きベクトルの符号量削減を主題に据える。

動画像符号化のための高能率化フレーム間動き補償予測を目的としている。特に動き補償を中心として考える。動き補償は、動画像データの時間軸方向の冗長度を取り除く方法である。動きが近似している動きベクトルを効率よく削減することにより、動きに関する符号量を削減し、動き補償予測が符号化効率を高める上で重要であると考え、その点について詳細に検討を行なう。そして、上記を考慮に入れた新しい動き補償法を提案する。

## 1.3 論文の構成

本論文は7章からなり、論文の前半(第2章～第3章)で、動画像符号化について概説する。特に国際標準と動きベクトル検出法について詳しく説明する。後半(第5章～第7章)では、フレーム間動き補償予測を行なう際に、候補動きベクトルを削減し、全体の符号量を低く抑える方法についてのアルゴリズムを提案する。遺伝的アルゴリズムを用いて有効な使用動きベクトル数を限定する手法を提案する。そして、実際の動画像データを用いたシミュレーションによりその有効性を実証する。

本論文の第2章では動画像の符号化の概略を述べる。実際の動画像符号化では、入力信号種別並びに適用形態から求められる特性(符号化速度、符号化品質)を考慮して、様々なアルゴリズムを、選択あるいは組み合わせることによって符号化方式が構成される。ここでは、テレビ会議/テレビ電話信号、蓄積用動画像の符号化方式を現在進められている標準化の動向を踏まえて概説する。

第3章では、動画像符号化における重要な技術の1つである動きベクトルの検出法につ

いて述べる。高精度な動きベクトルの検出は動き補償を行なう上で必要なことであり、フレーム内挿やフレーム数変換に対しても必要とされる。この技術は後に続く処理の性能に大きく影響を及ぼすため今後の発展が大いに期待される。この章では動きベクトルの検出法についてブロック・マッチング法を中心に歴史的な流れとともに概説する。

第4章では、遺伝的アルゴリズム (Genetic Algorithm : GA) の概略について述べる。GAは生物進化(選択淘汰・突然変異)の原理に着想を得たアルゴリズムであり、確率的探索・学習・最適化の一手法である。GAはほとんどの最適化・探索の問題に適応可能な枠組で、緩やかな枠組である。この章では、GAの各種の規則やパラメータの設定等について説明する。

第5章では、動画像符号化のためテーブル参照による動き補償を提案する。様々な動きを格納した‘動きテーブル’を複数フレーム毎に作成し、対象ブロック毎に動きテーブル内を探索し、ブロック内の動き補償予測誤差を最小にする動きをテーブル内から選択する。本手法のポイントは動きに関する符号量削減のために動きベクトル数を削減することにある。動きベクトル数の減少による画質の悪化を抑えるために、次の2つの動きテーブル作成法を提案する。一つ目は、動き補償を行なった際に使用される動きベクトルの出現頻度により、動きベクトル数を削減する方法。二つ目は、遺伝的アルゴリズムを用いて、動き補償予測誤差が最小になるように動きベクトル数を削減し、動きテーブルを作成する方法である。こうして作成された動きテーブルを用いることにより、動き補償予測画像の画質を劣化を低く抑えつつ符号量を削減することを示す。本章では、動画像には動きの種類に偏りがある事を利用して、テーブルサイズを限定したテーブル参照による動き補償を用いる事によって、画質の劣化を抑えつつ従来法のブロックマッチング法より符号量の削減を実現する。実際の動画像データを対象とした符号化実験の結果により、本手法の有用性を示す。

第6章では、第5章で提案したテーブル参照による動き補償を用いて、ブロックの平行移動だけではない動きモデルを組合せた動き補償を提案する。一つ目に、小領域アフィン変換を用いた動き補償法を提案し、実際のシミュレーションによりその有効性を確認する。この方法は、小領域において、ブロック毎の動きベクトルから画素毎の動きベクトルをアフィン変換で表現する。アフィン変換のパラメータは、動きベクトルが空間方向に関して連続的であるという条件から決定される。動きベクトルが空間的に滑らかになるために、ブロック歪が抑えられ高周波成分を抑圧することができる。二つ目に、離散コサイン

変換 (DCT) を用いた動き補償を提案する。動画像を構成する各フレームをブロックに分割した時の、ブロック内の画素の動きには、高い相関があることが容易に推測することができる。この動きモデルは、画素毎の動きをブロック毎に DCT 変換を行ない、変換係数を求める。パラメータ数は、変換係数を低周波成分から高周波成分へとジグザグ走査をしながら、パラメータ数を決定する。動きを表すパラメータ数を自由に設定することができ、他の動きモデルに比べ柔軟に動きに関する符号量を変化させることができる。

これらの動きモデルを、実際の動画像データを対象とした符号化実験を行ない、その有効性を示す。

第7章では、まとめとして本研究の成果、残された課題及び将来への展望について述べる。

以上のように本研究では、動画像符号化のための動き補償予測に検討を行ない、高能率符号化を行なうための幾つかの手法を提案し、その有効性を数値実験により明らかにする。

## 第 2 章 国際標準動画像符号化

### 2.1 はじめに

本章では、動画像の符号化の概略を述べる。動画像符号化に用いられる各種技術を説明する。前章で述べたように、実際の動画像符号化では、入力種別並びに適用形態から求められる特性 (符号化速度, 符号化品質) を考慮して、様々なアルゴリズムを選択あるいは組み合わせることによって符号化方式が構成される。

構成される符号化方式は、ISDN (Integrated Services Digital Network) における伝送を考える場合、何らかの標準化が必要となってくる。よって、ITU-T (国際無線通信連合電気通信標準化部門) 等で標準化が行われ、様々な標準が生まれた。動画像については、先ず、64 kb/s ~ 2 Mb/s の ISDN を用いて動画像通信を行なうために「p × 64 kb/s オーディオビジュアル・サービス用ビデオ符号化方式」と題される勧告 H.261 という映像符号化標準が生まれた。次に、現在マルチメディアの世界で話題となっている「デジタル蓄積メディアのための動画と関連する音響の符号化」である MPEG (Moving Picture Experts Group) という映像符号化標準が登場した [18] ~ [23]。

このように、動画像符号化に関する標準化活動が活発に展開されている。動画像符号化の研究を行なう上では、これらの国際標準を常に念頭に置いて研究を行なう必要がある。動画像符号化に関する国際標準においては、デジタル蓄積メディア用符号化、テレビ会議 / テレビ電話用の通信用符号化に大別できる。本章では、動画像符号化に関する国際標準について説明する。但し、本論文では動画像符号化を対象としているので、各国際標準に付随する音響の符号化に関する標準については説明を省略する。

## 2.2 動き補償

ここでは、次節以降で説明する動画像符号化の国際標準を理解するための基礎知識として、動き補償の概念 [24],[25],[26] を説明する。

### 2.2.1 フレーム間相関の利用

隣接画素間の相関を利用して圧縮を行なう静止画の符号化と違って、動画像の符号化の場合は、フレーム間の相関も利用することができる。すなわち、動画像において静止している部分はフレーム内の相関よりもフレーム間の相関の方が非常に高い。従って、フレーム間の相関を利用することによって、非常に高い圧縮が可能となる。

しかし、フレーム間符号化においては、画面中に動きがある場合には逆にフレーム間の相関は低くなり、むしろフレーム内の隣接画素間の相関に比べても低くなってしまふ。つまり画像中で静止している部分は非常に良い予測が行われる反面、動いている部分での効率の劣化が激しい。従って、これを補って、平均して優れた符号化効率を実現する必要がある。

そこで、フレーム間の予測誤差分を適当な閾値と比較し、大きく予測が外れているような画素が集まっている領域(有効画素)と予測が当たっている画素の領域(無効画素)とに分離する。そして有効/無効の画素のアドレスを送ると共に、有効画素に関してはその予測誤差分を量子化して符号化を行なう。これを条件付画素補充方式と呼ぶ。

### 2.2.2 動き補償

条件付画素補充を用いる時には、画面中の有効画素を少なくすることができれば、より圧縮率を高めることが可能となる。そこで、より効率の良いフレーム間予測を行なう方法として、動き補償予測が考えられた。今単純な例として、前フレーム中の画面中央にいた人が次のフレームでは右側にそのまま平行移動した場合を考える。この場合、差分を取ると画面で有効となるのは図 2.1 (a) に示すように動いた物体の在った場所アと新たに移った場所イの両方となる。一方、1フレームの間(1/30秒)に話をしている人の顔等は、そう大きくは変化しないものと思われるので、人の顔の部分イについては前画面中の相当する部分から予測を行なうことができるはずである。つまり画面中の動いている物体に対しその動きを検出し、その動き量分だけ前の画面中で予測に用いる位置を補正してやることで、図 2.1 (b) に示すようにイの部分は予測が当たることになり、今まで隠されていた背

景の部分アのみが有効な画素となる。

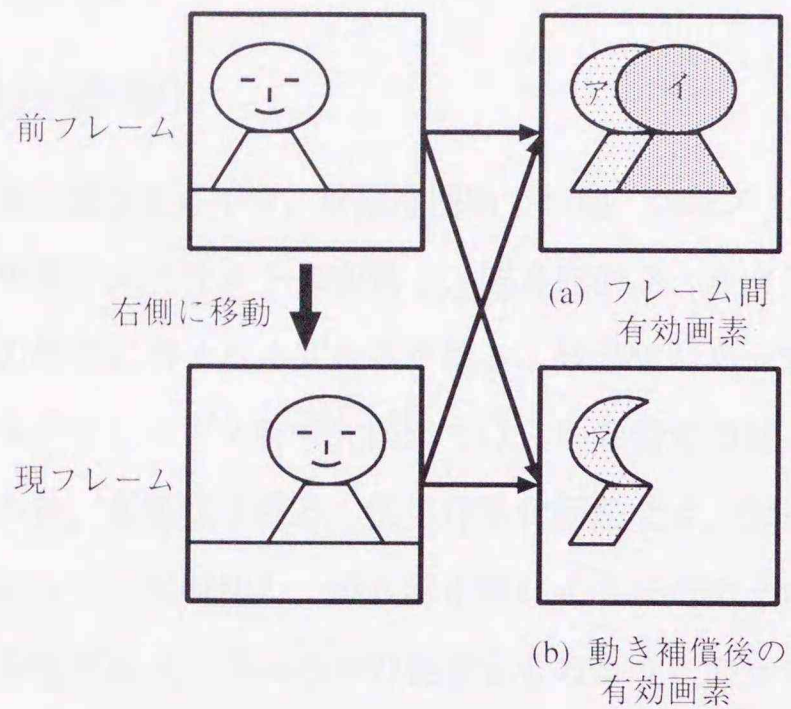


図 2.1: フレーム間符号化における有効画素

### 2.3 通信用符号化

ITU-T 勧告 H.261 は「 $p \times 64$  kb/s オーディオビジュアルサービス用ビデオ符号化方式」と題され、64 kb/s ~ 2 Mb/s の ISDN レートを用いる動画像通信用の国際標準であり、主要な用途はテレビ会議/電話である。

この映像符号化標準に対する要求条件は以下の通りである。

1. 用途は、テレビ電話 (肩上像表示)、テレビ会議 (3人横一列もしくはそれをスプリットスクリーン形式にした6人表示) とする。
2. 映像の解像度は、3人横一列の会議シーンを扱えるものとする。対話型グラフィックス (interactive graphics: カメラの前で図を描きながら説明する等) についても考慮に入れる。
3. 通信の始めに共通のパラメータを設定するための通信手順を用意する。
4. テレビ会議/電話は他の全てのオーディオビジュアルサービスと相互接続できることが望ましい。

このうち1と2は符号化勧告 H.261 で満たされ、3と4はシステム勧告 H.221/230/242/320 で対応されている。

### 2.3.1 p × 64 kb/s 標準化

符号化の全体構成図を図 2.2 に示す。前処理回路で時間・空間フィルタにより、入力ビデオ信号を世界共通の中間フォーマットに変換し、雑音除去フィルタリングを行なう。ソース符号化回路では入力信号に含まれる冗長度を除き、残差信号を一定の誤差の範囲内で量子化する。ビデオマルチプレックス符号化回路では、画像信号の他、各種ブロック属性信号を可変長符号化した後、多重化される。伝送符号化回路では、伝送バッファにより変動する情報発生を一定レートに平滑化し、誤り訂正等により伝送チャンネルに適合させる。符号化制御回路は、符号化アルゴリズムの中の要素をどのように使って与えられたチャンネルレートのビット列を発生するかを決定する。

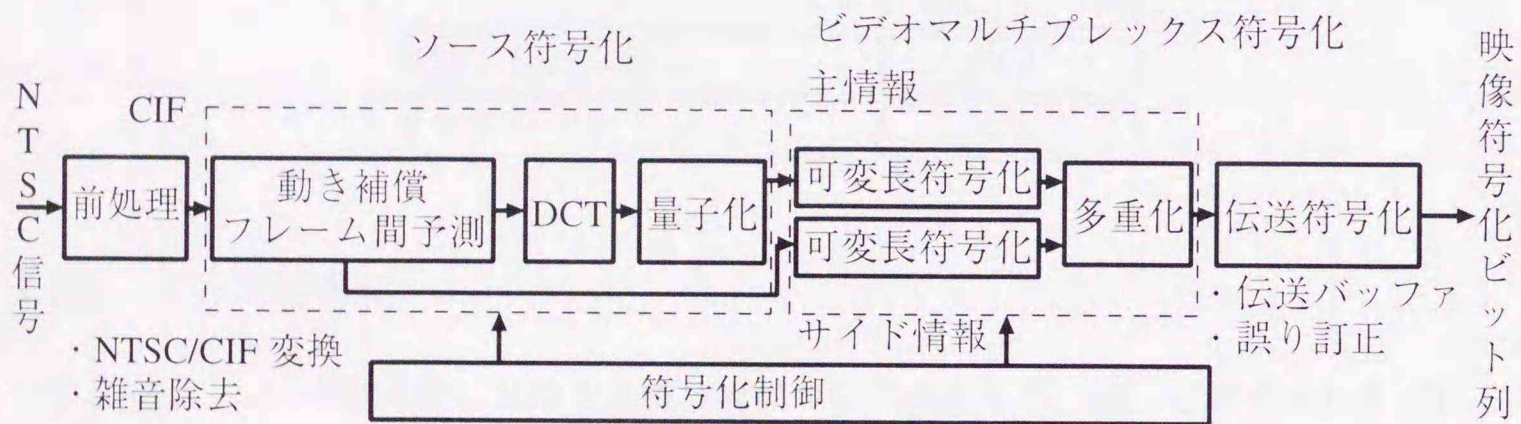


図 2.2: p × 64 kb/s 映像符号化方式

### 2.3.2 映像フォーマット

ソース符号化器への入力信号は、地域によるテレビジョン方式の違いを解決し全てのコーデック間で相手を意識することなく通信できるように図 2.3 に示す共通の中間フォーマット (CIF : Common Intermediate Format) を定め、各国はそのフォーマットに変換して送信する。

また、低ビットレート (p=1 等) での動作のために、解像度を水平・垂直ともその半分にした QCIF (Quarter CIF) を定義し、両者が対向した場合は小さい方のフォーマットにより通信することになっている。CIF, QCIF のパラメータを表 2.1 に示す。



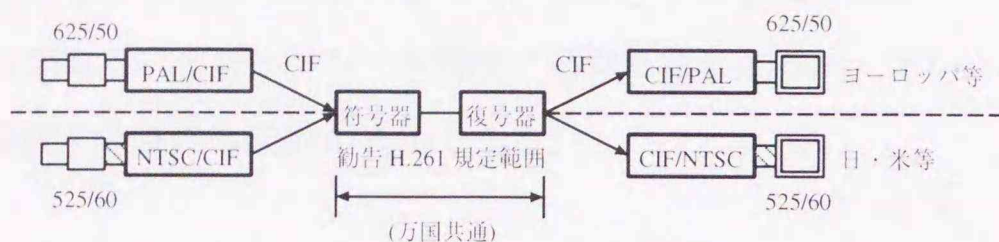


図 2.3: CIF による世界単一標準

表 2.1: 共通中間フォーマット

パラメータ		CIF	QCIF
1 ラインの画素数	Y	360	180
	Cr	180	90
	Cb	180	90
1 フレームのライン数	Y	288	144
	Cr	144	72
	Cb	144	72
毎秒フレーム数		29.97	
インタレース		なし	

### 2.3.3 ソース符号化

ここでは、ソース符号化における各種技術について概説する。尚、ビデオマルチプレックス符号化や伝送符号化については説明を省略する。

#### (1) 符号化アルゴリズム

ソース符号化アルゴリズムの構成を図 2.4 に示す。先ず動き補償を  $16 \times 16$  pels で行い、その予測誤差を  $8 \times 8$  pels のブロックサイズで DCT (Discrete Cosine Transform: 離散余弦変換) した後、量子化・可変長符号化するハイブリッド方式である。この方式によって、動き補償フレーム間予測で時間的冗長度が、DCT で空間的冗長度が除かれる。

#### (2) 予測

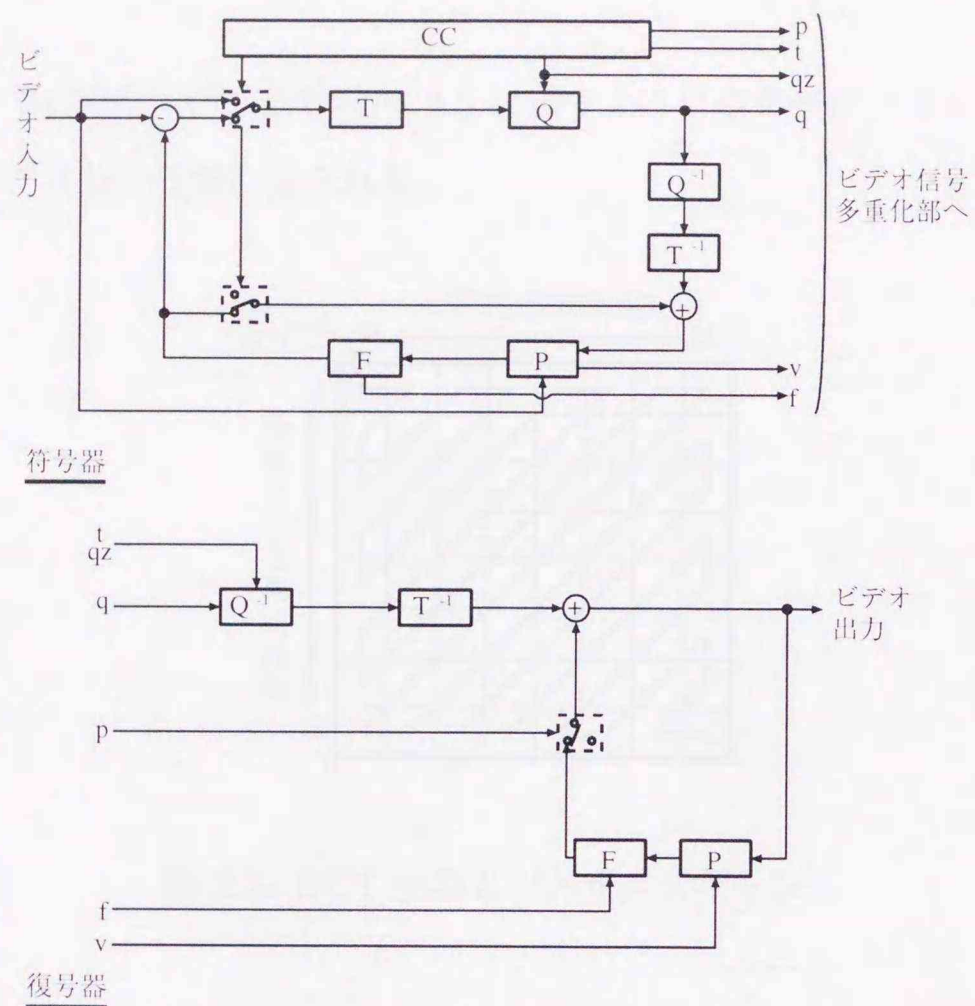
予測は通常フレーム間モードであるが、シーンチェンジ時等は入力信号が直接変換器に送られる (フレーム内モード)。フレーム間モードでは、オプションとして動き補償を用いることができ、マクロブロック毎に  $\pm 15$  pels の範囲で動き補償される。動きベクトルの

値はサイド情報として復号器に送られるので、どのような手段で動きベクトルを検出するかは設計の自由である。また、フレーム間モードで処理をするか、フレーム内モードで処理をするかを選択する基準も設計の自由である。

### (3) 変換器

画像符号化に用いる変換器の主要なパラメータは、ブロックの大きさで、 $8 \times 8$  pels または  $16 \times 16$  pels が選択の対象となる。フレーム内符号化では、一般にブロックが大きいほうが符号化効率を高めるが、フレーム間符号化では、実現上小さいブロックの方が良いこと、対象画像の局所的な性質に適応しやすいことから、通常小さいブロックで DCT が行われる。従って、勧告 H.261 では  $8 \times 8$  pels というサイズが採用されている。

DCT は、演算精度や演算方法が厳密に規定されない限り、異なる変換器では異なる変換結果が得られる。この違いがたとえ僅かなものでも、蓄積されれば、ついには目につく誤差となって表れる。勧告 H.261 では、基準となる演算方式と市販されているチップと比べた許容誤差を定義し、許容誤差範囲内の設計であれば良く、132 回符号化する間に1度はフレーム内モードを使ってリフレッシュを行なうことにより誤差の蓄積を防ぐ。



- T : 変換器
- Q : 量子化器
- P : 動き補償用可変遅延機能を持つ画像メモリ
- F : ループフィルタ
- CC : 符号化制御部
- p : フレーム間/フレーム内の識別フラグ
- t : 伝送/非伝送識別フラグ
- qz : 量子化特性指定
- q : 変換係数の量子化インデックス
- v : 動きベクトル
- f : ループ内フィルタのオン/オフ信号

図 2.4: 情報源符号器及び復号器

(4) 量子化

フレーム内モードの DC 成分には、単純な直線量子化が用いられ、フレーム内モード AC 成分、フレーム間モードの DC 成分、AC 成分には、デッドゾーン付きの直線量子化が適用される。量子化入力のどの範囲を特定の量子化インデックスに対応させるかは設計の自由である。

## (5) スキャン

量子化された64個のDCT係数は図2.5に示すようにジグザグにスキャンされ、ビデオマルチプレックス符号化部に送られる。

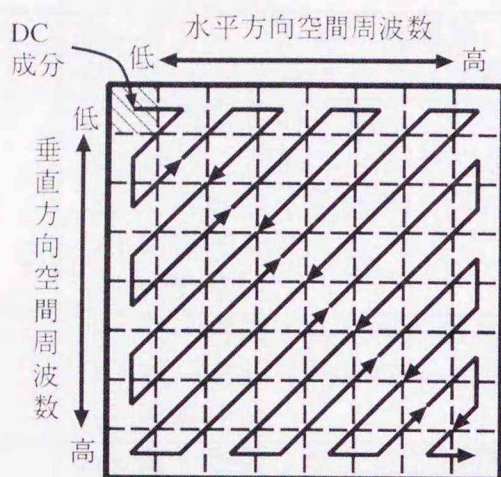


図 2.5: DCT 係数とジグザグスキャン

## (6) ループ内フィルタ

ループ内フィルタは、符号化に際して発生する雑音がフレームメモリの中に累積して再生画像を汚し、また予測効率を下げることを防止するためのもので、低ビットレートでは品質改善効果が顕著である。しかし、全てのブロックにフィルタを作用させると、逆に帯域制限した画像しか得られなくなるので、ループ内フィルタを適用するか否かは、適応的でなければならない。フィルタオン/オフ選択の基準は設計の自由に任せられている。

## 2.4 MPEG1

表 2.2: MPEG1 の対象とするアプリケーション

項目	内容
電子出版 (エンドユーザー側)	(a) 対話型の統合マルチメディア (b) 電子新聞
電子出版 (製作側)	(a) デジタルビデオ編集 (b) デジタルオーディオ編集
旅行ガイド	(a) 都市景観の下見 (b) 飛行, 運転シミュレーション (c) 公共施設 (空港等) での対話型ビデオ (d) 対話型案内 (e) 体験旅行
教育	(a) 電子書籍 (高レベルの対話機能) (b) 電子書籍 (低レベルの対話機能) (c) 遠隔授業
ビジネス	(a) 電子書籍 (カタログ販売等) (b) マルチメディア電子メール
ゲーム	(a) 個別ゲーム (b) 多地点間ゲーム (通信網を利用)
娯楽	(a) CD 上での MTV (b) TV 番組や映画の蓄積と分配 (c) 広帯域ステレオ分配 (d) オーディオとビデオの試演システム
データベース	(a) オーディオデータベース (b) データベース上 / 別の場所でのオーディオとビデオ
マルチメディアメモと掲示板	(a) 電子ニュース収集 (b) 掲示板 (c) マルチメディアメモを用いたコンピュータ会議 (d) MPEG 方式によるテレビ会議
音楽放送	(a) 製作 (b) スタジオ品質のオーディオの伝送 (c) 分配のための伝送 (d) デジタルオーディオ放送
マルチメディア芸術	
医療	(a) 教育 (b) 診断のための画像

本節で説明する蓄積用動画像符号化方式は、MPEG (Moving Picture Experts Group) と命名された。この方式は正順再生だけでなくランダムアクセスや高速再生、逆順再生のような特殊機能についても考慮されている。MPEG のうち MPEG1 (MPEG-2 は次節で説明する) が対象としている蓄積媒体は、CD (Compact Disk) や DAT (Digital Audio Tape)、ハードディスクのような連続的な転送速度が約 1.5 Mb/s 以下のものである。蓄積媒体は直接復号器に接続されるだけでなく、コンピュータのバス、LAN (Local Area Network) 等の伝送媒体を介して接続されることも想定されている。表 2.2 に MPEG1 の対象とする

アプリケーションを示す。

### 2.4.1 符号化システム

本標準案による動画符号化・復号化システム全体について、図 2.6 にその一例のブロック図を示す。

1. 前処理器：前処理器の機能は、入力を符号器の必要とするデータフォーマットに変換することである。符号器の入力フォーマットは、輝度 (Y) , 色差 (Cr,Cb) からなり、解像度については輝度信号は縦横とも最大 4096 画素で、色差信号は輝度の縦横とも半分である。画像の入力レートは、23.976 ~ 60 Hz である。
2. 符号器：入力動画像をデータ圧縮し、標準案に合ったビットストリームを作成することである。
3. システム多重符号器：符号化された複数のビデオ信号、オーディオ信号、同期信号、テキスト等の情報を、再生時に同期がとれるように多重化する。
4. 蓄積装置：蓄積装置自体は何でもよい。蓄積装置においては、ビデオ信号は符号化された状態で編集されることもある。
5. システム多重化復号器：多重化された再生ビデオ信号、オーディオ信号等を同期を取りながら分離する。
6. 復号器：標準案で決められた仕様のビットストリームから、動画像データを作り上げる。
7. 後処理器：出力ディスプレイの仕様に従ってライン補間、画像補間、レート変換、フレームフィールド変換、画素の縦横比の変換等を行なう。

### 2.4.2 符号化アルゴリズム

標準案の高能率符号化方式では、先ず時間軸方向の冗長度を落とすため動き補償を行い画像間の差分を取り、その後空間軸方向の冗長度を落とすため DCT と可変長符号を使っている。

図 2.7 に、符号器と復号器の一例のブロック図を示す。符号器は、差分器、DCT、量子化器、可変長符号器 (VLC)、バッファ、予測器、逆 DCT、逆量子化器、加算器からな

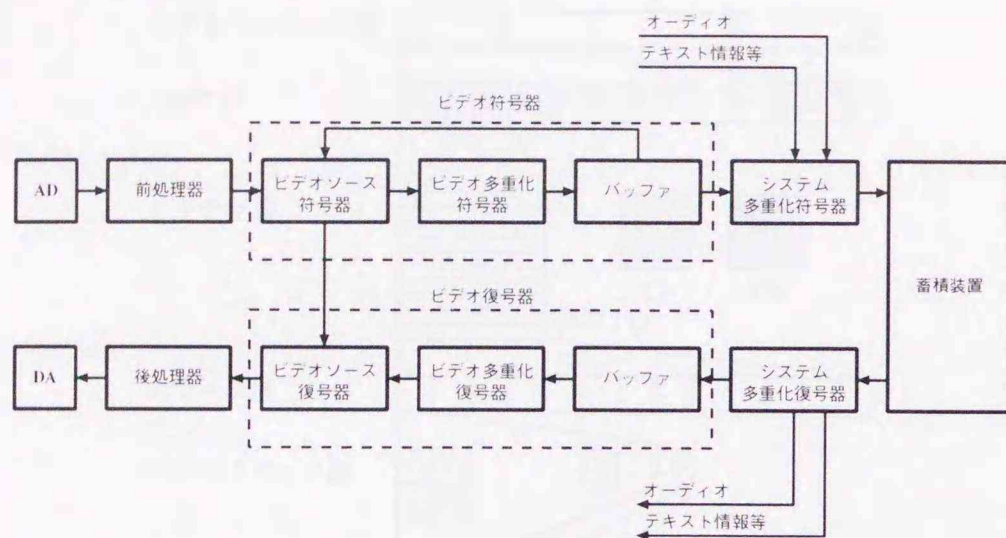


図 2.6: 動画符号化, 復号化システム

り, 動きベクトルの検出, 予測モードの判定などは予測器で行われる. このうち逆 DCT, 逆量子化器, 加算器, 予測器は, 局部復号器を構成している.

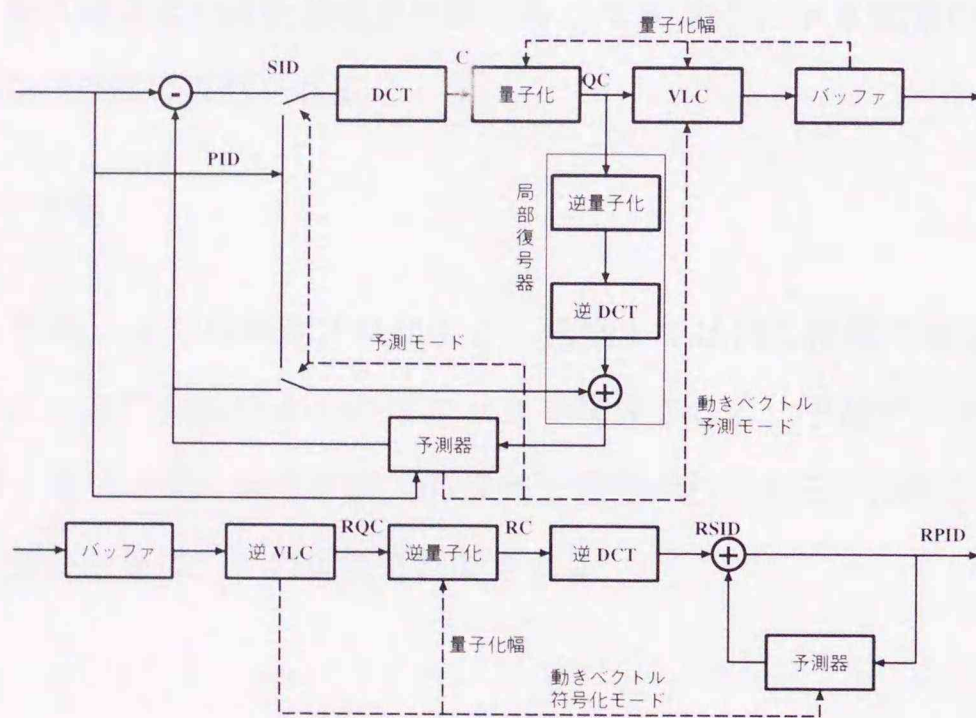


図 2.7: 符号器・復号器ブロック図

復号には, 可変長復号器, 逆 DCT, 逆量子化器, 加算器, 予測器からなる. 復号器の予測器では, 予測モードや動きベクトルはビットストリームから取り出したものを使う.

### 2.4.3 データ構造

図 2.8 に, 本標準が取り扱うデータの構造を示す. 各層の内容は下から次の通りである.

#### (1) ブロック層

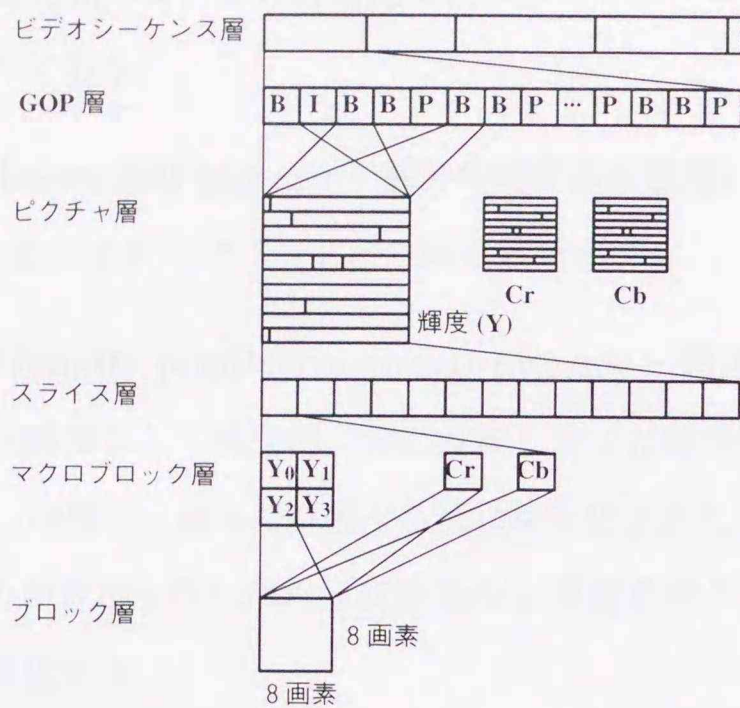


図 2.8: データ構造

ブロックは、輝度成分または色差成分の隣にあった 8 ライン × 8 画素の画素から構成される。DCT はこの単位で実行される。

(2) マクロブロック層

マクロブロックは、4つの輝度ブロックと、画像上では同じ位置に当たる Cb, Cr それぞれの色差ブロックの全部で 6つのブロックで構成される。予測データ (差分を取る基準の画像データ: 前方予測, 後方予測, 両方向予測等で作られる) に何を用いるか, 差分を送らなくても良いか等は, この単位で判断される。

(3) スライス層

画像の走査順に連なる 1つまたは複数のマクロブロックで構成される。スライスの頭では、画像内における動きベクトル及び直流成分の差分がリセットされる。最初のマクロブロックは画像内での位置を示すデータを持っており、エラーが起こった場合でも復帰できるように考えられている。

(4) ピクチャ層

**I ピクチャ (Intra-coded picture: イントラ符号化画像):** 符号化する時そのフレーム内でのみ行われる。すなわち、静止画モードである。実際には、差分を取らずにその



まま DCT して符号化する。この符号化方式はランダムアクセスや高速再生を可能とするためのものである。

**P ピクチャ (Predictive-coded picture : 前方予測符号化画像) :** 動き補償された予測画像 (I ピクチャまたは P ピクチャ) との差を符号化する。

**B ピクチャ (Bidirectionally predictive-coded picture : 両方向予測符号化画像) :** B ピクチャは、予測画像として時間的に前に位置しすでに復号化された I ピクチャまたは P ピクチャ、時間的に後ろに位置する既に復号化された I ピクチャまたは P ピクチャ、及びその両方から作られた補間画像の 3 種類を使う。この 3 種類の動き補償後の差分を符号化する。

**D ピクチャ (DC coded picture : DC イントラ符号化画像) :** DCT の DC 係数のみで構成されるイントラ符号化画像。

#### (5) GOP 層 (グループオブピクチャ層)

GOP は、1 または複数枚の I ピクチャと 0 または複数枚の非 I ピクチャから構成される。

#### (6) ビデオシーケンス層

ビデオシーケンスは、画像サイズ、画像レート等が同じ 1 または複数の GOP から構成される。

### 2.4.4 復号化方式

復号化画像は、MB (マクロブロック) タイプの判定、係数の復号化、逆量子化、逆 DCT、動きベクトルの復号化、予測ブロックとの加算、の順で再構成される。

#### 2.4.4.1 マクロブロックタイプの判定

MB (マクロブロック) の種類は、基本的に予測方式から、イントラ MB、前方予測 MB、後方予測 MB、両方向予測 MB の 4 種類に分けられる。

イントラ MB (I) フレーム内符号化のみを行なう MB

前方予測 MB (F) 動き補償フレーム間差分を時間軸方向に行った予測画像との差分を符号化する MB

後方予測 MB (B) 時間軸逆順に予測等を行なう MB

両方向予測 MB (IP) 時間順, 時間逆の双方で作った予測画像を補間 (interpolation) することによって予測画像を得, 差分を符号化する MB

I ピクチャ, P ピクチャ, B ピクチャそれぞれに存在可能な MB タイプを表 2.3 に示す.

表 2.3: MB タイプとピクチャの関係

	I ピクチャ	P ピクチャ	B ピクチャ
I : イントラ MB	○	○	○
F : 前方予測 MB	×	○	○
B : 後方予測 MB	×	×	○
IP : 両方向予測 MB	×	×	○

#### (1) I ピクチャ

I ピクチャでは MB タイプは, QS (量子化データ) 付きかなしかのイントラ MB 2 種だけである.

#### (2) P ピクチャ

P ピクチャに許される MB タイプは, QS の有無によるイントラ MB 2 種と, QS や係数の有無による前方予測 MB 5 種である.

#### (3) B ピクチャ

B ピクチャに許される MB タイプは, QS の有無によるイントラ MB 2 種と, 予測 MB 9 種 (両方向予測 MB 3 種, 前方予測 MB 3 種, 後方予測 MB 3 種) の計 11 種である.

### 2.4.4.2 係数の復号化

#### (1) CBP (Coded Block Pattern : 符号化ブロックパターン) の復号化

マクロブロック中の 6 つのブロックの 1 つでも非零の係数を持てば CBP が存在する. 符号化においては VLC を 2 進表示に変換し, MSB (Most Significant Bit) より (Cr, Cb, Y<sub>3</sub>, Y<sub>2</sub>, Y<sub>1</sub>, Y<sub>0</sub>) の順に見て 1 となるブロックに非零係数があることになる.

## (2) 係数の VLC

非零係数を持つブロックの係数の VLC は、イントラ MB と非イントラ MB とで異なるが、ピクチャタイプによる違いはない。

## (3) イントラ MB

イントラ MB の係数の VLC は、DC 係数と AC 係数では異なる。

DC 係数は、ブロック単位で伝送順に差分が取られている。Y は直前の Y の DC 係数 ( $Y_0$  の時直前の MB の  $Y_3$ ,  $Y_1$  の時  $Y_0$  等),  $C_b$  は直前の MB の  $C_b$ ,  $C_r$  は直前の MB の  $C_r$  の DC 係数同士で差分が取られる。これらの DC 係数の差分は、その値を表せるビットサイズの VLC とその値自身に分けて符号化されている。

AC 係数に関しては、図 2.5 のようにジグザグ順に伝送され、VLC として非零の係数と伝送順で前の零の数を 1 組としたものを、いわゆる 2 次符号化されたものを使用している。

## (4) 非イントラ MB

非イントラ MB では、DC 係数、AC 係数ともにイントラ MB の AC 係数のように復号化される。

### 2.4.4.3 逆量子化

係数の逆量子化は、イントラ MB と非イントラ MB とで異なるが、ピクチャタイプによる違いはない。

#### (1) イントラ MB

イントラ MB では、DC 係数とそれ以外の係数とは、逆量子化の方法は異なる。DC 係数は送られて来た量子化値に 8 を掛けて逆量子化される。AC 係数の逆量子化では係数は、送られて来た量子化値を 2 倍し、QS (量子化幅データ) と量子化マトリクスの対応位置を掛けた後、更に 16 で割って切り捨てて得られる。尚結果が偶数となるものは 1 増減して 0 に近い奇数とする。また 2047 より大きいものは 2047 に、-2048 より小さいものは -2048 に制限する。

また、量子化マトリクスには、表 2.4 に示すようにデフォルト値が決められている。この量子化マトリクスは、ビデオシーケンスヘッダからロードし変更することも可能である (IQM 使用)。但し次のシーケンススタートコード (SSC) でデフォルトマトリクスに戻る。符号器側での量子化方法は、AC 係数をまず 16 倍し、量子化マトリクス値及び QS の 2 倍でそれぞれ割り算し四捨五入する。尚量子化結果の絶対値が 255 を超える場合は 255 に制限する。

表 2.4: 量子化マトリクス

輝 度 信 号	8	16	19	22	26	27	29	34
	16	16	22	24	27	29	34	37
	19	22	26	27	29	34	34	38
	22	22	26	27	29	34	37	40
	22	26	27	29	32	35	40	48
	26	27	29	32	35	40	48	58
	26	27	29	34	38	46	56	69
	27	29	35	38	46	56	69	83
色 差 信 号	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16
	16	16	16	16	16	16	16	16

## (2) 非イントラ MB

非イントラ MB とイントラ MB とで次の 3 点で異なる。

- イントラ MB では、DC 係数の前ブロックとの差分を VLC するが、非イントラ MB では、DC も AC も同じ扱い。
- 量子化マトリクスのデフォルト値が異なる。
- 非イントラ MB にはデッドゾーン (QS/2) がある。

非イントラ MB の量子化マトリクスデフォルト値をやはり表 2.4 に示す。表 2.4 の量子化マトリクスは、やはりビデオシーケンスヘッダからロードし変更することが可能である。但し次のシーケンススタートコード (SSC) でデフォルトマトリクスに戻る。

#### 2.4.4.4 逆 DCT

逆 DCT は全てブロックに対して同じでその仕様は決められている。

#### 2.4.4.5 動きベクトルの復号化

動きベクトルの復号化方法は、マクロブロック毎ではなく、ピクチャタイプ毎に決められている。また I ピクチャ及びイントラ MB には動きベクトルはない。

##### (1) P ピクチャ

前方予測 MB の動きベクトルは (水平成分, 垂直成分) の 1 組で,  $Y_0, Y_1, Y_2, Y_3$  については同じ Y 用動きベクトルを Cr, Cb については Y 用動きベクトルをそれぞれ 2 で割って切り捨てた同じ C 用動きベクトルを使う。動きベクトルが送られない前方予測 MB では、動きベクトルは 0 である。MB が動きベクトルを持つ時、その動きベクトルは、伝送順に MB 単位に水平, 垂直成分ともそれぞれ差分が取られ符号化される。但しスライスの頭及び直前の MB がイントラ MB の時、それぞれ 0 からの差分が符号化される。そのベクトル値の差分は、ピクチャヘッダの動き補償の範囲を示すパラメータ (FF : 1 ~ 6 のいずれか) に従い、それぞれ復号化される。注意すべきは動きベクトルは時間的に前の基準画像との間で作られるが、符号化に際しての差分は、同一画像内の伝送順位が前の同じ性質の MB との間で計算されることである。尚ピクチャヘッダでベクトル精度がハーフペル (1/2 画素) 単位の場合の指示がある。

##### (2) B ピクチャ

B ピクチャの非イントラ MB では、P ピクチャと違い、1 つまたは 2 つの動きベクトルが存在し、前方動きベクトルしかなければ前方画像からだけの予測 (前方予測 MB) となり、後方動きベクトルしかなければ後方画像からだけの (後方予測 MB)、両方あれば前・後方両画像を使った予測 (両方向予測 MB) となる。前方動きベクトル、後方動きベクトルは、それぞれ MB で 1 組で、 $Y_0, Y_1, Y_2, Y_3$  については同じ前方動きベクトル、後方

動きベクトルを、Cb, Cr の両方に対しては、やはり Y 用動きベクトルを 2 で割った 1 つの前方動きベクトル、後方動きベクトルを使う。それぞれの動きベクトルは、独立に P ピクチャ同様伝送順に MB 単位で差分が取られ符号化される。尚、P ピクチャと同じくベクトル精度を示すパラメータがある。

#### 2.4.4.6 予測ブロック

予測ブロックの作り方は、MB タイプによって異なるが、ピクチャタイプによっては変わらない。また、イントラ MB は予測ブロックを持たない。

##### (1) 前方予測 MB

前方予測 MB では、復号化された動きベクトルを使って、前方復号化画像から予測ブロックを作る。動きベクトルが整数の場合は、動きベクトルに  $1/2$  画素の端数がある時には、最も近い整数 (大・小 2 つある) を動きベクトルとして取ってきた 2 つのブロックを補間 (加算して平均) して予測ブロックとする。  $1/2$  画素の端数が水平方向のみ、垂直方向のみのどちらかであれば 2 つのブロックの補間であれば良いが、双方とも時は 4 つのブロックを補間する必要がある。

##### (2) 後方予測 MB

後方予測 MB でも前方予測 MB と同様に動きベクトルを用いて後方復号化画像から予測ブロックを求める。

##### (3) 両方向予測 MB

両方向予測 MB では前方動きベクトルから前方予測ブロックを作り、後方動きベクトルから後方予測ブロックを作り、この両者を足して 2 で割り両方向予測ブロックを求める。

復号器側では前方復号化画像及び後方復号化画像より、適当な動きベクトル探索法により前方動きベクトルと後方動きベクトルを求める。符号器・局部復号器に対する予測ブロックは、今求めた前方・後方動きベクトルを入力すれば、上述の方法で与えられることになる。

#### 2.4.4.7 復号化画像ブロック

復号化画像ブロックの作り方は、イントラ MB と非イントラ MB とで異なるが、ピクチャタイプによっては変わらない。

## (1) イントラ MB

イントラ MB では、最終的な復号化画像ブロックは、復号化 DCT 係数からそのまま逆 DCT を行えば良く、このとき画素値を  $[0,255]$  の範囲に制限する。

## (2) 非イントラ MB

最終的な復号化ブロックは、前項の予測ブロックに、逆 DCT で与えられた画素値を加算すれば良い。やはり結果を  $[0,255]$  の範囲に制限する必要がある。符号器側では、予測ブロックが求めれば、今符号化すべき画像ブロックとの差分を求め、これを DCT により符号化して送信することになる。

## 2.5 MPEG2

表 2.5: MPEG2 のアプリケーション分野

略称	アプリケーション
BSS	Broadcasting Satellite Service 放送のサテライトサービス
CATV	Cable TV Distribution on optical fiber, copper, etc 光ファイバ, 銅線等によるケーブルテレビ放送
CDAD	Cable Digital Audio Distribution ケーブルデジタルオーディオ分配
DAB	Digital Audio Broadcasting デジタルオーディオ放送
DTTB	Digital Terrestrial Television Broadcast デジタル地上テレビ放送
EC	Electric Cinema 電子映画
ENG	Electronic News Gathering (including SNG ; Satellite News Gathering) 電子ニュース収集 (衛星ニュース収集を含む)
HTT	Home Television Theatre 家庭テレビ劇場
IPC	Inter Personal Communications (video conferencing, videophone, etc) 個人間通信 (テレビ会議, テレビ電話等)
ISM	Interactive Storage Media インタラクティブ蓄積メディア
MMM	Multimedia Mail マルチメディアメール
NCA	News and Current Affairs ニュースと時事
NDB	Networked Database Services (via ATM, etc) ネットワークデータベースサービス (ATM による)
RVS	Remote Video Surveillance 遠隔監視ビデオ
SSM	Serial Storage Media (digital VCR, etc)

MPEG2 標準規格草案は, 表 2.5 に示すようにデジタル蓄積メディアだけでなく放送や通信の分野のアプリケーションに対して, 汎用的に用いることを目的としている. 対象とする画像は主にインタレース画像である. NTSC 等の TV 画像を 3 ~ 10 Mb/s 前後で, HDTV を 15 ~ 20 Mb/s 程度で符号化することを目標にしている.

MPEG1 で制定された蓄積メディアに適するパッケージストリームだけでなく, 放送に適するトランスポートストリームを規定し, 複数の符号化アルゴリズムを整理するために, 符号化機能に基づく“プロファイル”と, パラメタサイズに起因する“レベル”という概念を導入して, 広範囲なアプリケーションへの適用を可能としている. MPEG1 と MPEG2 との主な違いについては表 2.6 に示す.



表 2.6: MPEG1 と MPEG2 との主な違い

項目	MPEG1	MPEG2
画像フォーマット	順次走査 (ノンインタレース) のみ 色差フォーマットは 4:2:0 のみ	飛び越し走査 (インタレース) もサポート 色差フォーマットは全てをサポート
画像品質	放送品質には及ばない	HDTV のような高品質画像
符号化	フレーム構造のみ	フレーム構造とフィールド構造の両方をサポート
ビット・ストリーム・スケーラビリティ	なし	空間スケーラビリティ 時間スケーラビリティ SNR スケーラビリティ

### 2.5.1 標準アルゴリズム

#### 2.5.1.1 プロファイル (profile) とレベル (level)

MPEG1 が順次走査 (progressive) 画像を符号化の対象としていたのに対し、MPEG2 では順次走査画像だけではなく、インタレース画像の符号化効率を向上させることに主眼が置かれている。多くのアプリケーションに適用できるように、符号化に必要なパラメータは、固定せず変数としてヘッダに記述される。デコーダではパラメータを解読してから処理を開始する。画像フォーマットに関する可変パラメータ及び、符号化モードに関するパラメータを表 2.7 に示す。パラメータの値は、“プロファイル”と呼ばれる機能セット毎に特定の値に定められる。また、MPEG2 は多様なアプリケーションへ適用するため、多くのツール (要求条件を満たす機能) からなる。しかし、全てのツールを含有した装置を作ることは当面現実的でない。そこで、限られた数のツールを持つサブセットを“プロファイル”として定義する。同一プロファイル内で異なるパラメータ値を識別する必要がある場合は、“レベル”として定義する。

#### 2.5.1.2 符号化アルゴリズム

##### (1) シーケンスヘッダと GOB 構造

MPEG2 ではシーケンスヘッダを 1 つのプログラムの中で繰り返し使うことができる。これは放送等のアプリケーションで必要な、任意の時刻に復号化を開始する処理を可能に

表 2.7: プロファイル・レベルに関わるパラメータ

分類	パラメータ
画像フォーマット	符号化画素数 (縦横画素数は 14 bit × 14 bit の整数) 画素アスペクト比 (4 bit, 0.6735 ~ 1.2015) 符号化フレームレート (4 bit, 23.936 ~ 60 Hz) 順次走査画像/インタレース画像 表示領域 (画素数, 中心位置情報) 色差形式 (4:4:4, 4:2:2, 4:2:0)
符号化モード	符号化レート (上限約 100 Mb/s) 符号化バッファサイズ (符号化レートの約 1/8) フレーム構造/フィールド構造 (ピクチャの単位毎に設定) フレーム構造での動き補償 (16 × 16 フレーム予測/16 × 8 フィールド予測/dual-prime) フィールド構造での動き補償 (16 × 16 フィールド予測/16 × 8 フィールド予測/dual-prime) I ピクチャ間隔, P ピクチャ間隔, B ピクチャの有無, スライス長の設定, インTRASライス間隔 量子化マトリクスの変更 (ピクチャ毎にダウンロード可能) イントラ用 VLC, 9, 10 bit DC 係数, ジグザグ以外の係数スキャン

するためである。GOP 構造は MPEG1 と同じく I ピクチャで始まり、P ピクチャ、B ピクチャからなる。

## (2) フレーム構造とフィールド構造

MPEG2 では画像の符号化の単位をフレームだけでなく、フィールド単位にも設定することができる。ピクチャコーディングエクステンションに含まれる“picture.structure”がフレーム/奇フィールド (odd field)/偶フィールド (even field) の区別を示す。フレームの状態のままマクロブロックを構成し、動き補償や DCT を行なうフレーム構造の符号化では、フレームピクチャを指定する必要がある。前方予測の場合のフレーム構造における参照画像はインタレースされた前フレームの画像である。

それに対しフィールドの画素からなるマクロブロックを用いるフィールド構造の符号化では、フィールドピクチャが指定される。前方予測の場合のフィールド構造における参照画像は最も近い異なった 2 つのフィールドであり、符号化の対象となっているフィールドが奇フィールドであるか、偶フィールドであるかにより取るべき参照フィールドが異なる。フィールド構造の符号化では、奇フィールドが P ピクチャあるいは B ピクチャの時、偶フィールドでは同じ予測方式を取るよう制限されている。但し、奇フィールドが I ピク

チャの時は、偶フィールドは P ピクチャとすることが許されている。

### (3) フレーム/フィールド適応動き補償

フレーム構造の符号化では、インタレース信号対策としてフィールド単位の動き補償とフレーム単位の動き補償をマクロブロック単位に切り換えることができる。フィールド構造の符号化では、 $16 \times 16$  画素のマクロブロックに対して、 $16 \times 16$  画素を単位として 1 ベクトル、あるいは上下  $16 \times 8$  画素の領域を単位として 2 ベクトルを用いて動き補償される。また、“dual-prime” というインタレース画像用の特殊な動き補償をマクロブロック単位に切り換えて用いることもできる。

dual-prime 予測方式が B ピクチャなしの符号化モードに対して使用可能である。これは、インタレース構造を利用して、動きベクトルの内、外挿と平均化により、予測効率を向上させる技術である。同パリティ (奇なら奇、偶なら偶) のフィールド間ベクトルによる動き補償を予測の一候補とする。次にフィールド距離に応じて 0.5 あるいは 1.5 倍したベクトルを  $[-1,1]$  の範囲で修正 (誤差最小化) したベクトルによる動き補償を予測の別候補とする。これら 2 つの候補の予測画像の平均値により予測を行なう。平均操作が必要であるため、画像メモリへの早いアクセスが必要であり、平均モードを用いる B ピクチャと同じになる。この dual-prime 予測はインタレース画像の場合に問題であった、ゆっくりした垂直方向の動き補償に効果を発揮する。また平均処理による予測はループ内フィルタと同様の効果がある。

フィールド予測では  $16 \times 16$  画素のブロックサイズと、それを上下に分割した  $16 \times 8$  画素のサブブロック単位に細かな予測を行なう動き補償とを用いることができる。またフレーム構造と同様に、dual-prime も用いることができる。

### (4) フレーム/フィールド適応 DCT と量子化

フレーム/フィールド適応 DCT はマクロブロック内のデータに対して、交互に混ぜてフレーム構成として DCT を行なうモードと、フィールド別に DCT を行なうモードとを切り換える技術である。

フレームピクチャかつ符号化ブロックであれば、マクロブロック毎に DCT のモードを選択できる。ビットストリームシンタックス上の “dct.type” を “1” に設定すればフィールド DCT となり、“0” に設定すればフレーム DCT となる。フィールド/フレーム適応動き補償とフィールド/フレーム適応 DCT とは独立に設定できる。

DCT 係数の量子化ステップは H.261, MPEG1 にみられる等間隔のものに加え, 0.5 刻みの量子化ステップが用意された。H.261, MPEG1 の DCT-AC 係数の可変長符号 (VLC) は EOB が 2 ビットであり, イントラ用としては無駄が多く, EOB が 4 ビットの VLC テーブルが準備されている。フィールド構造では横長の画面となる点と, インタレース構造のために, ジグザグスキャンは必ずしも有効ではない。縦方向に優先的に係数をたどる new scan が定義された。いずれも, 高い符号化レートでの I ピクチャの符号化に有効である。

### 2.5.2 スケーラビリティ機能

スケーラビリティ機能は, MPEG2 で新しく導入されたものである。

#### 2.5.2.1 空間スケーラビリティ

空間解像度が小さいレイヤを基本レイヤ, 大きいレイヤを高位レイヤと呼ぶことにする。基本レイヤは, 通常の MPEG2 の符号化法で符号化される。これに対し高位レイヤでは, 基本レイヤの画像を画素間を内挿したりしてアップ・サンプルして高位レイヤと同じ大きさの画像を作り出し, 高位レイヤ中の画像からの予測だけでなく, このアップ・サンプルされた画像からも予測することにより, より効率の良い符号化をすることができるようになる。

空間スケーラビリティを使うと, 2 つの動画像データを別々に MPEG2 で符号化して送るよりも, 効率良く 2 つの動画像データの符号化ができる。また, 通常のテレビ放送と HDTV 放送を同時に送り, 受信端末の性能に応じて画像を選ぶこともできる。

#### 2.5.2.2 時間スケーラビリティ

空間解像度は同じであるが, 1 秒当たりのフレーム数が違うもの同士を効率良く送るために, 時間スケーラビリティを用いる。時間解像度が小さいレイヤを基本レイヤ, 大きいレイヤを高位レイヤと呼ぶ。基本レイヤは, 通常の MPEG2 の符号化法で符号化される。これに対し高位レイヤでは, 高位レイヤ中で I, P, B ピクチャを使うだけでなく, 基本レイヤの画像を使って予測できる。

### 2.5.2.3 SNR スケーラビリティ

空間解像度も時間解像度も同じであるが、画質の違った2つの動画像データを同時に効率良く送るために、SNR スケーラビリティが用いられる。画質が低いレイヤを基本レイヤ、画質が高いレイヤを高位レイヤと呼ぶ。基本レイヤは、MPEG1 か MPEG2 で符号化される。これに対し高位レイヤでは、基本レイヤで符号化されたデータを復元し、元のデータから復元画像を引き、この誤差分を基本レイヤの量子化ステップサイズより小さなステップサイズで量子化して符号化する。

### 2.5.3 復号化アルゴリズム

復号化処理は、基本的には MPEG1 と同様な処理を行なう。但し、MPEG2 に固有の処理を行なうために以下のことを識別する必要がある。

- プロファイルとレベル
- フレーム構造とフィールド構造
- フレーム予測とフィールド予測
- 予測モード
- DCT モード
- スキャン
- 色差フォーマット
- イントラ用 VLC

MPEG1 に比べて予測モードが増えたために、動きベクトルの予測値の設定が複雑化している。

## 2.6 MPEG4

### 2.6.1 MPEG4 の概要と特徴

前節までに、蓄積型 DSM(Digital Storage Media), CD-ROM を対象に「1時間の映画を CD-ROM に入れる」ことを目標に掲げ、実時間エンコードではない符号化方式の確立を目指して活動が開始された。従来の通信分野で用いられて来た実時間処理符号化方式を離れたことで、通常「B フレーム」と呼ばれる双方向動き予測技術が導入されて動き補償予測の精度が著しく改善された。また、離散コサイン変換方式 DCT, さらに「GOP(Group of Picture)」という、蓄積系での高速アクセス機能を意識したストリーム構造などの特徴もあり、その後の動画像符号化の基本技術が確立した。

MPEG-1 では、CD-ROM ベースから来る伝送レート 1.5Mbps により、具体的製品応用の「Video CD」や「カラオケ」として、広く使用され社会に定着している。

MPEG-1 での成功を基に、より高画質を実現し、放送分野での応用を主目的に、より広範囲な応用を目指す動きが始まり、MPEG-2 標準化が始められた。放送用の画質を得ることが目的であり、そのために新たなビットレートを設定し、ビットレートを高くしてインターレースにも対応できるようにしたり、動きの速い画面で高画質を得ると同時に、静止画像でも高解像度を得るためのフィールド/フレーム適応符号化を採用し、画像の性質に応じて VLC テーブルを切替える等の工夫が行なわれた結果、著しく画質が向上し、放送画質もクリアした。

MPEG4 は、携帯端末、携帯用 TV 電話の可能性を示唆し、その目的の符号化方式の標準化としてスタートした。その後、単に超低ビットレート符号化機能のみでなく、インターネットやパソコン上でのマルチメディアデータを自由に扱うための汎用符号化標準として進められた。MPEG-1, MPEG-2 および H.263 に使われている技術が基本になっているが、大きく異なっているのは、オブジェクトを重ね合わせて1つのシーンを構成できるよう、コンポジションの仕様をシステムパートに定めたことと、各オブジェクトの映像データが、そのオブジェクト形状および透明性の情報を持っていることである。この情報は、形状情報と呼ばれ、従来の輝度や色差情報はテクスチャ情報と呼ばれている。形状情報の符号化は、従来に無い機能であり、そのために各種の高効率符号化が研究されている。

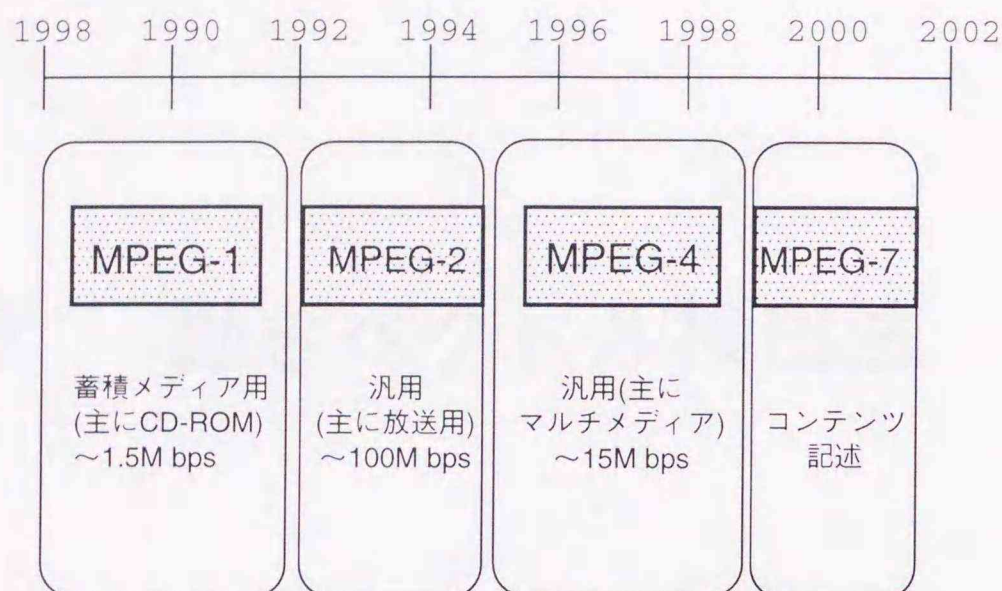


図 2.9: MPEG 標準化の流れ

## 2.7 まとめ

本章では、動画像の符号化を ITU-T, MPEG 等の機関による標準化を中心に概説した。動画像の符号化のための各種技術を開発していく上で、標準化は様々な場所間の通信のために考慮に入れるべきものである。

MPEG はオーディオ・ビジュアル信号の符号化技術の国際標準化方式からマルチメディア統合の標準符号化技術、システム技術へと進展してきた。その結果、ビデオ CD の標準符号化方式から、放送・通信システム、蓄積システムにおけるオーディオ・ビジュアル情報の符号化標準へと幅広い分野でその重要性が増しつつある。画像符号化技術について着目してみると、MPEG-1 では動き補償予測と直交変換方式を中心として、両方向予測という新しい技術を実現した。MPEG-2 では、MPEG-1 を基礎として、インタレース対応のフィールド/フレーム適応動き補償予測、フィールド/フレーム適応直交変換技術などを標準化することによって、高画質なデジタル放送、DVD などを実現した。そして MPEG-4 ではオブジェクト符号化等の技術によって、低ビットレートでの高画質化、様々なマルチメディア・インタラクティブシステムでの映像情報の効率的な取り扱いが可能となってきた。

画像符号化技術の世代区分からみると MPEG も、分析合成符号化、認識合成符号化、知的符号化とまだまだ発展する余地があるといえる。この発展を実現することによって、マルチメディアにおける画像・映像情報の性質、利用目的、内容に対応した最適な符号化方式を選択することができ、もっとも効率的な情報伝達・蓄積が可能になる。

## 第3章 動きベクトル検出法

### 3.1 はじめに

動画データの圧縮のためにはまず、時間方向の冗長度の削減が重要である。従って、2枚の連続する画像において、高精度で動き情報を抽出することは全体の符号化効率に大きな影響を及ぼす。動き情報がどのように抽出されるかで、その後の動き補償予測誤差信号の性質が異なる。このことにより、予測誤差信号の高周波領域及び低周波領域へのビットの割り当て方が変わって来るので、画質にも影響を及ぼす。よって、動画データの圧縮のための動きベクトル検出法が色々検討されている。

計算の複雑さと実時間処理のために、主に動きの平行移動成分のみで記述する動きモデルが現在一般的に研究されている。すなわち、物体の動きを並進運動のみであると仮定し、その動きベクトルを検出するということである。この動きベクトルの検出であるが、主に勾配法に基づく方法とブロック・マッチング法に基づく方法の2つに大きく分けられる。本章では、動きベクトルの検出法を概説する。その中でも特に有名な3段階探索ブロック・マッチング法 [27] と、その改良版である可変段階探索法 [28] とを重点的に説明する。尚、現在の動き補償方式においては、ブロック毎に動き検出を行い、ブロック毎に動き補償を行なうので、ブロック歪が生じる。このブロック歪は動き補償予測誤差信号の高周波成分を増加させ、高能率な符号化を困難にしていることに注意する。

### 3.2 勾配法

テレビジョン信号において、動いている物体はフレーム間の輝度変化をもたらす。この輝度変化は物体の動きを記述する数学的なモデルのパラメータを検出するために用いることができる。動きモデルは予測や内挿等のテレビジョン符号化技術の効率を改善するため



に用いられる。実時間処理を行なうために、動きの平行成分のみを考慮する比較的簡単なモデルがテレビジョン符号化に対し研究されている。テレビジョン符号化に対する動きベクトル検出アルゴリズムは最初に1975年にLimbとMurphy [29]によって、1976年にCafforioとRocca [30]によって提案されている。

LimbとMurphyのアルゴリズム [29] は図3.1に示されるような簡単な動いているエッジを考慮する。変位  $D$  は次のように検出される。

$$\hat{D} = \hat{dx} = \frac{\sum_M |FD|}{\sum_M |ED|} \quad (3.1)$$

ここで  $|FD|$  はフレーム差分信号の大きさであり  $|ED|$  は画素差分信号の大きさを表す。与えられた閾値よりも大きいフレーム差分によって定義される領域  $M$  にわたって和が求められる。分子

$$\sum_M |FD|$$

は斜線の領域に対応し、分母

$$\sum_M |ED|$$

は平行四辺形の高さに相当する。式 (3.1) は変位の方向は示していない。CafforioとRoccaは異なったアプローチで始め、変位の検出に対する彼らのアルゴリズムはLimbとMurphyのアルゴリズムを特別に単純化された解として含んでいることを示した [29]。それによって後者は実現が容易になった。

$s_k(x, y)$  をフレーム  $k$  の動いている物体の点  $(x, y)$  における輝度値とする。動いている物体はフレーム間で輝度は変化しないと仮定する。動きベクトル  $D$  が成分  $(dx, dy)$  を持つ純粋な平行移動である場合、フレーム差分  $FD$  は

$$\begin{aligned} FD(x, y) &= s_k(x, y) - s_{k-1}(x, y) \\ &= s_k(x, y) - s_k(x + dx, y + dy) \\ &= -\frac{\delta s_k(x, y)}{\delta x} \cdot dx - \frac{\delta s_k(x, y)}{\delta y} \cdot dy - n(x, y) \\ &= -D^T \nabla s_k(x, y) - n(x, y) \end{aligned} \quad (3.2)$$

ここで  $n(x, y)$  は無視されるテーラー級数展開の高次項を表す。勾配  $\nabla s$  はその要素が画素差分  $ED$  とライン差分  $FD$  のベクトルである。

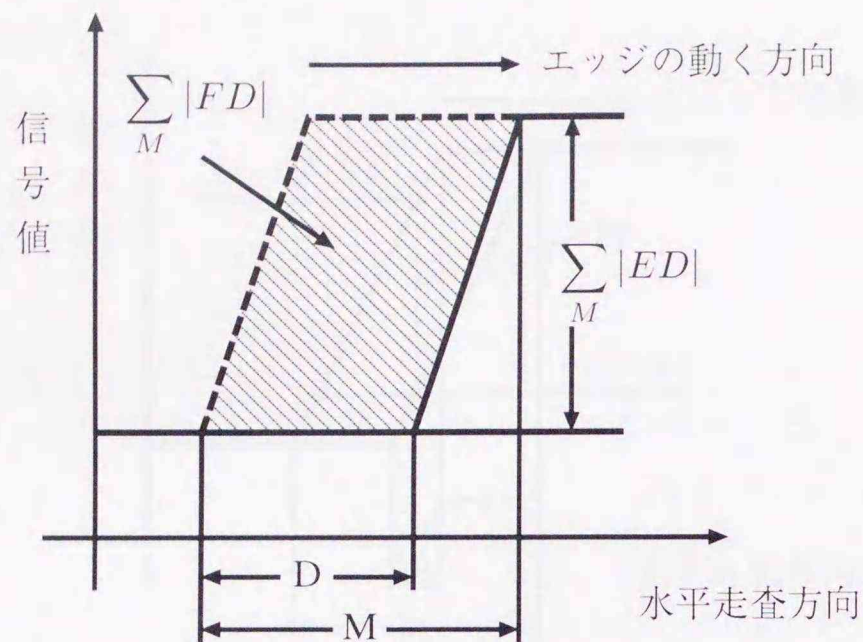


図 3.1: Limb と Murphy の検出法

(3.2) を説明するために、 $x$  方向における 1 次元の変位を図 3.2 で考慮する。この場合、動きベクトル  $D$  の検出は以下ようになる。

$$\hat{D} = \hat{dx} = -\frac{FD(x, y)}{\partial s(x, y)/\partial x} \quad (3.3)$$

動いている物体の境界が分かっている場合、(3.2) は動いている物体の全ての画素に対して評価することができる。線形回帰を用い、 $x, y$  交差項を無視すると、動きベクトル  $D$  は以下のように近似される。

$$\hat{dx} = -\frac{E[FD(x, y) \cdot \partial s(x, y)/\partial x]}{E[(\partial s(x, y)/\partial x)^2]} = -\frac{\sum(FD \cdot ED)}{\sum(ED)^2} \quad (3.4)$$

$$\hat{dy} = -\frac{E[FD(x, y) \cdot \partial s(x, y)/\partial y]}{E[(\partial s(x, y)/\partial y)^2]} = -\frac{\sum(FD \cdot LD)}{\sum(LD)^2} \quad (3.5)$$

但し、 $ED$  は  $x$  方向画素差分、 $LD$  は  $y$  方向画素差分である。ここで統計的な平均は動いている物体の全体の領域  $M$  にわたって合計することによって計算される。

Cafforio と Rocca の動きの検出アルゴリズム [30] は点  $(x, y)$  における線形な輝度関数を仮定している。例えば、図 3.2 でこの仮定は領域内  $M_1$  のみで有効である。この領域は変位が増加すると減少する。それゆえに、この検出アルゴリズムは小さな変位  $D$  を測定するために用いられる。

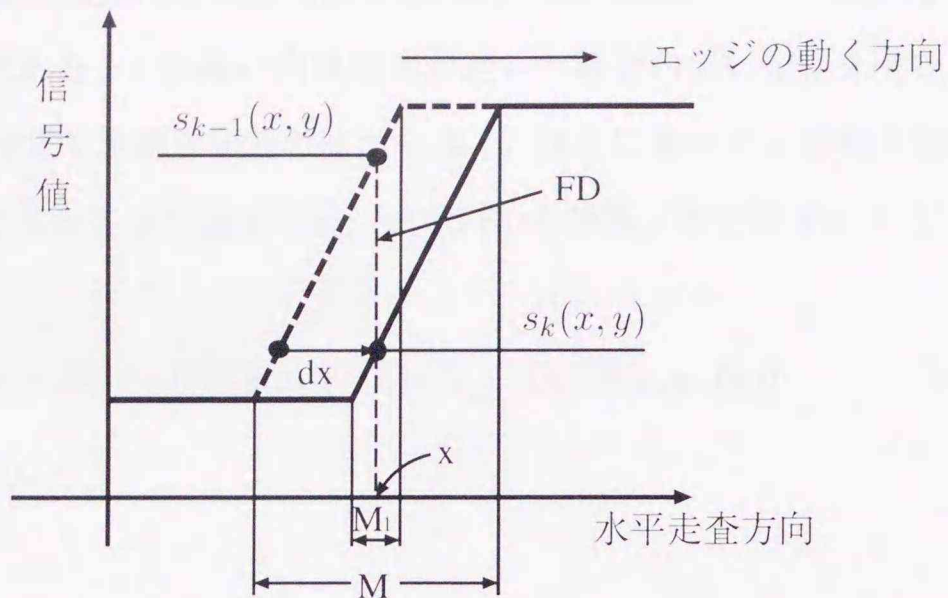


図 3.2: Cafforio と Rocca の検出法

### 3.3 勾配法：画素反復法

1978年に、Netravali と Robbins は検出精度を改善し、 $D$  の測定範囲を増加させるために最初の画素反復アルゴリズムを発表した [31]。画素反復アルゴリズムにおいて最初の変位  $\hat{D}_i$  は新しい改善された変位  $\hat{D}_{i+1}$  を生み出すために用いられる。それは以下の式のようになされる。

$$\hat{D}_{i+1} = \hat{D}_i + U_i \quad (3.6)$$

ここで  $U_i$  は繰り返し  $i$  における更新項である。繰り返しは走査ラインに沿って連続的な画素の中の1つの画素に対して、ラインからラインへ、あるいはフレームからフレームのどちらかで実行される。 $\hat{D}_i$  を知ると、変位フレーム差分  $DFD$  という関数

$$DFD(x, y, \hat{D}_i) = s_k(x, y) - s_{k-1}(x - \hat{d}x_i, y - \hat{d}y_i) \quad (3.7)$$

は変位  $\hat{D}_{i+1}$  を計算するための基準として使われる。Netravali と Robbins は勾配法を用いて  $i$  で再帰的に変位フレーム差分の自乗値を最小にしようとする検出アルゴリズムを提案した [31]。

$$\hat{D}_{i+1} = \hat{D}_i - \frac{1}{2} \varepsilon \nabla_{\hat{D}_i} [DFD(x, y, \hat{D}_i)]^2 \quad (3.8)$$

ここで  $\nabla_{\hat{D}_i}$  は  $\hat{D}_i$  に関する勾配オペレータであり、 $\varepsilon$  は正の定数である。図 3.3 はこのア

ルゴリズムが何回かの繰り返しの後実際の変位  $D$  にどのように近づくかを説明している。 $\varepsilon$  の選択は妥協が必要である。 $\varepsilon$  の高い値は収束が速いが雑音の多い変位を生む。反対に小さな  $\varepsilon$  は収束が遅いがより正確な変位の検出を導く。[31] において  $\varepsilon$  は動き補償予測符号化に対して  $1/1024$  であるように選ばれる。式 (3.8) を評価すると以下のようなになる。

$$\hat{D}_{i+1} = \hat{D}_i - \varepsilon DFD(x, y, \hat{D}_i) \cdot \nabla_{\hat{D}_i} [DFD(x, y, \hat{D}_i)] \quad (3.9)$$

そして定義式 (3.7) を用いて、

$$\hat{D}_{i+1} = \hat{D}_i - \varepsilon DFD(x, y, \hat{D}_i) \cdot \nabla s_{k-1}(x - \hat{d}x_i, y - \hat{d}y_i) \quad (3.10)$$

ここで  $\nabla$  は水平座標, 垂直座標  $x, y$  に関する勾配オペレータである。

$$\nabla s_{k-1}(x - \hat{d}x_i, y - \hat{d}y_i) = \begin{bmatrix} \partial/\partial x \\ \partial/\partial y \end{bmatrix} \cdot s_{k-1}(x - \hat{d}x_i, y - \hat{d}y_i) \quad (3.11)$$

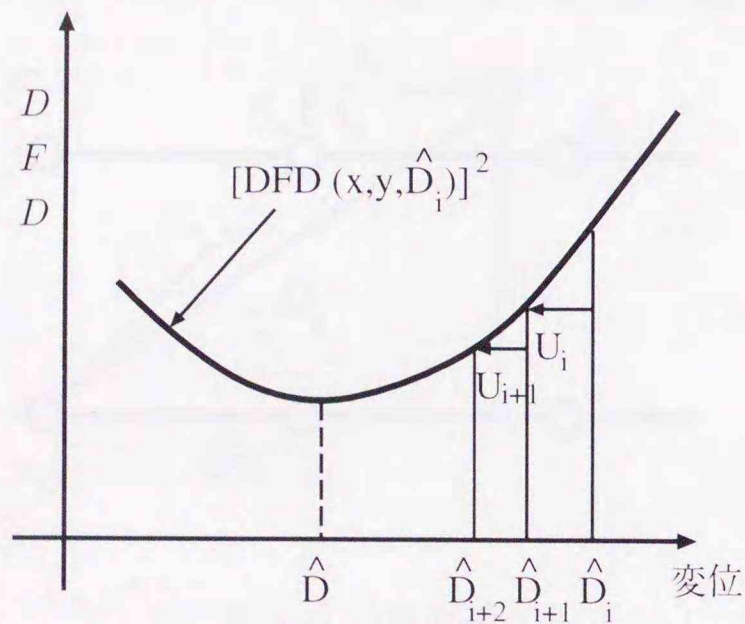


図 3.3: 収束の様子

アルゴリズム (3.8) はいくつかの画素, 例えば領域  $M$  から更新項を計算するために拡張することができる。それは量子化雑音をならすため、更新項は次のように与えられる。

$$U_i = -\frac{1}{2}\varepsilon \nabla_{\hat{D}_i} \sum_{j \in M} W_j [DFD(x, y, \hat{D}_i)]^2 \quad (3.12)$$

ここで  $W_j \geq 0$  であり、

$$\sum_{j \in M} W_j = 1$$

式 (3.10) において  $DFD$  と  $\nabla s_{k-1}$  の評価は整数でない変位  $\hat{d}x_i, \hat{d}y_i$  に対して輝度  $s_{k-1}(x - \hat{d}x_i, y - \hat{d}y_i)$  の内挿を必要とする. 4つの近傍画素  $S_A, S_B, S_C, S_D$  は図 3.4 で説明されているように, この内挿のために考慮される. 最初に,  $\hat{d}x_i$  と  $\hat{d}y_i$  の整数部分が  $S_A$  の位置を決定する.  $\hat{d}x_i, \hat{d}y_i$  の小数部分  $\delta_x, \delta_y$  は以下の式に従う線形2次元内挿のために用いられる.

$$s_{k-1}(x - \hat{d}x_i, y - \hat{d}y_i) = (1 - \delta_y)[(1 - \delta_x)S_A + \delta_x S_B] + \delta_y[(1 - \delta_x)S_C + \delta_x S_D] \quad (3.13)$$

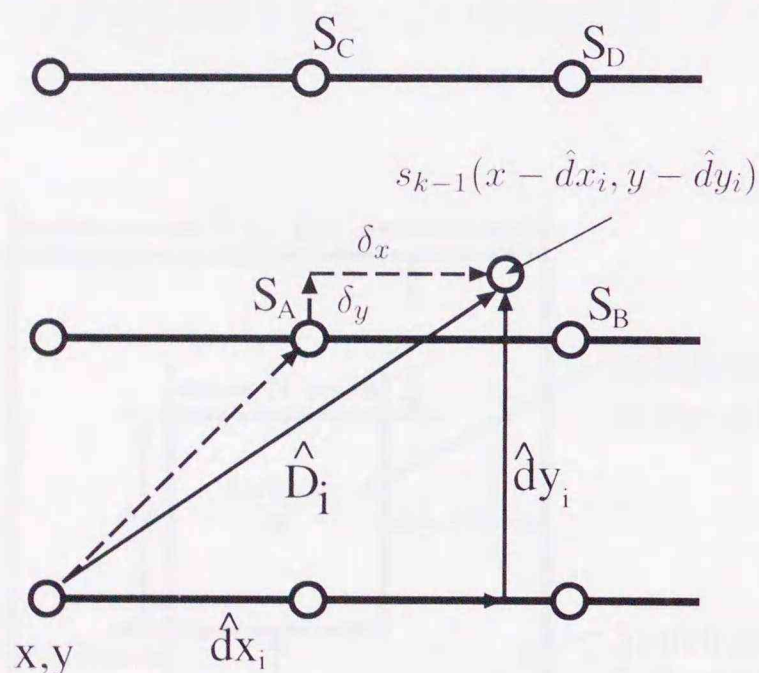


図 3.4: 内挿方法

実現の複雑さを減らすために Netravali と Robbins は内挿式 (3.13) と変位の検出アルゴリズムに対して簡単にしたものを [31] の中で提案した.

### 3.4 ブロック・マッチング法

再帰的に変位を検出する代わりに, 規格化された2次元の相互相関関数  $NCCF$

$$NCCF(D) = \frac{R_{s_k s_{k-1}}(D)}{\sqrt{R_{s_k s_k}(0) \cdot R_{s_{k-1} s_{k-1}}(0)}} \quad (3.14)$$

を測定することで、変位  $D$  の検出が相関のピークの位置から得られる。点  $(x, y)$  に対する変位を見つけるために、点  $(x, y)$  を中心とする  $M \times N$  画素のブロックがフレーム  $k$  から取られ、最も良い整合を見つけるためにフレーム  $k-1$  の探索領域  $SR$  における画素と相関を取る。  $dm$  を画素の最大の水平あるいは垂直の変位と仮定すると、探索領域  $SR$  は図 3.5 で示されるように次の式によって与えられる。

$$SR = (M + 2dm) \times (N + 2dm) \quad (3.15)$$

ブロックサイズが  $7 \times 7$  で最大の変位が  $dm = 10$  である場合、探索領域は  $SR = 27 \times 27$  である。

計算の煩雑さを減らす簡単な方法は固定された数の方形ブロックに画像を分割し、1つのブロックの全ての画素が同じ変位であることを仮定する方法である。このように1つのみの動きベクトルがブロック毎に計算される。この技術はブロック・マッチング法と呼ばれている [35],[34]。

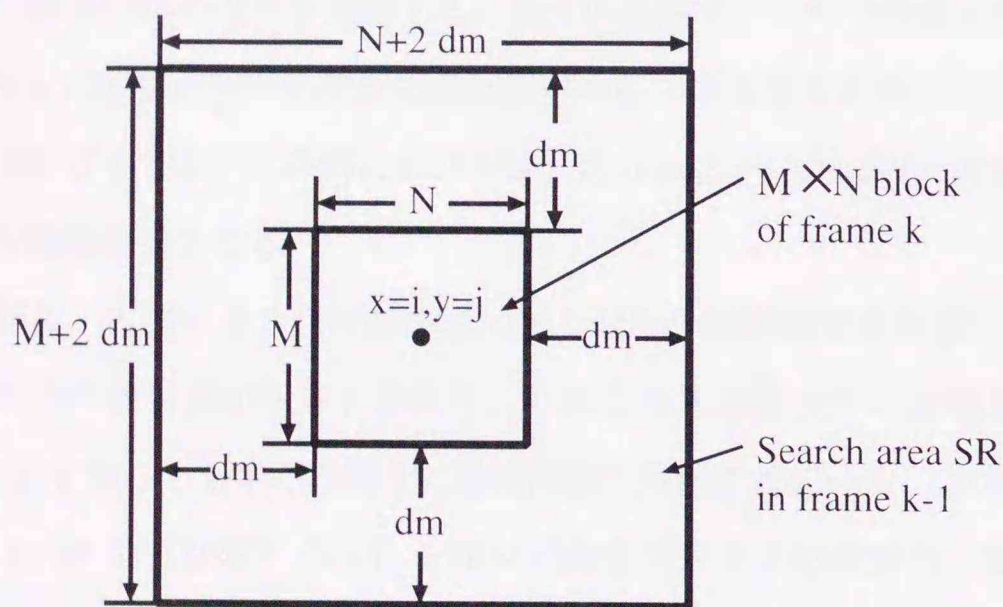


図 3.5: 探索領域

計算の煩雑さを減らすもう1つの方法はマッチングの規準の  $D(i, j)$  を簡単にするものである。以下の式に従ったマッチングの基準である規格化相互相関関数  $NCCF$

$$NCCF(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N s_k(m, n) \cdot s_{k-1}(m+i, n+j)}{\left[ \sum_{m=1}^M \sum_{n=1}^N s_k^2(m, n) \right]^{1/2} \left[ \sum_{m=1}^M \sum_{n=1}^N s_{k-1}^2(m+i, n+j) \right]^{1/2}} \quad (3.16)$$

$-dm \leq i, j \leq dm$  に関し, J.R.Jain と A.K.Jain [32] は平均自乗誤差  $MSE$  を適用し,

$$MSE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N [s_k(m, n) - s_{k-1}(m+i, n+j)]^2 \quad (3.17)$$

古閑ら [27] は絶対値のフレーム差分の平均  $MAD$  を使うことを提案した.

$$MAE(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |s_k(m, n) - s_{k-1}(m+i, n+j)| \quad (3.18)$$

$MAD$  の規準は乗法と除法を必要としない利点を持っている.

探索領域  $SR$  内を全探索すると計算量が多すぎるので, 探索の手順を簡単にするいくつかの方法が提案されている. これらの技術のうちの3つ, すなわち2D対数探索 [32], 3段階探索 [27], 改善型共役方向探索 [33] を以下で簡単に説明する.

2D対数探索は J.R.Jain と A.K.Jain によって発表されたが [32], マッチングの規準  $D(i, j) = MSE(i, j)$  は探索が最小の歪の方向から離れるに従って単調に増加するという仮定に基づいている. 最小の歪の方向は  $(i, j)$  によって定義され,  $D(i, j)$  は最小になる. 2D対数探索は最小の歪の方向を追跡する. 各々の段階で, 5つの探索点が図 3.6 のようにチェックされる. 探索点間の距離は探索位置が中心である場合かあるいは探索領域の境界である場合に小さくする. この例においては, 点  $(i+2, j+6)$  において動きベクトルを求めるために5段階必要となる.

ほとんど同時に, 古閑らは2D対数探索によく似た3段階探索を発表した [27]. 最初の段階では, 9つの探索点がテストされる. これらの点は図 3.7 に説明されているように  $(x, y) = (i, j)$  を中心としてその周りに比較的粗く配置されている. この例においては, 点  $(i+3, j+3)$  が  $MAD$  規準 (3.18) を用いた動きベクトルの最初の近似として求められる. 第2段階では, 最初の近似点の周りにあまり粗くなく8つの探索点が配置され, 点  $(i+3, j+5)$  が求められる. 第2段階は要求される精度が達成されるまで繰り返される.  $dm \leq 6$  の探索領域の場合には, 第3段階はこの例では点  $(i+2, j+6)$  への動きベクトルが求められる.

Srinivasan と Rao は, 共役方向探索と呼ばれる新しい探索方法を提案した [33]. この方法を簡単に図 3.8 の例を基にして説明する. このアルゴリズムは  $MAD$  規準 (3.18) によって定義される最小の歪  $D(i, j)$  の方向を探索する. 最初の探索において,  $i$  方向における最小値は  $D(i-1, j), D(i, j), D(i+1, j)$  を計算することによって決定される.  $D(i+1, j)$  が最も小さいと判明した場合,  $D(i+2, j)$  もまた計算され,  $D(i, j), D(i+1, j), D(i+2, j)$

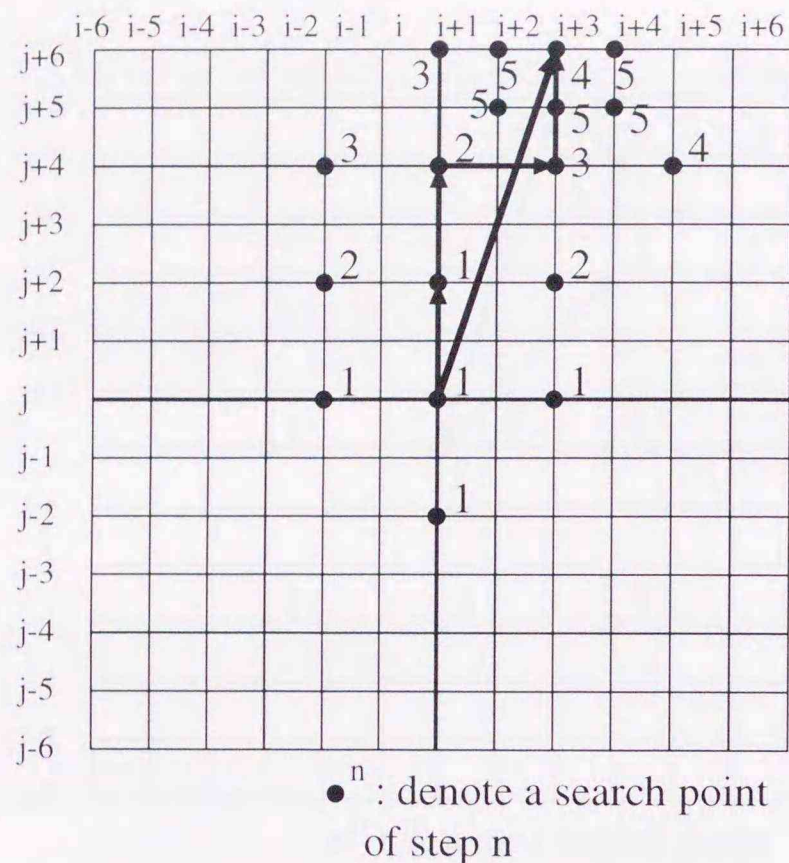


図 3.6: 2D対数探索

の内の最も小さい値が求められる。この方法を進めることで、最小値が2つの高い値の間に位置する場合に  $i$  方向における最小値が検出される。2番目の探索においては、 $j$  方向における最小値は  $i$  方向での方法と同様に決定される。図 3.8 に示される例の場合には、最初の探索の結果は点  $(i+2, j)$  で2番目の探索の結果は点  $(i+2, j+6)$  である。

シミュレーションを行なうのは

### 3.5 まとめ

本章では、動画の符号化技術の中で重要な技術である動きベクトルの検出技術を概説し、中でも有名な3段階探索法とその改良版である可変段階探索法について述べた。可変段階探索法は3段階探索法に比べ、計算量が半分程度に減少するが、その分精度は落ちる。従って、閾値  $T1, T2$  を実際の応用に適した精度が得られるように調整して決定してやれば、可変段階探索法は有効な探索法といえる。これらの技術はブロック毎に行われるため、動き補償後の画像にブロック歪が生じる。このブロック歪は予測誤差信号に高周波成分が多く発生し、全体的な符号化効率を低下させる。従って、本来生じるはずのない場所に生じるブロック歪を周りのブロックとの関係等を利用して低減する必要がある。



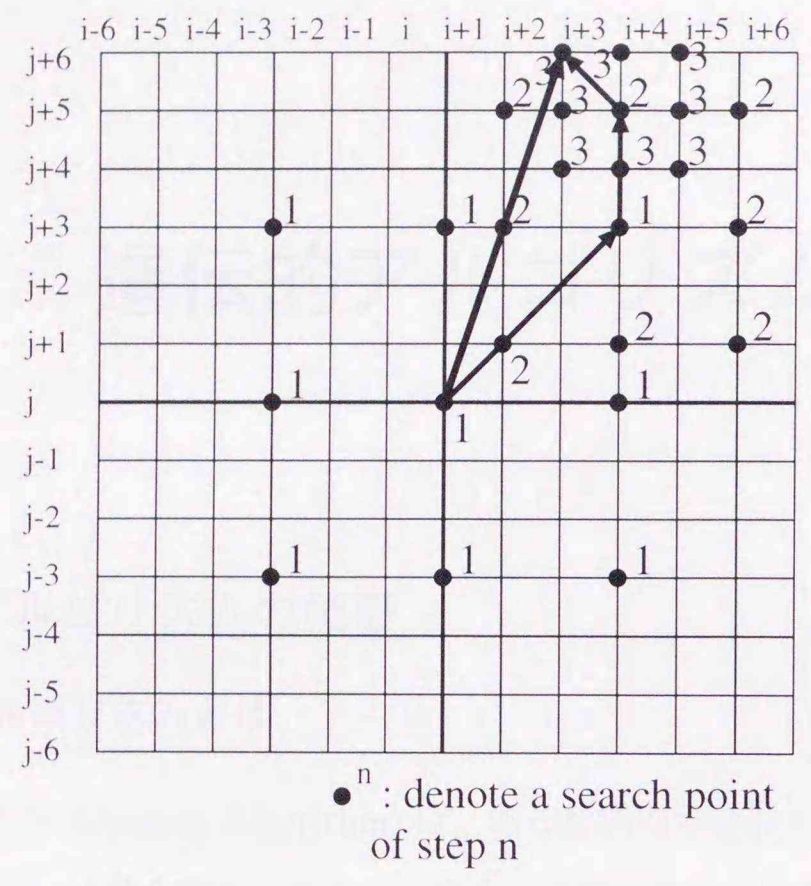


図 3.7: 3 段階探索

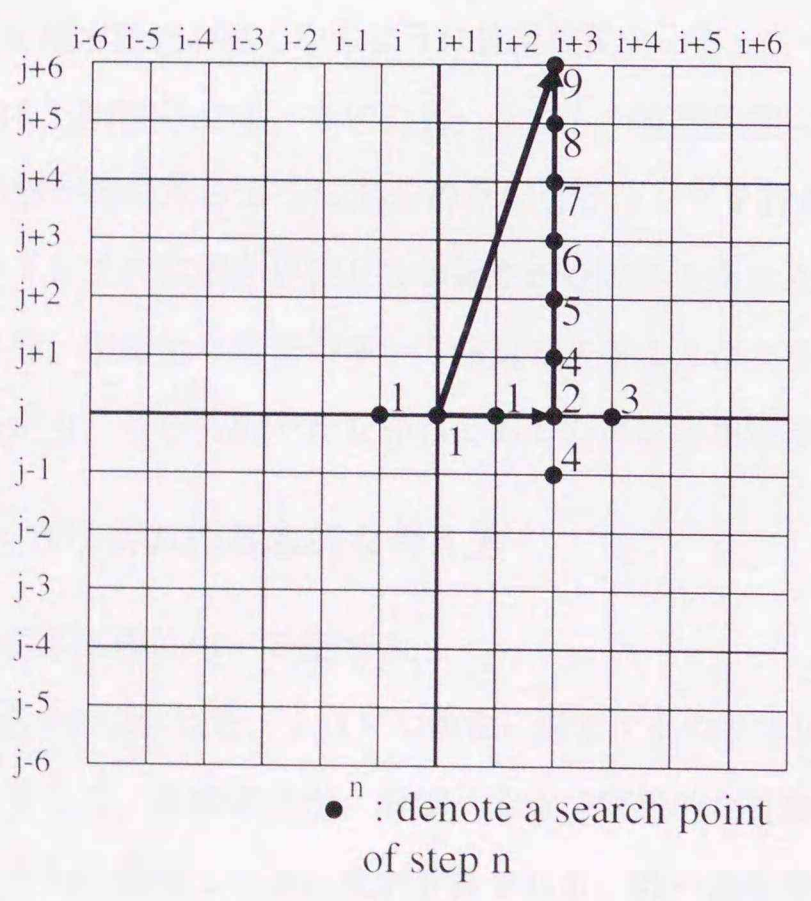


図 3.8: 共役方向探索

## 第 4 章 遺伝的アルゴリズム

### 4.1 遺伝的アルゴリズムの概要

#### 4.1.1 遺伝的アルゴリズムとは

遺伝的アルゴリズム (Genetic Algorithm) は、ある範囲内で定義されている変数  $x$  の関数  $f(x)$  の最大値あるいは最小値を与える  $x$  の値を、高速に求めるための最適化・探索アルゴリズムの一種である。GA は生物の進化の過程をヒントを得た比較的単純な基本原理を基にしており、ほとんどあらゆる最適化・探索の問題に適用可能な枠組である。GA では、遺伝子を持つ仮想的な生物集団を計算機内に設定し、あらかじめ定めた環境に適応している個体が、子孫を残す確立が高くなるように世代交代シミュレーションを実行し、遺伝子および生物集団を”進化”させる。このため、これらの仮想生物の進化によって、与えられた工学的課題の解が得られるように GA のプログラミングを行なう必要がある。GA は、実際のプログラミングの詳細を規定しない緩やかな枠組であるため、各種の規則やパラメータの設定方法等、不確定な要素が多い方法論であることが欠点として指摘されることが多い。しかしながら、むしろ緩やかな枠組であるために応用範囲が広いともいえる。

#### 4.1.2 遺伝的アルゴリズムの基本的な考え方

ここでは、GA の基本原理について述べる。

GA では、探索空間中の探索点を、1点ずつ順番に探索するのではなく、複数個の探索点を同時に用いる。そして、各探索点が、遺伝子を持つ仮想的な生物であるとみなす。各個体に対して、それぞれの環境との適応度が計算される。低い適応度を持つ個体を淘汰して消滅させ、高い適応度を持つ個体を増殖させ、親の形質を継承した遺伝子を持つ子孫の個体を生成する世代交代シミュレーションを実行する。この際、実際の生物の生殖にお

いても生ずる，遺伝子の交叉，および突然変異と呼ばれる操作を行なう．そして，最終的に非常に高い適応度の個体，言い替えれば最大値と考えられる  $f(x)$  を与える  $x$  の値を求める．

以上の操作をまとめると次のようになる．

- はじめはランダムに記号列を発生させる．
- 試した中で高い評価点を得た記号列群を参考にし，部分的なコピーや部分修正を行なって，次に試すべき記号列を作る．評価点の全体の平均が低い時は，記号列に対して大幅な修正を行なうような操作を行なう．
- 評価点の全体の平均が比較的高くなってきたら，記号列の修正量を少なくして，より詳細に記号列を決定する．

上記の操作は，非常に自然で合理的な手順のように思える．GAは，これらの操作を，個体の遺伝子に対する比較的単純な基本操作の繰り返しによって実現している．

## 4.2 単純遺伝的アルゴリズム

4.1では，GAの概要をつかむための定性的な説明を述べた．本節では，より具体的な処理アルゴリズムについて述べる．

### 4.2.1 仮想生物と環境の設定

仮想生物の進化シミュレーションを実行する前に，次に述べるいくつかの事項を設定しておく必要がある．

#### (1) 染色体と遺伝子型の決定

はじめに，個体 (individual) の染色体 (chromosome) を決定する．すなわち，世代交代の際に親の個体から子孫の個体に，どのような内容のデータをどのような形式で伝えるかを決定する．染色体は，図 4.1に示すように一般に複数の遺伝子 (gene) から構成されている．また，それぞれの遺伝子の位置は，遺伝子座 (locus) と呼ばれる一種の座標によって規定される．

個々の個体の染色体の内部表現を遺伝子型 (genotype) という，genotype は個々の遺伝子の型を表すものではないため，遺伝学分野では遺伝子型ではなく遺伝型と呼ばれる場合が多い．

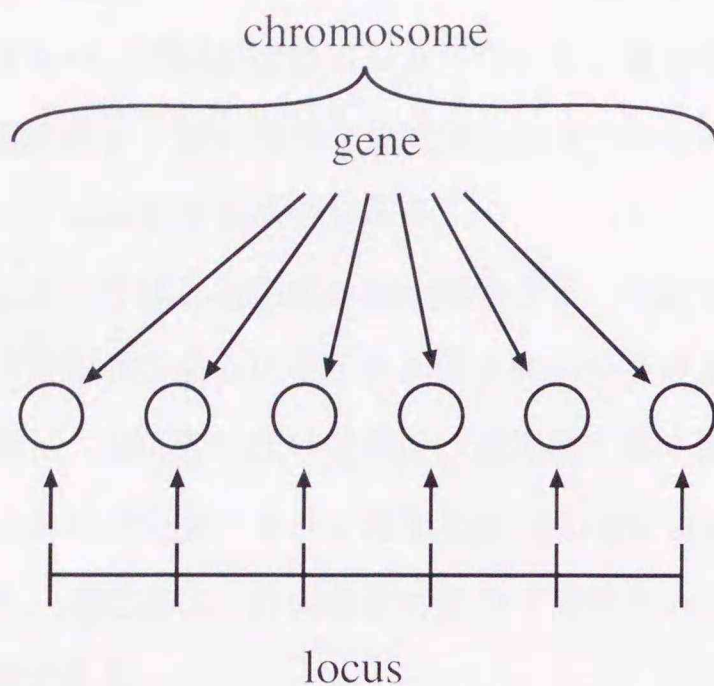


図 4.1: 染色体・遺伝子・遺伝子座

遺伝子を表す記号列としてどのようなものを用いるかは任意であるが、一般的には、0と1の並びであるビット列を用いる方法である。遺伝子型の長さについては、固定長にするのが一般的であるが、可変長にし、世代交代が進むにつれて遺伝子型を複雑化させ、長くなるように設定するGAのアルゴリズムもある。遺伝子型として何を用いるかは、後で述べる淘汰や増殖の規則とも関連しており、適用する問題に応じてプログラマが思考錯誤的に決定する。

### (2) 表現型の設定

次に、上記の遺伝子型をもつ各個体の実体に相当する表現型 (phenotype) を決定する。多くの探索問題において、表現型=遺伝子型とすれば十分であることが経験的に知られているが、このように表現型を遺伝子型と等しくした場合、世代交代につれて遺伝子型が複雑化あるいは多用化するような複雑な進化シミュレーションが困難であることが指摘されている。このため、遺伝子型に対するなんらかの変換処理によって、表現型を生成する試みのなされている。いずれにしても、表現型の決定方法には、遺伝子の場合と同様に標準的な手法が無く、問題に応じて適切に与える必要がある。

### (3) 環境と適応度の計算方法の決定

遺伝子型、表現型の決定に続き、各個体が環境にどの程度適応しているかを表す適応度 (fitness) の計算方法を決定する。適応度は、探索空間中の探索点を各個体とみなしたとき、この点における表価値に相当する。

適応度の定義方法にも一般論はないため、与えられた問題に応じて適宜設定しなければならない。一般に、与えられた問題が複雑になるにつれて、適応度の計算方法も複雑になる。また、ある個体の適応度が、他の個体と無関係に計算されるのではなく、他の個体との関係から決定されるように設定する場合もある。

シミュレーションにおいては、適応度の高い個体ほど、環境に適応している優れた個体であるとみなし、生き残る確立あるいは子孫を残す確率はその適応度の低い個体よりの高くなるようにする。逆に、適応度の低い個体は、環境にうまく適応していない個体とみなされ、消滅させる。これは進化論における自然淘汰 (natural selection) の原理を反映したものである。すなわち、適応度は、各個体が生存の可能性という面からみて、どの程度優れているかを表す尺度である。

#### 4.2.2 GA の処理手順

上で述べた仮想生物と環境の設定の後、GA では、図 4.2 に示すような手順にしたがって、生物集団 (population) を進化させる。

以下、同図中の各項目について述べる。

##### (1) 初期集団の発生

GA では、探索空間中に複数の探索点、言い替えれば複数の個体を設定してそれらの協調あるいは競合を用いる。探索開始時においては、探索空間は一般に全くのブラックボックスであり、どのような個体が望ましいかは全く不明である。このため、通常、初期の生物集団は乱数を用いてランダムに発生させる。探索空間に対してなんらかの予備知識がある場合は、評価値が高いと思われる、部分を中心にして生物集団を発生させるなどの処理を行なうこともある。

ここで、発生させる個体の総数を  $N$  とし、各個体を  $I_i (i = 1, 2, \dots, N)$  と表すことにする。また、個体  $I_i$  の遺伝子型を  $G_i$  と表記する。

##### (2) 各個体の適応度の計算

生物集団中の各個体  $I_i$  の環境と適応度  $f(I_i)$  を、あらかじめ決めた計算方法に基づいて計算する。

##### (3) 淘汰および増殖の実行

次に、生物集団の淘汰 (selection) と増殖 (multiplication) を実行する。今の世代の  $N$  個の個体  $I_1 \sim I_N$  から、重複を許して  $N$  個の個体をランダムに選択することによって、次の世代

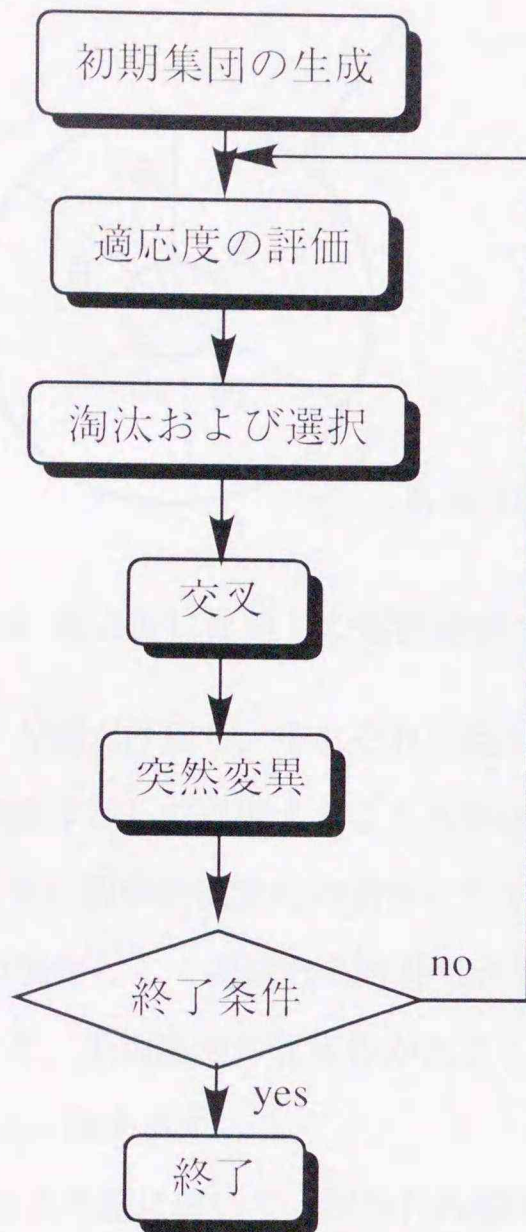


図 4.2: GA の処理手順

の  $N$  個の個体を決定する。ここで、ある個体  $I_i$  が、次の世代の個体として選択される確率  $P(I_i)$  を、次式によって決定する。

$$P(I_i) = \frac{f(I_i)}{\sum_{j=1}^N f(I_j)/N} \quad (4.1)$$

ここで右辺の分子は個体  $I_i$  の適応度、分母は現在の世代の生物集団の平均適応度である。すなわち、各世代における生存の可能性は、自分の適応度に比例する。このため適応度が高い個体ほど、次世代の個体として選ばれる確率が高い。

このようにして次世代の個体を決定する処理は、図 4.3 に示すようにルーレットにたとえることができる。

同図のルーレットにおいて、個体  $I_1$  から  $I_N$  に対応する扇型の角度  $\theta_1$  から  $\theta_N$  は、それぞれ  $f(I_1)$  から  $f(I_N)$  に比例するように作られていたとする。単純 GA における淘汰と増殖

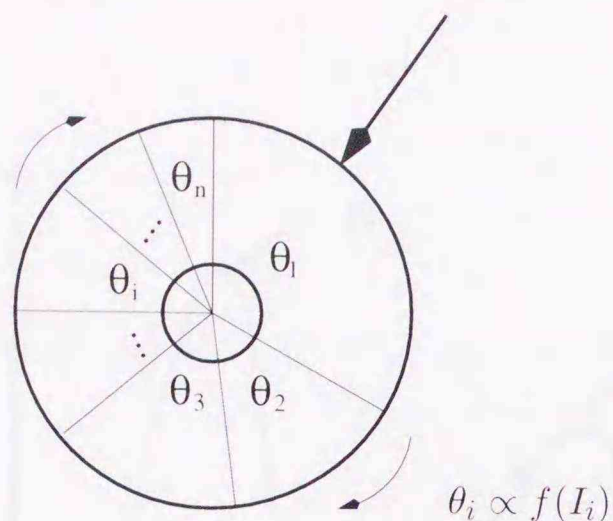


図 4.3: 適応度に比例した選択確率の設定

は、このようなルーレットを  $N$  回だけ回し、それぞれの回でルーレットの矢印にきた個体を、重複を許して次の世代の個体として選択することと等価である。

この方式では、適応度が大きい個体が次世代の個体として選ばれる可能性が高いが、適応度が低い個体でも次世代の個体として選ばれる可能性が残されることになる。これは、適応度の高い個体だけを残すと、生物集団の収束性が高まり、局所的な最適解にとらわれやすくなるためである。図 4.4 に例を示す。

図 4.4 に示す関数の最大値探索問題において、現世代の個体が  $I_1$  から  $I_N$  までであるとす。ここで、 $I_1$  から  $I_3$  のように適応度が大きい個体だけを選んで次世代の個体を決定すると、生物集団は極大値  $p_1$  の周囲に集まるため、真の最大値  $p_2$  に到達できない可能性が高い。これに対して、現時点では適応度が低い  $I_5$  の周囲にも、子孫が生成される可能性を残しておくことによって、 $p_2$  に到達できる可能性が高くなる。

以上に述べたように次世代の  $N$  個の個体を選択する処理を行なうだけでは、現時点の最大の適応度をもつ個体が、生物集団中に占める割合が高くなるだけで、新しい探索点が生じないことになる。このため、単純 GA では、次に示す遺伝子型の交叉と突然変異と呼ばれる操作を行なう。

#### (4) 遺伝子型の交叉の実行

生成された次世代の  $N$  個の個体の中から、二つの個体のペアを  $M$  個組だけランダムに選択し、それぞれに対して交叉 (crossover) と呼ぶ操作を実行する。ここで、交叉が生じる確率を、交叉率 (crossover rate) と呼ぶ。

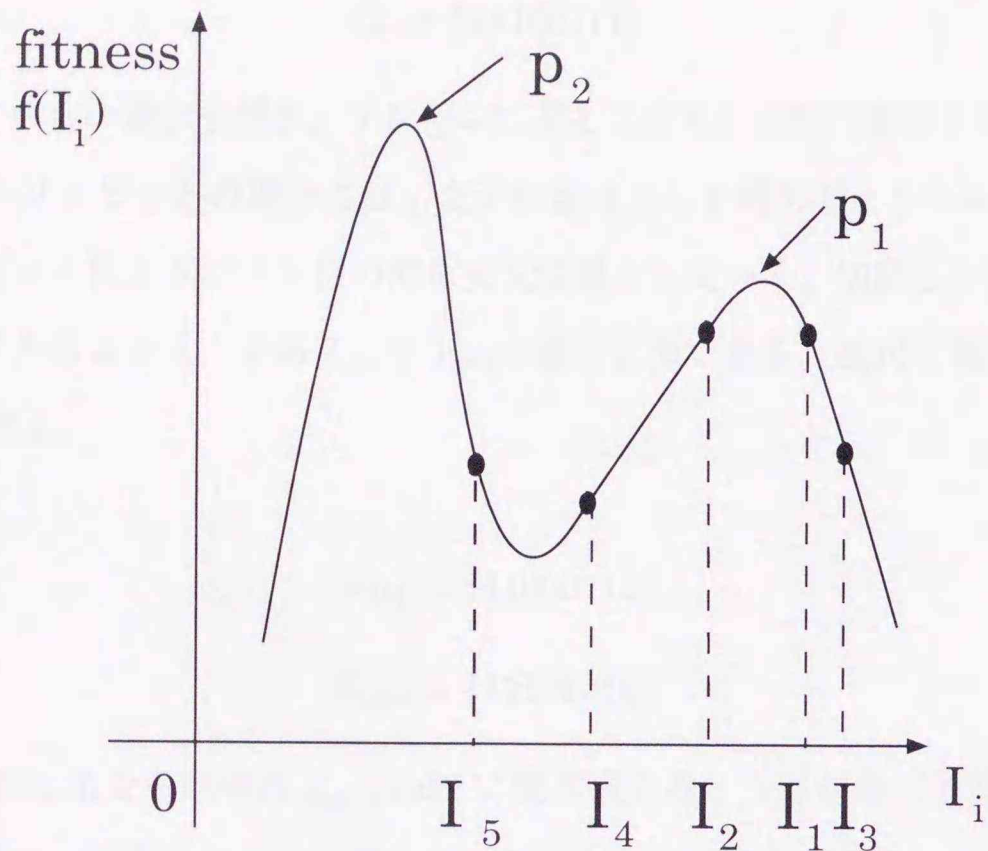


図 4.4: 局所的な最適解にとらわれる例

交叉は、二つの個体の遺伝子型をランダムな位置で部分的に入れ換える操作であり、単純GAでは、一点交叉と呼ばれるもっとも基本的な交叉を実行する。交叉には様々な手法がある。

図 4.5に、遺伝子型の一点交叉の例を示す。

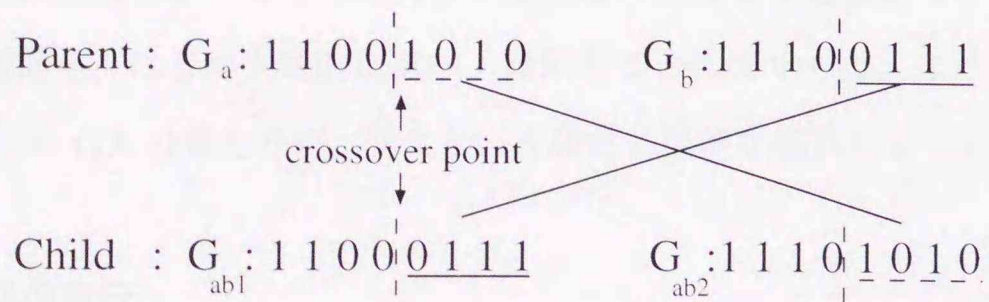


図 4.5: 遺伝子型の一点交叉の例

図 4.5に示すように、遺伝子型  $G_a, G_b$  を、ある一つのペアとして選ばれた個体  $I_a, I_b$  の遺伝子型であるとし、次式で表されるものとする。



$$G_a = \{11001010\} \quad (4.2)$$

$$G_b = \{11100111\} \quad (4.3)$$

このとき、これらの遺伝子型を、ランダムに選んだ交叉の位置で切断することを考える。遺伝子型の長さが  $n$  ビットの場合には、交叉位置は  $n - 1$  箇所考えられる。図 4.5 の例では、左から 4 ビット目と 5 ビット目の間を交叉位置としている。切断した部分な遺伝子型を入れ換えることによって、子孫  $I_{ab1}$  と  $I_{ab2}$  の遺伝子型である。次式で表される遺伝子型  $G_{ab1}$  と  $G_{ab2}$  を作る。

$$G_{ab1} = \{11000111\} \quad (4.4)$$

$$G_{ab2} = \{11101010\} \quad (4.5)$$

そして、個体  $I_a, I_b$  をそれぞれ  $I_{ab1}, I_{ab2}$  に置き換える。これが遺伝子型の交叉である。遺伝子型の交叉は、実際の生物の生殖においてもみられる減少である。

交叉によって生成された子孫の個体  $I_{ab1}, I_{ab2}$  は、親にあたる個体  $I_a$  および  $I_b$  のそれぞれから形質を継承した個体である。この交叉の処理によって、個体の遺伝子型の種類が多くなり遺伝子型の進化が生じる。これは、探索空間において、現在の探索点とは少し異なる位置に、新しい探索点を生成させることに相当している。初期の生物集団では、様々な遺伝子型をもつ個体があるため、交叉によっても同様にいろいろな個体が生じる。一方、進化が進み、遺伝子型がある傾向に定まりつつある過程では、どの個体遺伝子型にも大きな差が無いため、交叉によって作り出される個体の遺伝子型も似通ったものになる。すなわち、探索空間を、はじめ大局的に調べ、傾向をつかんでからは、より詳細に調べるという 4.1.2 で述べた GA の考え方が、交叉という非常に単純な操作によって実現されていることがわかる。

#### (5) 突然変異の実行

遺伝子型の交叉に続いて、突然変異 (mutation) と呼ばれる操作を実行する。この突然変異の生起確率を、突然変異確率 (mutation rate) とよぶ。突然変異にもいろいろな方式があるが、単純 GA における突然変異は、各個体の遺伝子に相当する各ビットを、突然変異確率で、0 を 1、あるいは 1 を 0 に変更する操作である。図 4.6 に突然変異の例を示す。

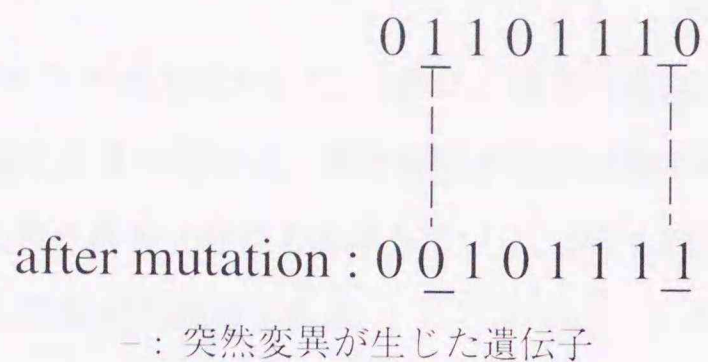


図 4.6: 突然変異の例

この図では、左から2ビット目と8ビット目の遺伝子に突然変異が生じ、それぞれ対立遺伝子に変更されている。遺伝子の内容が低い確率で書き換えられるという現象は、実際の生物の遺伝子においてもみられる現象である。

この突然変異の操作によって、交叉だけでは生じない遺伝子をもつ個体が生成されることになり、探索の観点からみれば、現在の探索点から大きく離れた場所の探索点を発生させることになる。これは、生物集団が、図 4.4 に示したような局所的な最適解に捕らわれそうになった時、そこから脱出する働きがある。このため、突然変異は進化シミュレーションに必要な操作ではあるが、突然変異確率を大きくしすぎると、先に述べた遺伝子型の交叉による親の形質の遺伝の特徴が失われ、探索空間中をランダムに探索することと同様になるので注意が必要である。このため、突然変異確率は、通常 0.1~5% 程度の値にすることが良いとされている。

以上の処理によって、次世代の  $N$  個の個体が決定された。

#### (6) 生物集団の評価

次に、生成された次世代の生物集団が、進化シミュレーションを終了するための評価基準を満たしているかどうかを調べる。評価基準は、単純 GA を適用する問題に依存するが、代表的な評価基準として次のようなものがある。

- (1) 生物集団中の最大の適応度が、ある閾値より大きくなった。
- (2) 生物集団全体の平均適応度が、ある閾値より大きくなった。
- (3) 世代交代回数に対する生物集団の適応度の増加率が、ある閾値以下の世代が一定期間以上続いた。

(4) 世代交代の回数が、あらかじめ定めた回数に達した。

(1) および (2) が標準的な評価基準である。(3) は、個体の進化が停止したことを表しており、適応度が低い状態のままの場合は、生物集団が探索空間中の局所的な安定点にとらわれてしまい、探索が失敗に終わったことを示している。(4) も同様に、探索が失敗するかも知れないことを想定した場合の設定である。

終了条件が満たされた場合は、進化シミュレーションをすべて終了し、現時点の生物集団中の個体の表現型を、求める工学的な課題の解答にする。終了条件が満たされない場合は、図 4.2 に示したように、手順 2 の各個体の適応度の処理に戻って、進化シミュレーションを実行する。

以上が単純 GA の処理手順である。

### 4.2.3 単純 GA の特徴

単純 GA に代表される GA の操作およびその効果等については、まだ理論的な検討が十分ではなく、多くの実際上の問題に適応し、その効果を実験的に調べて確認している段階である。単純 GA の基本操作によって、4.1.2 で述べたような特徴を持つ探索が行なわれるであろうことを定性的に理解することはできるが、単純 GA におけるパラメータである、生物集団中の個対数、交叉率、突然変異率などを、問題に応じて最適に決定する汎用的な手法は現在のところ分からないため、さまざまなパラメータを試して、そのだいたいの設定範囲などを経験から蓄積している。今後、理論面での GA の関する考察が必要である。

単純 GA による探索には、他の GA のアルゴリズムと同様に、最急降下法等の従来の探索手法に比較して、次のような特徴がある。

1. 複数の探索点から同時に探索を行ない、互いに協調または競合することによって局所的な安定点を避ける機能がある。
2. 次の探索点を設定するために、現在の探索点における評価値だけを用いており、評価値の勾配などを用いないので、不連続な評価関数の探索にも適応できる。

反面、単純 GA、および他の GA には一般に次のような問題点がある。

1. 生物、環境および淘汰・増殖などの設定に対して、現在のところ一般的な方法がなく、個々の問題に応じてプログラマが思考錯誤的あるいは経験に基づいて設定しなければならない。

2. 一般に多数のパラメータを設定する必要があり、これらの値を思考錯誤によって決定しなければならない。

### 4.3 スキーマ定理

GAに関する理論的解析は、現時点ではまだ十分ではない、スキーマ定理は、GAを体系化した J.H.Holland によって提唱された GA の基本理論であり、GA に関する数少ない数学定理となっている。この定理は、遺伝子型の中の注目している遺伝子の並びが、進化シミュレーションにおいてどのような生存の確率を持つのかを求める指針になっている。ここでは、スキーマ定理の概略について述べる。

スキーマ (schemata) は、アルファベットに don't care 記号 \* を加えた文字列の集合である。たとえば、次に示すスキーマ  $S$  は、四つの文字列を表す。

$$S = *0011*01 = \{00011001, 10011001, 00011101, 10011101\} \quad (4.6)$$

ここで、記号 \* は 1 または 0 のどちらでもよいことを示している。

このとき、スキーマ  $S$  の次数 (order)  $O(S)$  および構成長 (defining length)  $\sigma(S)$  を次のように定義する。

$$O(S) = (\text{全長 } L) - (* \text{ の数}) \quad (4.7)$$

$$\sigma(S) = (S \text{ の最左と最右の非 } * \text{ の記号間の距離}) \quad (4.8)$$

すなわち、式 4.6 スキーマの次数は 6、構成長も 6 である。

交叉と突然変異がなく個対数  $N$  が一定のときは、スキーマ  $S$  の世代交代での平均増加率  $R$  は、 $G_i$  を個体  $I_i$  の遺伝子型として次式で表される。

$$R = \frac{f(S)}{\{\sum_{i=1}^N f(G_i)\}/N} \quad (4.9)$$

ただし、 $f(S)$  は、スキーマ  $S$  で表される全てのスキーマに対する適応度の平均であり、式 4.6 の  $S$  については次式で計算される。

$$f(S) = \{f(00011001) + f(10011001) + f(00011101) + f(10011101)\}/4 \quad (4.10)$$

また、式 eq-gasedai 右辺の分母は、個体全体の平均適応度である。  
 交叉によってスキーマが分裂される確率  $R_c$  は次式で定義される。

$$R_c = p_c \frac{\sigma(S)}{L-1} \quad (4.11)$$

ただし、 $p_c$  : 交配率 ( $0 \leq p_c \leq 1$ )

時間  $t$  においてスキーマ  $S$  に属する遺伝子型をもつ個対数の期待値を  $P(S, T)$  とする。  
 突然変異が  $p_m$  のとき、スキーマが突然変異しない確率は  $(1-p_m)O(S)$  であり、交叉と突然変異は独立なので次式が成り立つ。

$$\begin{aligned} P(S, t+1) &\geq P(S, t) \cdot \frac{f(S)}{\bar{f}} \left| 1 - p_c \frac{\sigma(S)}{L-1} \right| (1-p_m)^{O(S)} \\ &\simeq P(S, t) \cdot \frac{f(S)}{\bar{f}} \left| 1 - p_c \frac{\sigma(S)}{L-1} - O(S)p_m \right| \end{aligned} \quad (4.12)$$

ただし、 $\bar{f}$  : 平均適応度 =  $\left\{ \sum_{i=1}^N f(G_i) \right\} / N$ ,  $N$  : 個体総数

ここで、突然変異確率  $p_m$  が 1 より十分小さいとして  $(1-p_m)O(S)$  をテイラー展開して 2 次の項以下を省略した。また、不等号は、交叉および突然変異によって生じる他のスキーマからの流入分などの高次の項を省略したことによる。

式 4.12 はスキーマ定理 (Schemata Theorem) または遺伝的アルゴリズムの基本原理と呼ばれ、次に示すことを表している。

●スキーマ定理 (遺伝的アルゴリズムの基本原理) “構成長  $\sigma(S)$  が短く、次数  $O(S)$  が低く、平均適応度  $\bar{f}$  より高い適応度を示すスキーマ  $S$  が次の世代で生き残る。”

上のスキーマ定理は、適応度が高い、短いスキーマが遺伝し型中に存在する場合、そのスキーマは定理によって分断される可能性が低く、世代交代につれてその数を増してゆくことを示している。このようなスキーマは、積木 (building block) と呼ばれる。進化シミュレーションにおいて、このような積木が何種類も組み合わせられることによって、高い適応度をもつ優れた個体が生成されると考えられており、積木仮説と呼ばれている。

スキーマ定理は、あるスキーマの生存の可能性を調べる手段を与えてくれているが、進化シミュレーション中で生じた新しいスキーマの挙動に関する解析を行なうものではない。進化シミュレーションによって変化してゆく遺伝子型の数理的解析・分析方法について、今後の研究が必要である。

#### 4.4 まとめ

本章では、遺伝的アルゴリズムについて説明した。遺伝的アルゴリズムは、生物進化(選択淘汰・突然変異)の原理に着想を得ており、多点情報を利用した確率探索法の一つで良好な解を短時間で見つける事ができるという利点がある。しかしながら、各種設定、パラメータなどは経験的に定めなければいけないという短所もある。

しかし、柔軟に最適化問題に対応でき、既に多くの実用例 [13, 14] があり、実用解を求める有効な手段の一つである。

## 第5章 テーブル参照による動き補償

### 5.1 はじめに

本論文では、動画像符号化のためテーブル参照による動き補償を提案する。様々な動きを格納した‘動きテーブル’を複数フレーム毎に作成し、対象ブロック毎に動きテーブル内を探索し、ブロック内の動き補償予測誤差を最小にする動きをテーブル内から選択する。本手法のポイントは動きに関する符号量削減のために動きベクトル数を削減することにある。動きベクトル数の減少による画質の悪化を抑えるために、次の2つの動きテーブル作成法を提案する。一つ目は、動き補償を行なった際に使用される動きベクトルの出現頻度により、動きベクトル数を削減する方法。二つ目は、遺伝的アルゴリズムを用いて、動き補償予測誤差が最小になるように動きベクトル数を削減し、動きテーブルを作成する方法である。こうして作成された動きテーブルを用いることにより、動き補償予測画像の画質を劣化を低く抑えつつ符号量を削減することを示す。本論文では、動画像には動きの種類に偏りがある事を利用して、テーブルサイズを限定したテーブル参照による動き補償を用いる事によって、画質の劣化を抑えつつ従来法のブロック・マッチング法より符号量の削減を実現する。

動画像伝送の際には、高能率符号化は不可欠であり、その一つとして、フレーム間の冗長性を利用する動き補償予測がある。その基本であるブロック・マッチング法では、矩形領域(ブロック)単位の平行移動モデルによるマッチングにより動きを求め、フレーム間の予測精度の向上をはかっている。

ブロック・マッチング法では平行移動のみを考慮している。しかし、実際の動画像内の動きには、様々な動きがあるものと考えられ、回転・伸縮等を考慮した動きモデルを使用し、動き補償予測精度の向上が提案されている。例えば、アフィン変換を用いた動き補償

[40] や, Warping 予測 [41], [42], グローバル動き補償 [43] によれば, 動き補償予測性能が向上する.

文献 [46] では, これら 4 つの方法を同一条件で比較し, いずれも符号量と画質の関係で従来のブロック・マッチング法を用いた動き補償予測性能を著しく上回るものではない事を明らかにしている.

本論文ではブロック・マッチング法に基づく動き補償における, 動き情報に与える符号量の削減法を提案する.

ブロック・マッチング法における動きベクトル数を制限し, テーブルに登録しておく. このテーブルを動きテーブルと呼ぶ. 実際の動き補償を行なう際は, 動きテーブルの中から動き補償予測画像の誤差が最小になる動きを選択する. 明らかに, 本方式においては, 動きテーブルの内容が動き補償予測精度を左右するので, その作成法を詳細に検討する.

## 5.2 動きテーブル参照による動き補償

動きテーブル参照による動き補償の考え方と, 動きテーブルの作成法を述べる.

### 5.2.1 動きテーブル参照による動き補償の考え方

予備調査を行ない, 使用動きベクトルを動きテーブルに登録する. 動き補償を行なう際に, ブロック毎の動き補償予測誤差が最小になるように動きベクトルを動きテーブル内から選択する. この方式が, テーブル参照による動き補償方式である.

動きテーブル内に収められている動きのみを使用して動き補償を行なうため, 動きテーブルの作成法は, 予測画像の画質に大きく影響を及ぼす.

探索範囲内のすべての動きベクトルを格納しているテーブルを作成し, これを以下では汎用動きテーブルと呼ぶ. 汎用動きテーブルから重要な動きベクトルのみを取り出し, 動きテーブルに登録する. 汎用動きテーブルに比べ, 動きテーブルは小さくなるので, 動きに関する符号量削減が達成される. ここで, 重要な動きベクトルとして, 本論文では 2 つの方式を提案する. 一つは, 汎用動きテーブルを用いて動き補償を行ない, 各動きベクトル出現頻度を求め, その頻度により動きテーブルを作成する方法である. 他方は, 遺伝的アルゴリズム (Genetic Algorithm:GA) を用いて動き補償予測誤差が小さくなるように動きベクトルを選択する事により, 動きテーブルを作成する.

提案手法では, 連続した  $P$  フレーム (計算機実験では  $P=8$ ) をひとまとまりとして動き



テーブルを作成する。従って、動きテーブルは  $P$  フレーム毎に更新される。

### 5.2.2 動きベクトルの出現頻度を用いた動きテーブル作成法

予備的に汎用動きテーブルを用いて動き補償を行ない、使用される動きベクトルの出現頻度の高い順に優先的に動きテーブルに登録する。以下にこの作成法を詳細に説明する。

探索範囲と精度を定め、その探索範囲内の可能な動きをすべて格納した汎用動きテーブルを作成する。ここでは、探索範囲 $\pm 15.5$ 画素、 $0.5$ 画素精度とし、縦方向を $-15.5$ から、 $+15.5$ まで $0.5$ 刻みで変化させ、縦方向の走査が一回終わる毎に横方向も同じ範囲、同じ精度で変化させる。このように作成した汎用動きテーブルに $0$ から $3968$ まで識別番号を与える。汎用動きテーブルの各行には識別番号と、対応する動きベクトルが格納される。

与えられた動画像に対して汎用動きテーブルを用いて動き補償を行ない、各動きベクトルの出現頻度を調べる。

図5.1にFootball 1~8フレームで動き補償のために使用された動きベクトルの出現頻度を示す。その横軸は汎用動きテーブル中における識別番号であり、縦軸は対応する動きベクトルの出現頻度である。なお、出現頻度 $0$ の識別番号は図中では省略されている。すなわち、動きベクトルは汎用動きテーブル中に $3969$ 個格納されているが、実際には約 $1500$ 個のみが使用されている。さらに、図5.2から分かるように $1500$ 個の動きベクトル中でも使用頻度に大きな差がある。他のテスト画像に関しても、動きに関して偏りがある傾向が見られる。

汎用動きテーブルにフラグ用のコラムを設け、動きテーブルに登録する動きベクトルにはフラグ'1'をたて、残りは'0'にする。このコラムを送信することにより、受信側でも動きテーブルを再現することができる。

図5.3にフラグ用にコラムを追加した汎用動きテーブルを示す。フラグ'1'を持つ動きベクトルが動きテーブルを構成する。動きテーブルにも識別番号を与える。その番号の範囲は汎用動きテーブルの識別番号の範囲よりも狭くなり、これが、動きに関する符号量の削減につながる。

動きテーブルに登録する動きベクトルの選択法を説明する。出現頻度の高い動きベクトルを優先して登録する。登録する動きベクトル数は、符号化に有利なように、 $2$ のべき乗数 $M$ に定める。しかし、同数の出現頻度の動きベクトルが、複数存在する可能性がある。出現頻度だけでは、 $M$ 個に絞りこめない場合には、 $M$ より多い動きベクトルを、その動き

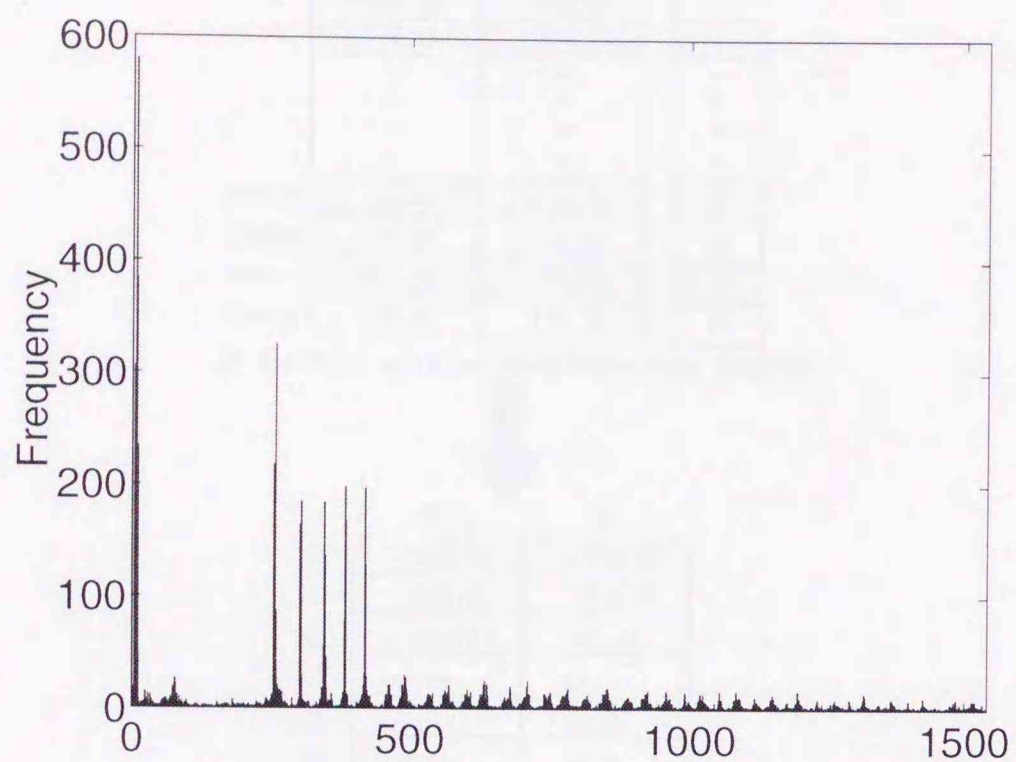


図 5.1: 動きベクトルの出現頻度 (Football)

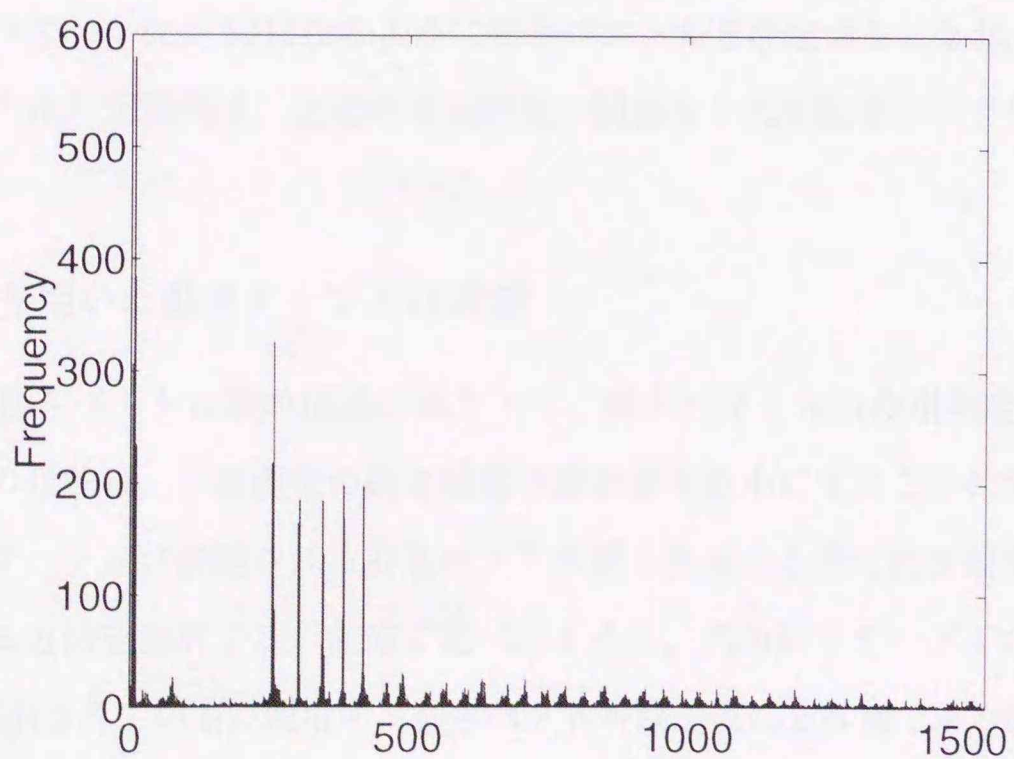


図 5.2: 動きベクトルの出現頻度拡大図 (Football)

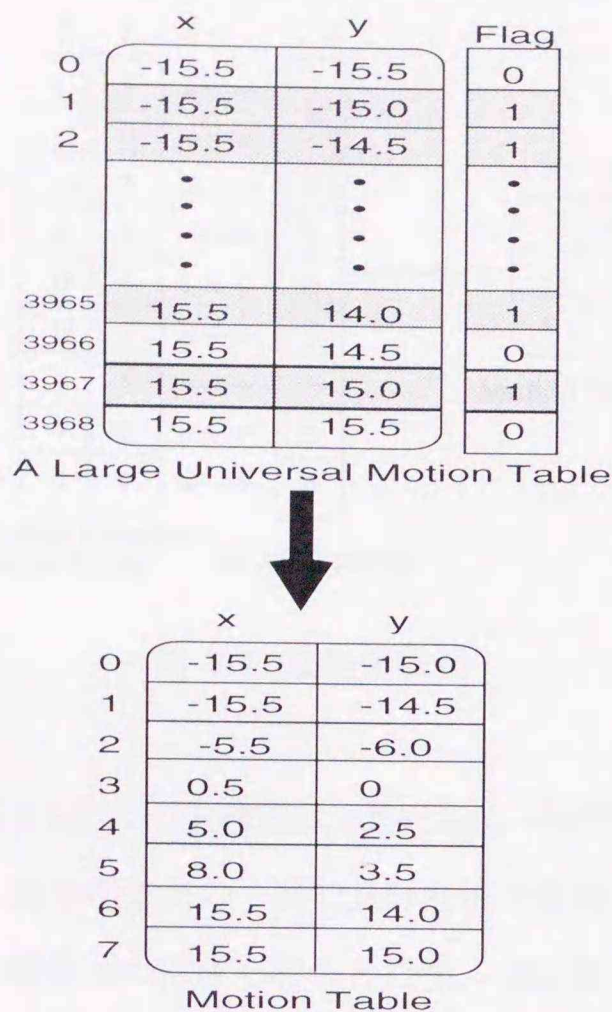


図 5.3: 動きテーブル

ベクトルの大きさが小さいベクトル，動きベクトル方向が水平方向に近いベクトルの順に優先し，動きベクトル数が  $M$  になるように動きベクトルを決定する．なお，動きテーブル内の動きベクトルの記載順は，上記の出現頻度に関係なく汎用動きテーブルの登録順そのままとする．

### 5.2.3 GA を用いた動きテーブル作成法

5.2.2では，動きベクトル数の削減にあたって，動きベクトルの使用頻度を目安とした．しかし，本来の目的は，予測画像の動き補償予測誤差を最小にすることにある．そこで本節では，動きテーブルに登録される動きベクトル数を削減する際に動き補償予測誤差を評価尺度に用いる方法を検討する．前節で述べたように，汎用動きテーブルには動きベクトルが 3969 個格納されている．使用する動きベクトル数が仮に 256 個である場合，動きテーブルは，約  $3 \times 10^{294}$  個存在する．各動きテーブル毎に一連のフレームに対して動き補償を行ない，最適な動きテーブルを見つけることは容易ではない．

本手法では，現実的に全探索困難な範囲において有効な探索を行なうため遺伝的アルゴ

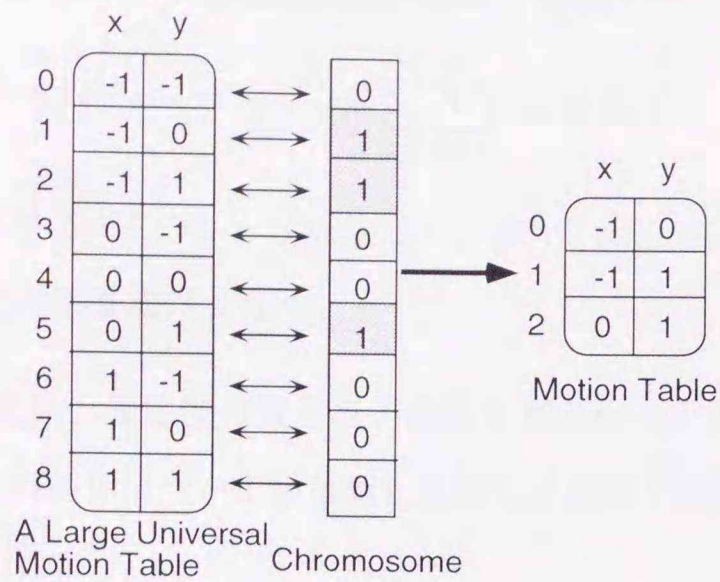


図 5.4: 染色体表現

リズム (GA) を用いる。GA[47] は、生物進化 (選択淘汰・突然変異) の原理に着想を得ており、多点情報を利用した確率探索法の一つで良好な解を短時間で見つける事ができるという利点がある。GA は初期集団の作成を行なった後、適応度の評価、選択、交叉および突然変異という一連の処理を終了条件を満たすまで繰り返す。

動きテーブル作成に GA を適応する方法を具体的に説明する。

**個体の形質表現** 本手法で用いた遺伝子型は、汎用動きテーブルから使用する動きベクトルに '1' を割り当て、使用しない動きベクトルには '0' を割り当てる。これを遺伝子型として用い、その一次元配列 (汎用動きテーブルフラグ) を染色体として扱う。図 5.4 に簡単な例を示す。この場合汎用動きテーブルは探索範囲±1 画素、9 行からなり、動きテーブルに登録する動きベクトル数は 3 個とする。汎用動きテーブル内の各動きベクトルがフラグを持つ。フラグからなるビット列 {0 1 1 0 0 1 0 0 0} が染色体を示し、動きテーブルを表している。

**初期集団の生成** 初期集団として、N 個の個体を生成する。N を個体数と呼ぶ。本手法では、個体数を 20 とした。染色体のビット列をランダムに生成するが、使用する動きベクトルを示す '1' の個数はあらかじめ与えておく。

**適応度の評価** 本手法で用いる評価尺度 (GA 処理では、適応度と呼ぶ) は、得られた動きテーブルを用いて動き補償を行ない、その予測画像の平均 2 乗誤差値 (Mean Square

Error:MSE) とする。即ち、適応度 (fitness) は、次式で与えられるとする。

$$fitness = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \hat{x}_{ij})^2 \quad (5.1)$$

ここで、フレームサイズを  $M \times N$  pixels とし、また、 $x_{ij}$ ,  $\hat{x}_{ij}$  はそれぞれ現フレーム、動き補償予測画像の画素濃度を示す。

**淘汰および増殖** 選択淘汰は、適応度の最も良い個体を無条件に次世代に残すエリート保存戦略を用い、残りの個体について評価値に比例した確率で選択淘汰を行なうルーレット戦略 [47] を用いた。

**交叉** 本手法では、一様交叉を用いているが、使用する動きベクトル数をあらかじめ与えているので、交叉の際にその数が変化してはならない。そのため、以下のような交叉を行なう。具体例により説明する。汎用動きテーブルは8行からなり、使用する '1' (使用する動きベクトルに対応) の個数は3個とする。

Parent-1	0	1	0	1	1	0	0	0
Parent-2	0	0	1	1	0	1	0	0
Mask	0	1	1	0	1	1	0	0
Child	0	1	0	1	0	1	0	0

適応度比例戦略により選択された Parent-1, Parent-2 の染色体の同一位置のビットの排他的論理和により Mask の各ビットを作成する。Mask のビットが0の場合には、Parent-1 = Parent-2 の遺伝子ビットを Child にコピーする。この場合は Child にコピーされたビットのうち1が1個含まれている。Child 中の1の数は、仮定において3個でなければならない。そこで、Child 内の残りの4ビットのうち2個をランダムに選び1を格納する。このように Child を生成する。

**突然変異** 突然変異確率は3%で、ランダムに個体を選び、その個体を初期集団と同じく生成し、入れ換える。通常の GA 操作における突然変異は、個体の遺伝子を突然変異確率により変化させる操作であるが、個体内の1の数は仮定において変化してはいけない。そのため、前述したように突然変異を行なう。

**探索終了条件** 本手法では各世代における個体の最大の適応度が30世代改善されない場合終了とする。これを越えて世代交替を行なっても結果は改善されないという予備実験結果からこのように定める。

動きテーブル内の動きベクトルの記載順は、5.2.2と同様であり、汎用動きテーブル上の登録順そのままとする。

### 5.3 実験および考察

提案方式の有効性を確認するため、実際に動画像を用いて実験を行なう。以下に示す3方式についてそれぞれ各隣接フレーム間において実験を行なった。画像を分割するブロックサイズは、 $8 \times 8$ の固定サイズで行なう。用いる画像は、Football, Flower Garden, Mobile & Calenderの3種類(画像サイズは $352 \times 240\text{pels}$ )のY信号(8bit階調)である。

各画像の1~8フレームを用い、各々のテスト画像に対して動きテーブルを5.2.2, 5.2.3に述べたように8フレーム毎に作成する。

#### 5.3.1 動き補償予測誤差による予測性能の比較

次の(i)~(iii)について実験を行なう。

- (i)BM 従来のブロック・マッチング法を用いた動き補償{探索範囲 $\pm 15.5$ 画素, 精度0.5画素}, {探索範囲 $\pm 15$ 画素, 精度1画素}の2通り
- (ii)Freq 予備動き補償における頻度の高い動きベクトルを動きテーブルに登録し, 動きテーブル参照による動き補償(5.2.2参照)
- (iii)GA 評価関数に予測画像のMSEを用いたGAにより, 動きテーブルを作成し, 動きテーブル参照による動き補償(5.2.3参照)

動きテーブル作成の際には, 0.5画素精度の汎用動きテーブル(探索範囲 $\pm 15.5$ 画素, 3969通り)を用いた。

動きに関する情報の符号化法を説明する。BMでは, 動きベクトル $(t_x, t_y)$ をそれぞれブロック間で差分を取り, VLC符号[48]により可変長符号化する。この符号量は, 動きベクトルの差分値のハフマン符号化を行なった場合の符号量と, ほぼ変わらないことを確認している。図5.5にGAの動きベクトル識別番号 $(i)$ の前ブロックとの差分値のヒストグラムを示す, また, 他のテスト画像やFreqも同様な傾向のヒストグラムが得られた。この図より, 動きベクトル識別番号 $(i)$ をブロック間で差分値を取り, 可変長符号化を用いることにより, 符号化効率が向上する。この識別番号 $(i)$ のブロック間における差分値の出現確率を式(5.2)を用いてラプラス分布に近似する。

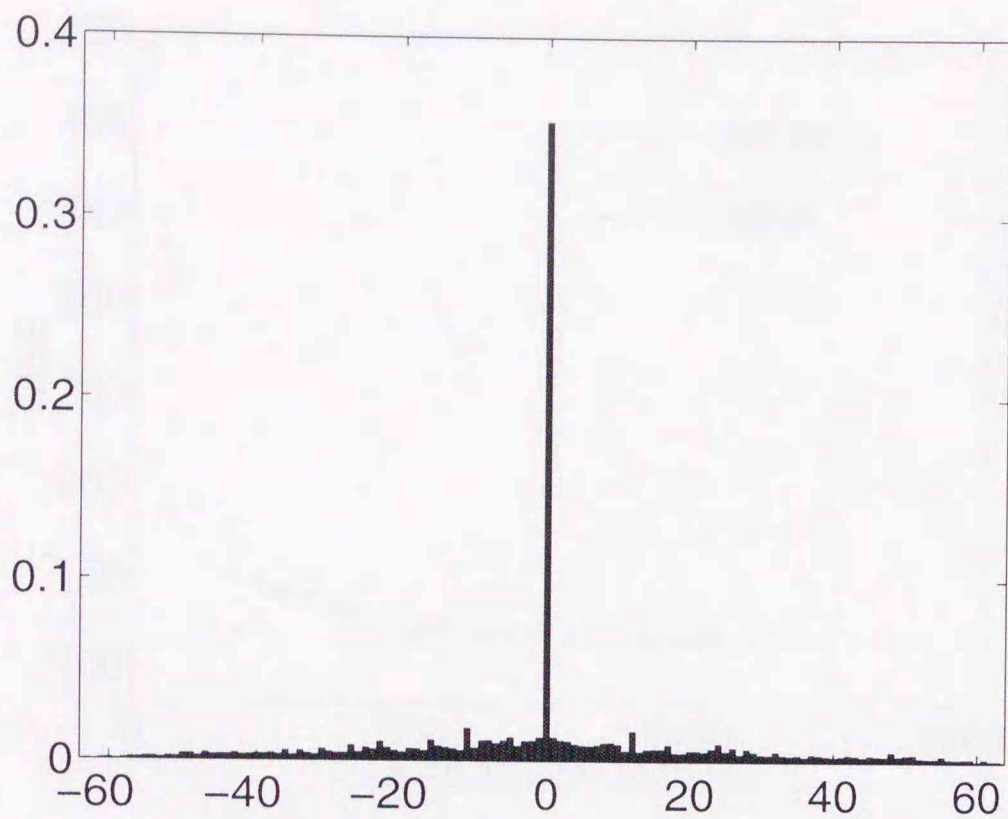


図 5.5: 動きベクトル識別番号のブロック間差分値のヒストグラム (Football)

$$y = Ae^{-\alpha|x|} \quad (5.2)$$

ラプラス分布近似した確率分布より、ハフマンテーブルを作成し、ハフマン符号化する。この確率は画像が変わっても安定していることを確認した。ここでは、Footballで得られた確率分布を用いて他のテスト画像の符号化を行なった。

汎用動きテーブルから使用する動きベクトル情報に必要な符号量は、8フレームの場合、 $\frac{3969}{352 \times 240 \times 8} \simeq 0.006[\text{bit/pel}]$ である。

Freq と GA の動きベクトル数と、その動きベクトルを使用した動き補償結果を図 5.6に示す。共に使用動きベクトル数を増加するにつれて予測画像の MSE は減少する。使用する動きベクトル数が少ない場合、GA の MSE が小さいことがわかる。

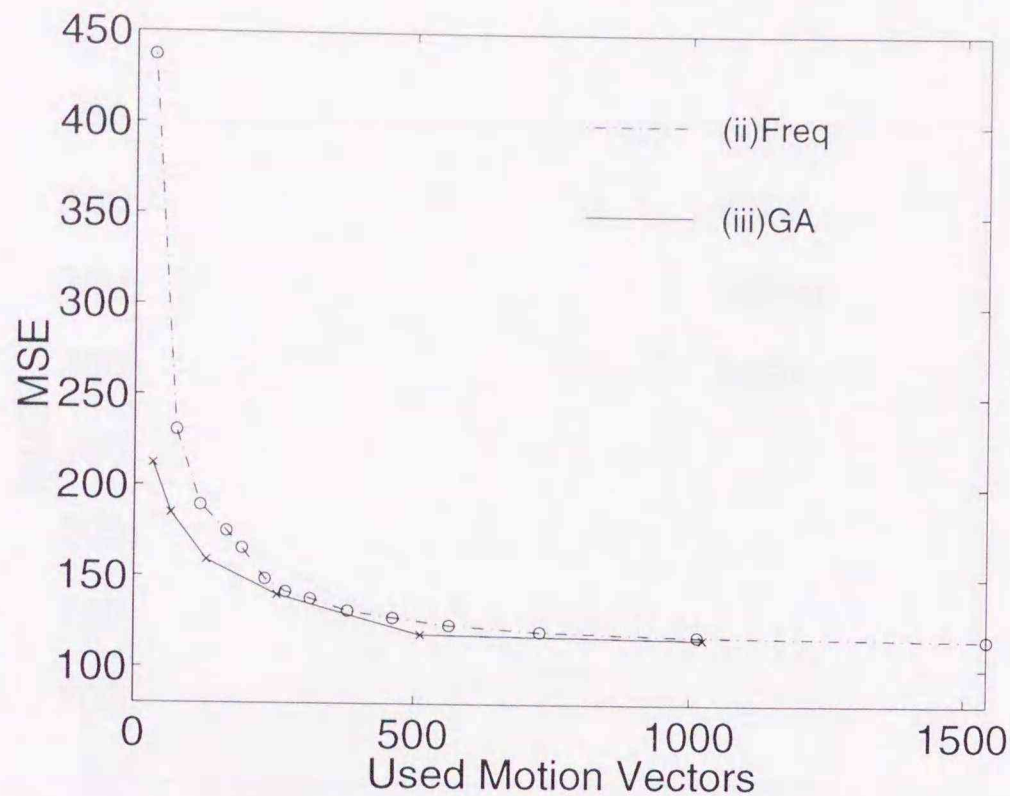


図 5.6: 使用動きベクトル数と MSE(Football)

BM, Freq, GA の符号量と予測画像の MSE との関係を図 5.7, 5.9, 5.11 に示す。

BM と比較して動きテーブル参照による動き補償 Freq, GA の方が全テスト画像に対し同じ符号量で MSE が小さい。図中の '○' は、ブロック・マッチング法による、探索範囲  $\pm 15.5$  画素、精度 0.5 画素の結果、'×' は探索範囲  $\pm 15$  画素、精度 1 画素の結果である。

また、BM では、符号量を小さくするためには、探索範囲を狭くするか精度を悪くしなければならない。しかし、本手法では無効な動きベクトルを間引くことにより、動き補償予測精度の低下を低く抑えつつ、動きに関する符号量を削減することが可能である。

### 5.3.2 Freq と GA の比較

図 5.7, 5.9, 5.11 より、全テスト画像において GA は Freq より同符号量時に MSE が減少している。Freq の動きテーブルは、使用される動きベクトルの出現頻度によって作成された。従って、動き補償予測誤差がある程度小さくなることは期待されるが、必ずしも保証されない。

一方 GA では、評価関数に予測画像の MSE を用いて動きベクトルを決定し、動きテーブルを作成している。使用される動きベクトルの頻度が多い動きベクトルが、予測画像全体の予測誤差を小さくする動きベクトルであるとは限らない。したがって、出現頻度に基づいて得られた動きテーブルを用いて動き補償を行なうより、GA を用いて作成した動き



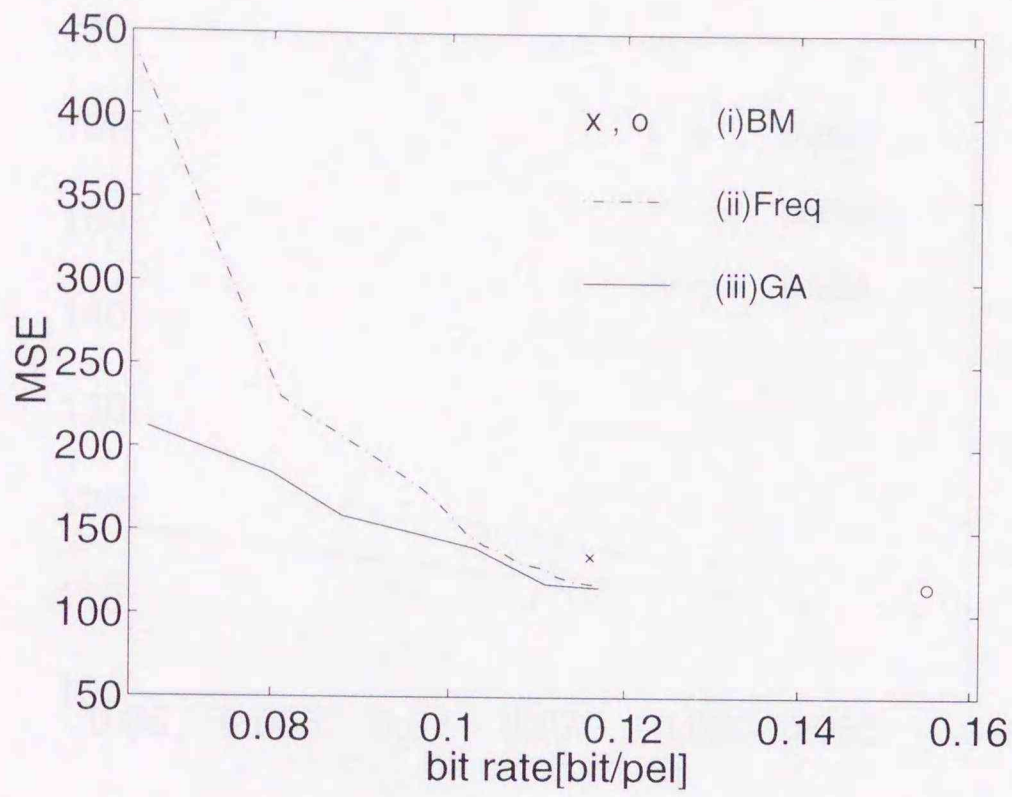


図 5.7: 符号量と MSE(8 フレーム平均)Football

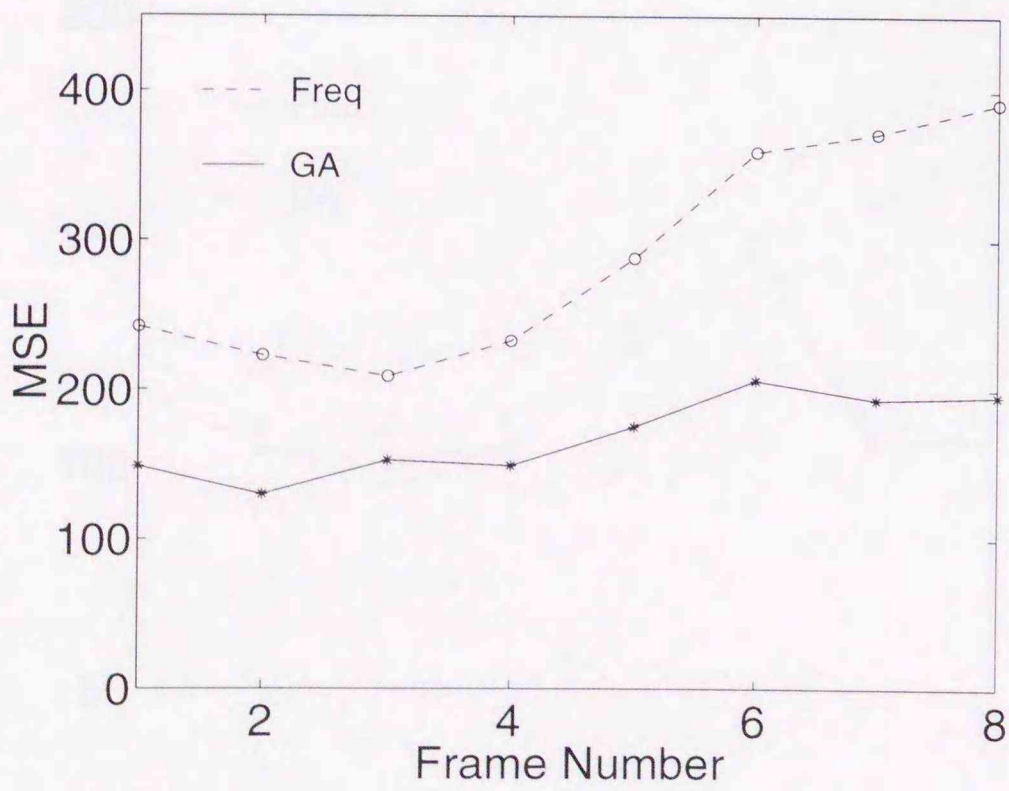


図 5.8: 各フレームにおける MSE. Freq=0.07bpp, GA=0.07bpp, Football

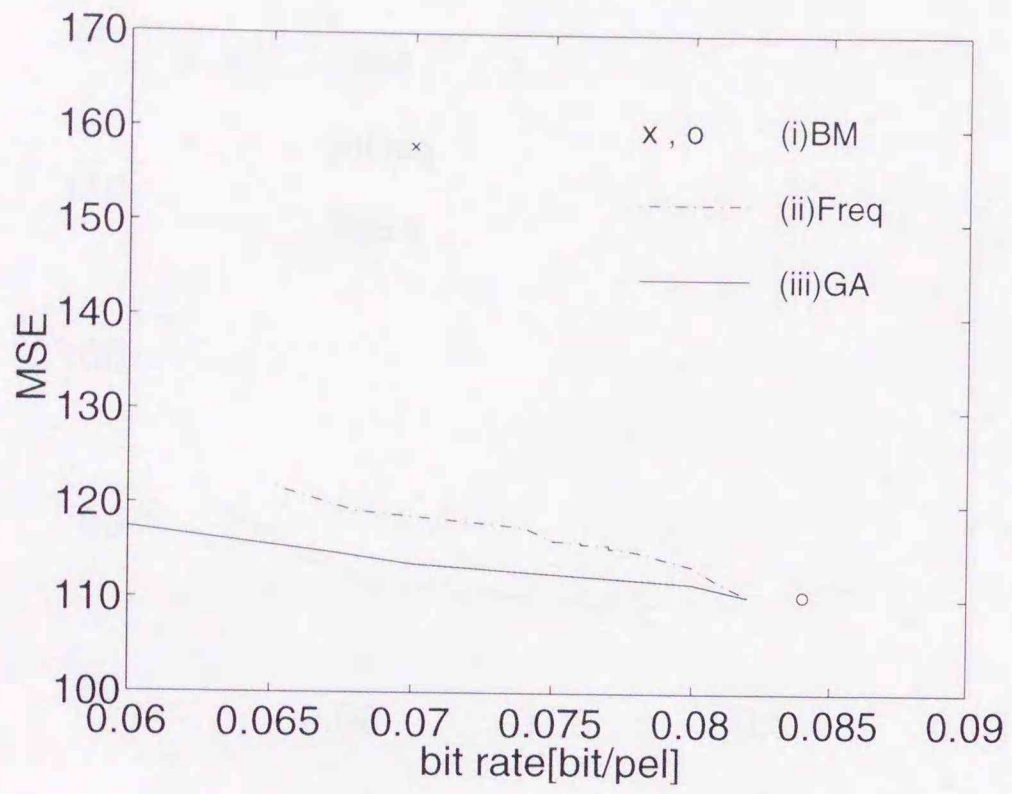


図 5.9: 符号量と MSE(8 フレーム平均) Flower Garden

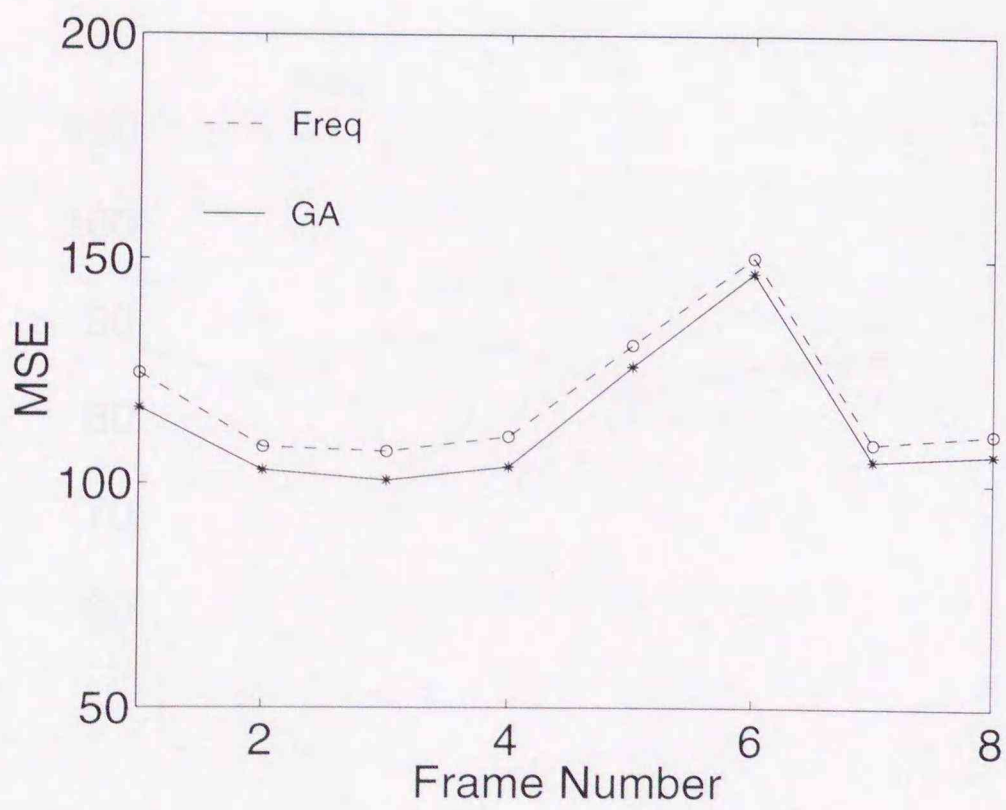


図 5.10: 各フレームにおける MSE, Freq=0.068bpp, GA=0.067bpp. Flower Garden

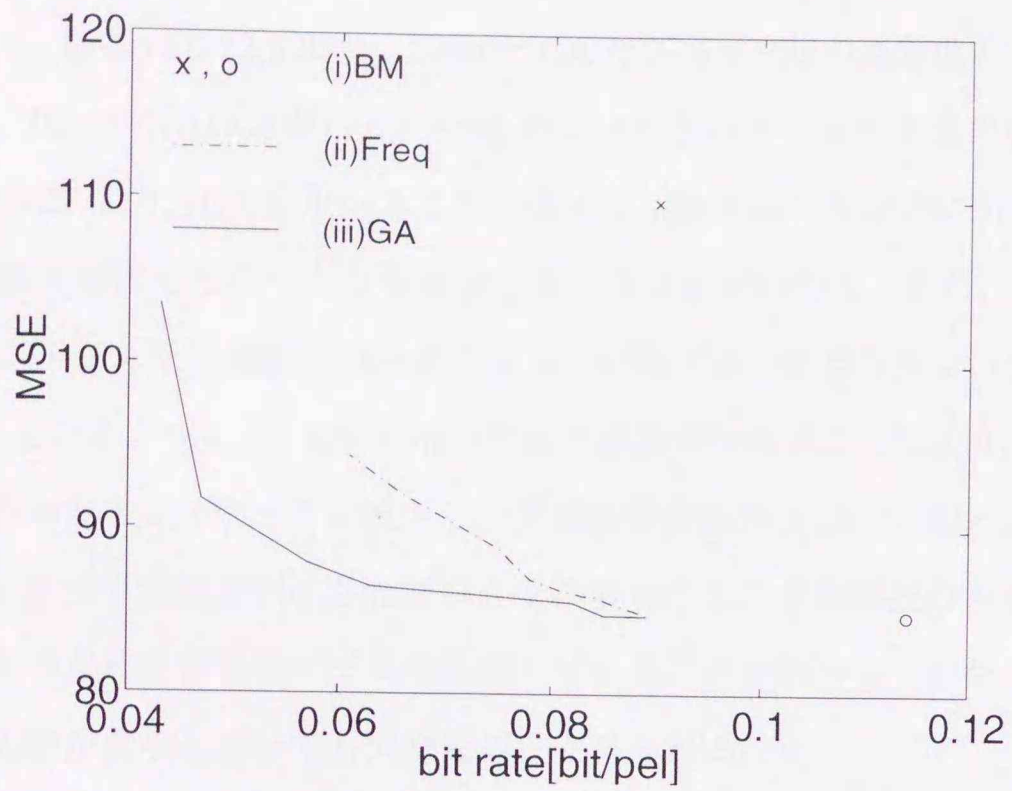


図 5.11: 符号量と MSE(8 フレーム平均) Mobile & Calendar

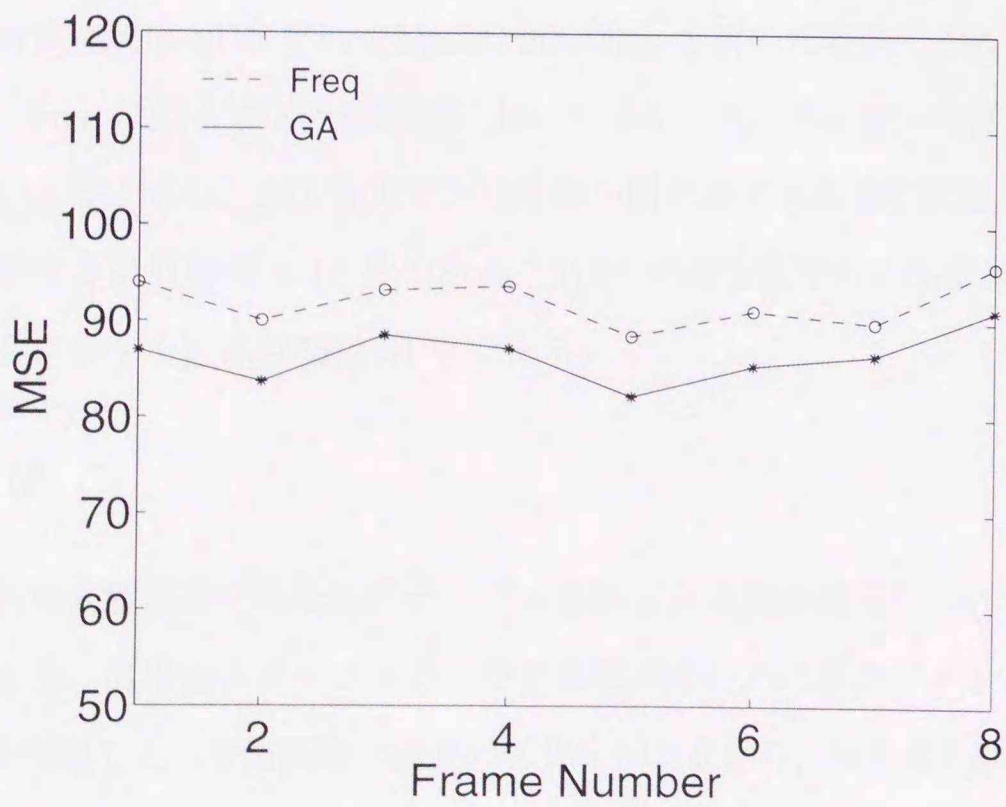


図 5.12: 各フレームにおける MSE, Freq=0.073bpp, GA=0.063, Mobile & Calendar

テーブル用いることにより符号量の削減が可能である。

図 5.14, 5.15, 5.19, 5.20, 5.24, 5.25 に動き補償予測画像を示す。別の観点から、Freq と GA の比較を行なう。図 5.17, 5.22, 5.27 は、Freq と GA でブロック毎の原画像とのそれぞれの予測誤差を求め、GA が小さい方が白く、Freq の方が小さいブロックを黒で示してある。この図 5.17, 5.22, 5.27 より、GA を用いることによって、動きの大きなブロックの予測誤差が Freq を用いて動き補償を行なうよりも減少していることがわかる。また、GA を用いた動き補償の方が、Freq より予測誤差の小さくなっているブロック数は少ないが、予測画像の MSE は減少している。つまり、GA を用いた動き補償を行なうことにより、予測性能が向上し、予測誤差の大きなブロックが減少し、予測画像全体の MSE も減少している。また、GA は、Freq に比べ予測誤差が小さなブロックの代わりに、予測誤差の大きなブロックの予測性能の向上のために動きテーブルを作成していることがわかる。また、他のテスト画像、動きの代表数が異なる場合でも同様な傾向が見られる。

図 5.14, 5.15, 5.19, 5.20, 5.24, 5.25 の動き補償予測画像に使用した動きテーブル内の動きベクトルを、表示した結果を図 5.28, 5.29, 5.30, 5.31, 5.32, 5.33 に示す。この例は、極端に異なるものを選んだ。動きベクトル数はテスト画像 football が 64 で、テスト画像 flower garden, Moblie & Calendar は 128 である。

なお、Freq, GA の動きベクトル数が 64 個のテスト画像 Football の場合の動きテーブル作成に要する計算時間は SUN UltraSPARC 200MHz を用いた場合、それぞれ 50 分、600 分程度である。Freq は動き補償予測精度において GA に比べ劣るが、計算量において優れている方法といえる。また、BM 方式で、符号量が同程度である探索範囲 $\pm 4$ 画素、精度 1 画素の場合に要する計算時間は 12 秒である。MSE が同程度である探索範囲 $\pm 8$ 画素、精度 1 画素の場合に要する計算時間は 31 秒である。

## 5.4 むすび

本論文では、動画像符号化のためのテーブル参照による動き補償における動きテーブル作成法を提案した。汎用動きテーブルから符号量削減のために動きベクトル数を減少させ動きテーブルを作成した。予測画像の画質の劣化を抑えながら、符号量の削減を計った。使用する出現頻度の高い動きベクトルを優先して動きテーブルに登録する事により、動きベクトル数を削減し、符号量の減少を可能にした。

動きベクトルの出現頻度により動きテーブルを作成したが、動き補償予測誤差を最小に

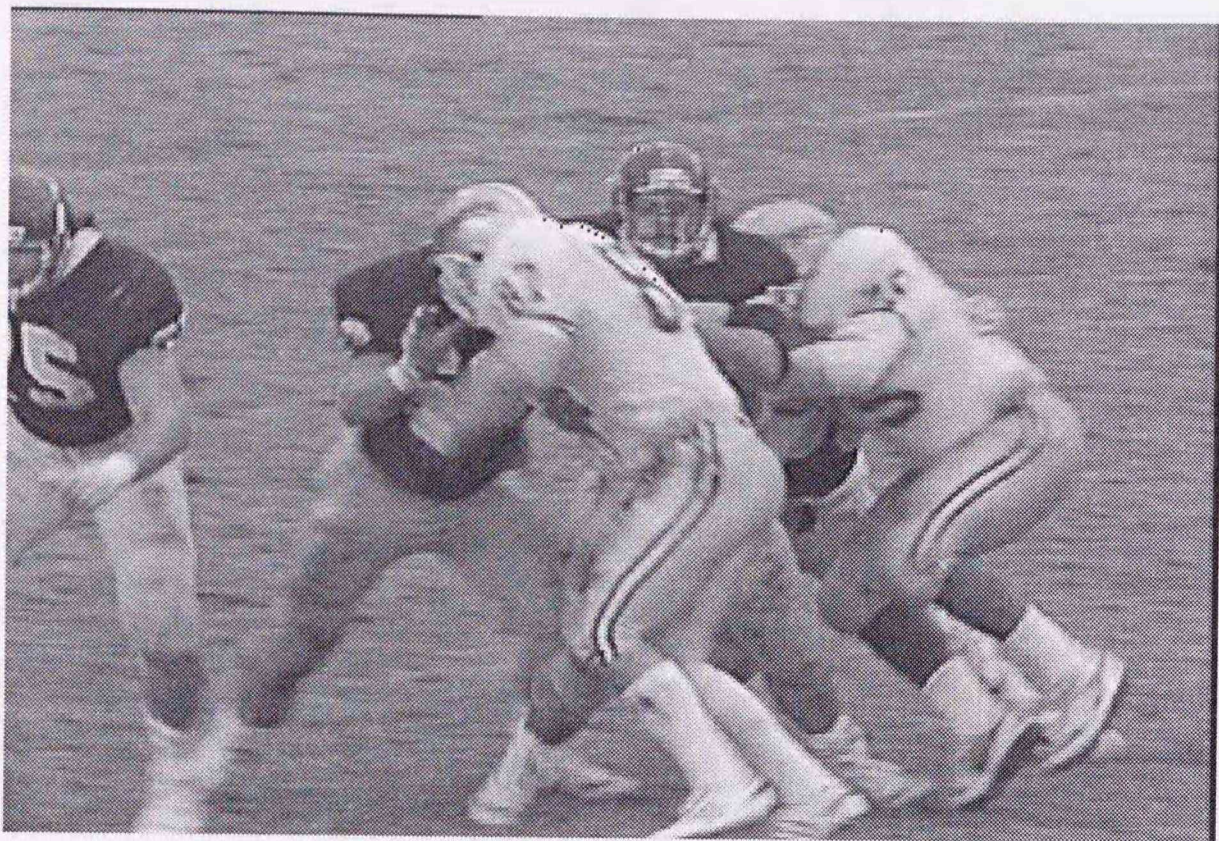


図 5.13: 原画像 (Football)

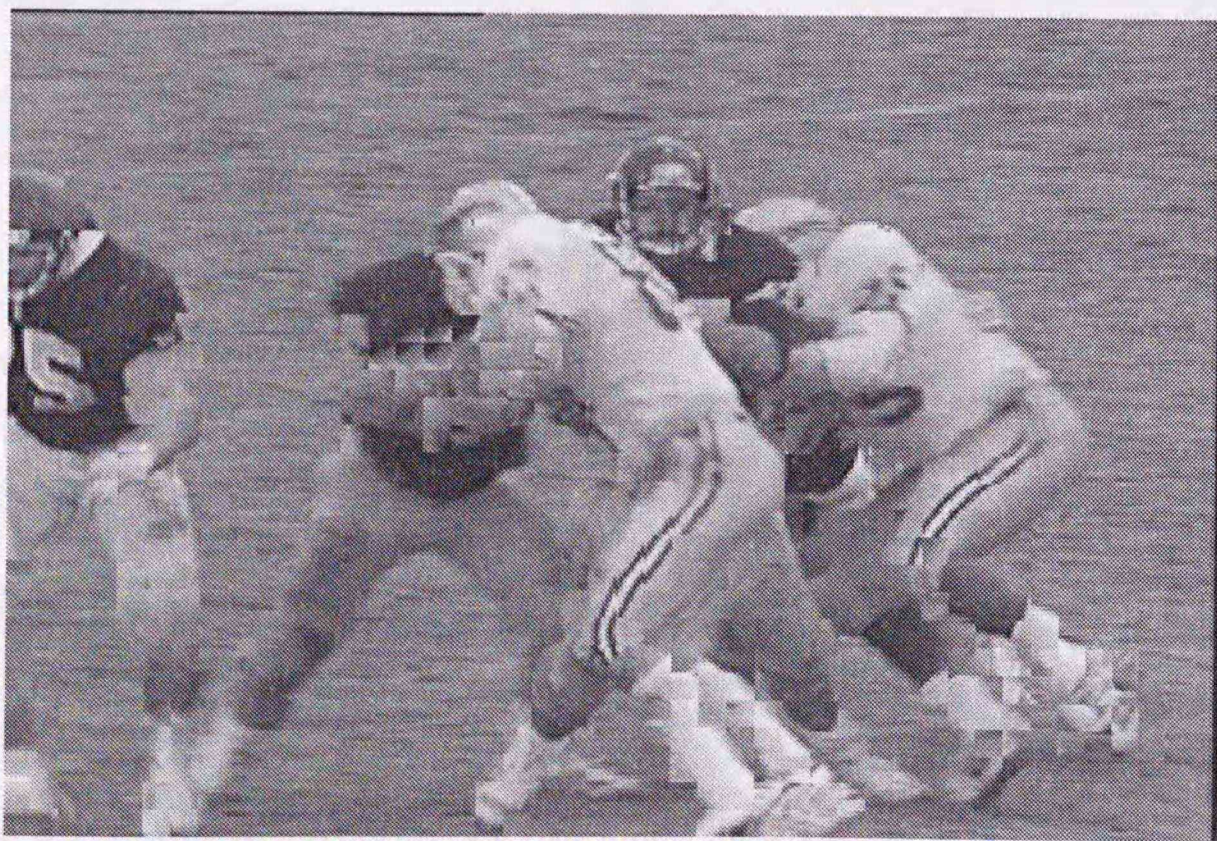


図 5.14: (b):**Freq**(bitrate=0.081[bit/pel],MSE=201.6)

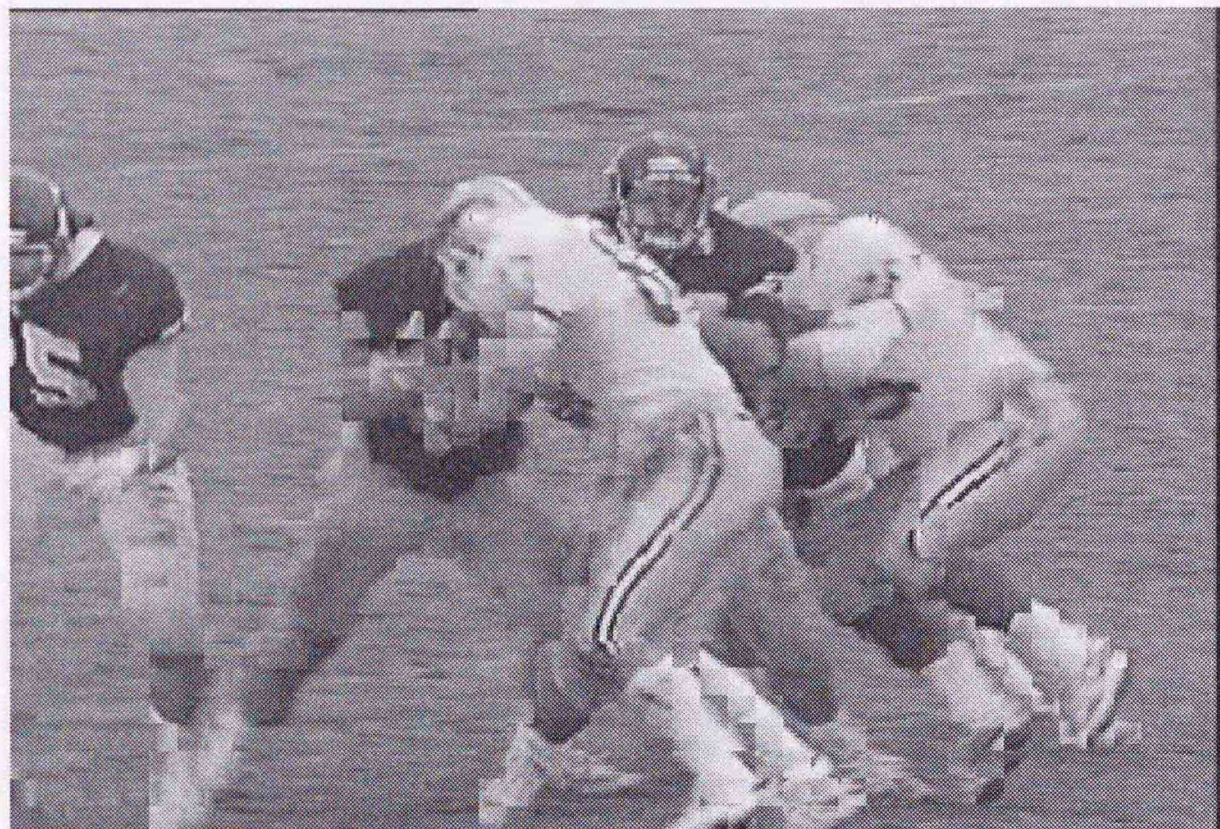


図 5.15: (c):GA (bitrate=0.080[bit/pel],MSE=170.5)

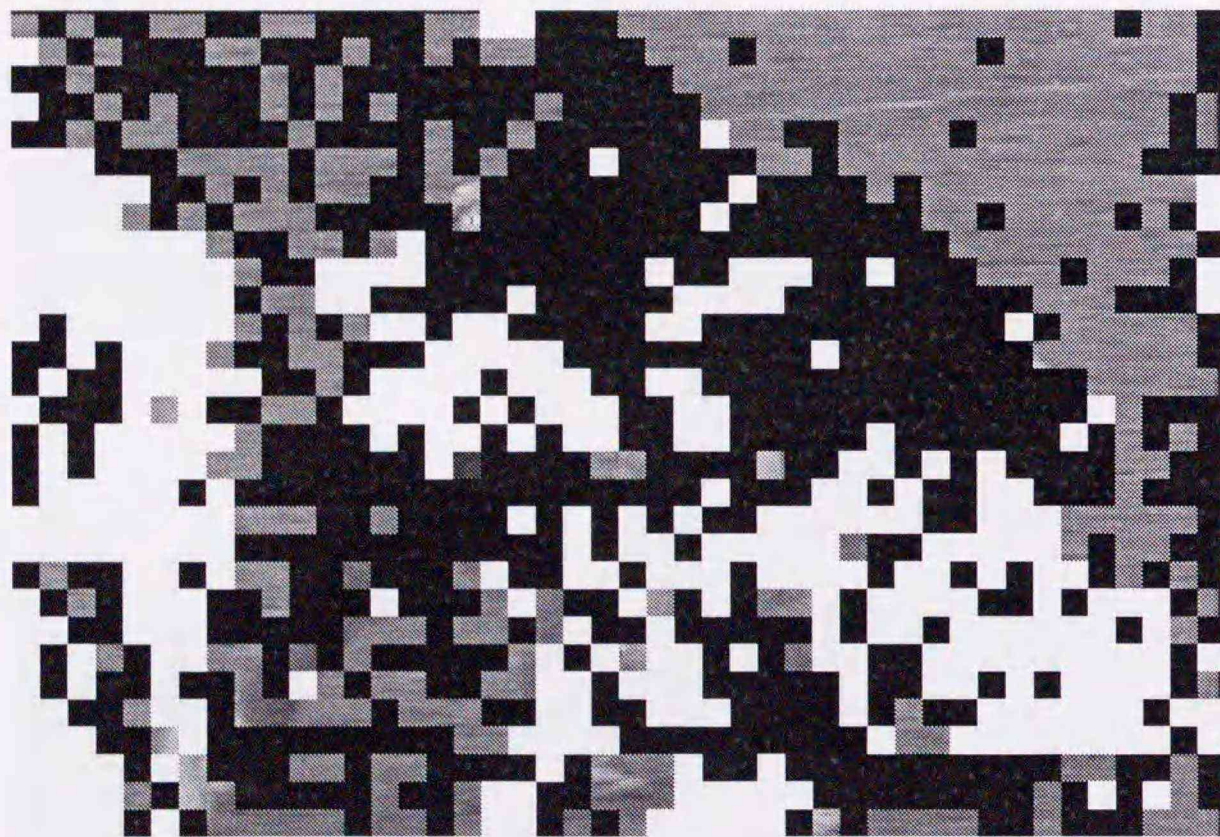


図 5.16: (d):Difference of prediction error between Freq and GA

図 5.17: 動き補償予測画像 (Football)



図 5.18: 原画像 (Flower Garden)



図 5.19: (b): Freq(bitrate=0.068[bit/pel], MSE=124.5)



図 5.20: (c):GA (bitrate=0.067[bit/pel],MSE=116.8)

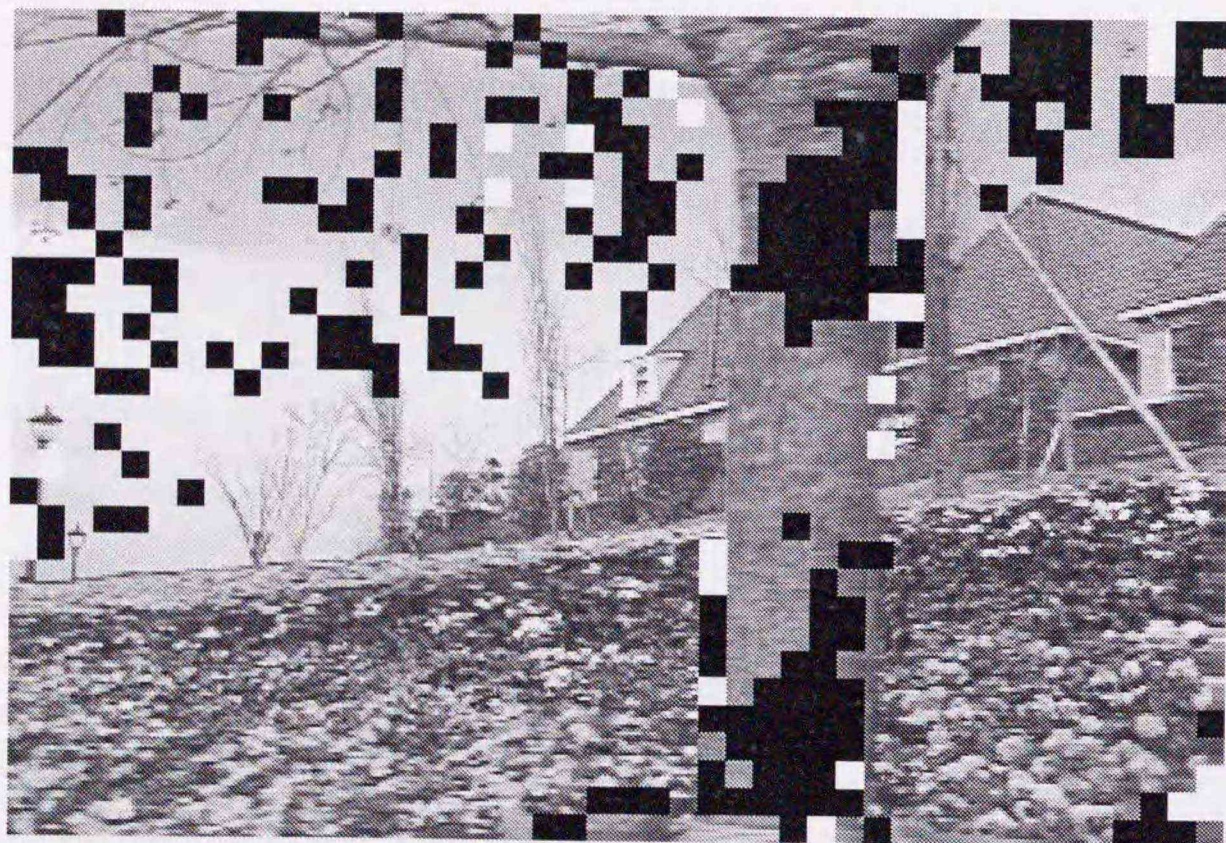


図 5.21: (d):Difference of prediction error between Freq and GA

図 5.22: 動き補償予測画像 (Flower Garden)



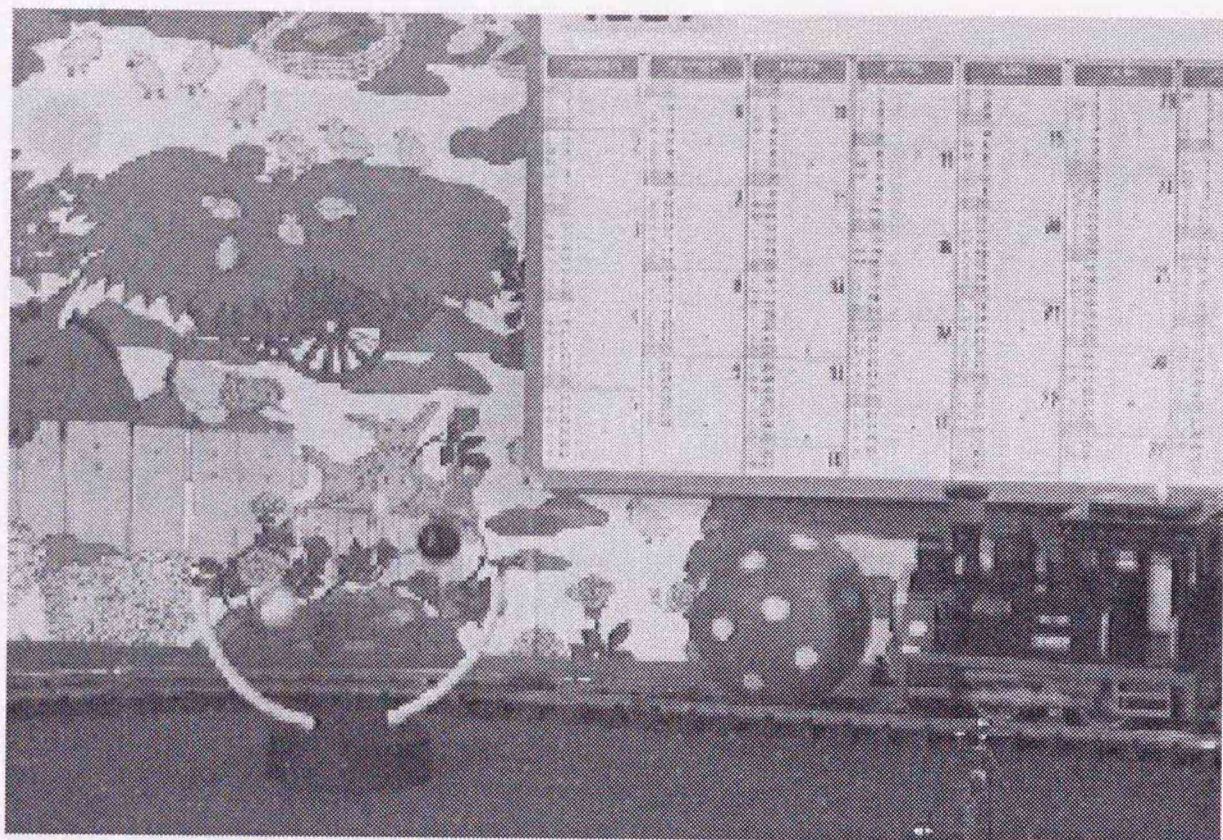


図 5.23: 原画像 (Moblie & Calendar)

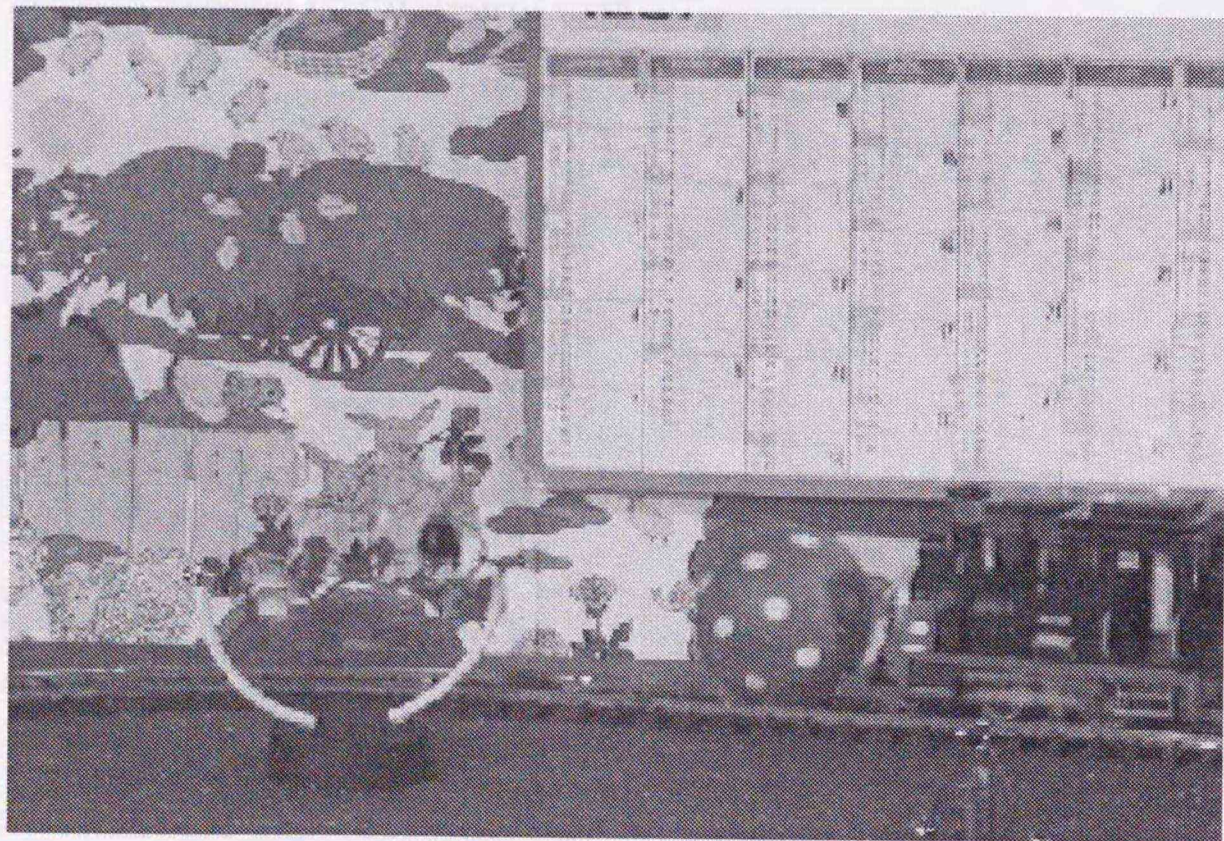


図 5.24: (b):Freq(bitrate=0.073[bit/pel],MSE=94.0)

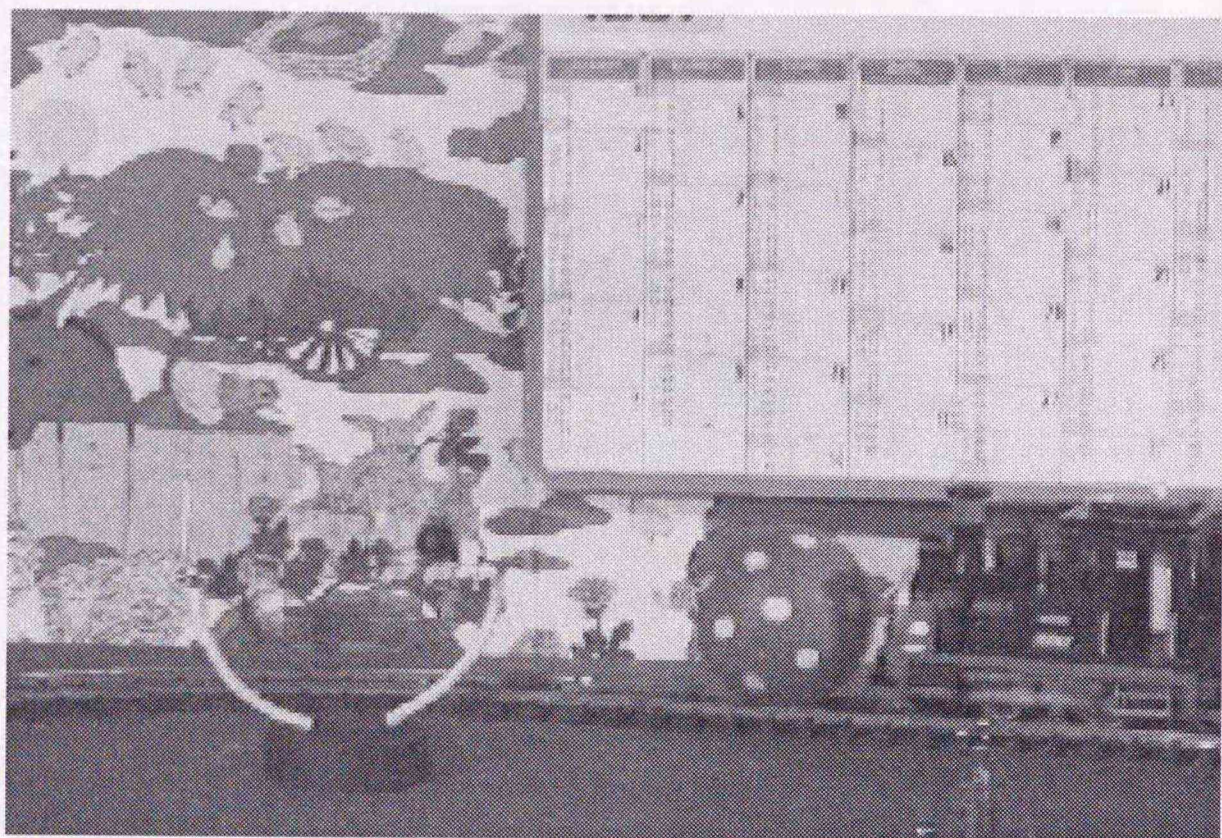


図 5.25: (c):GA (bitrate=0.063[bit/pel],MSE=86.9)

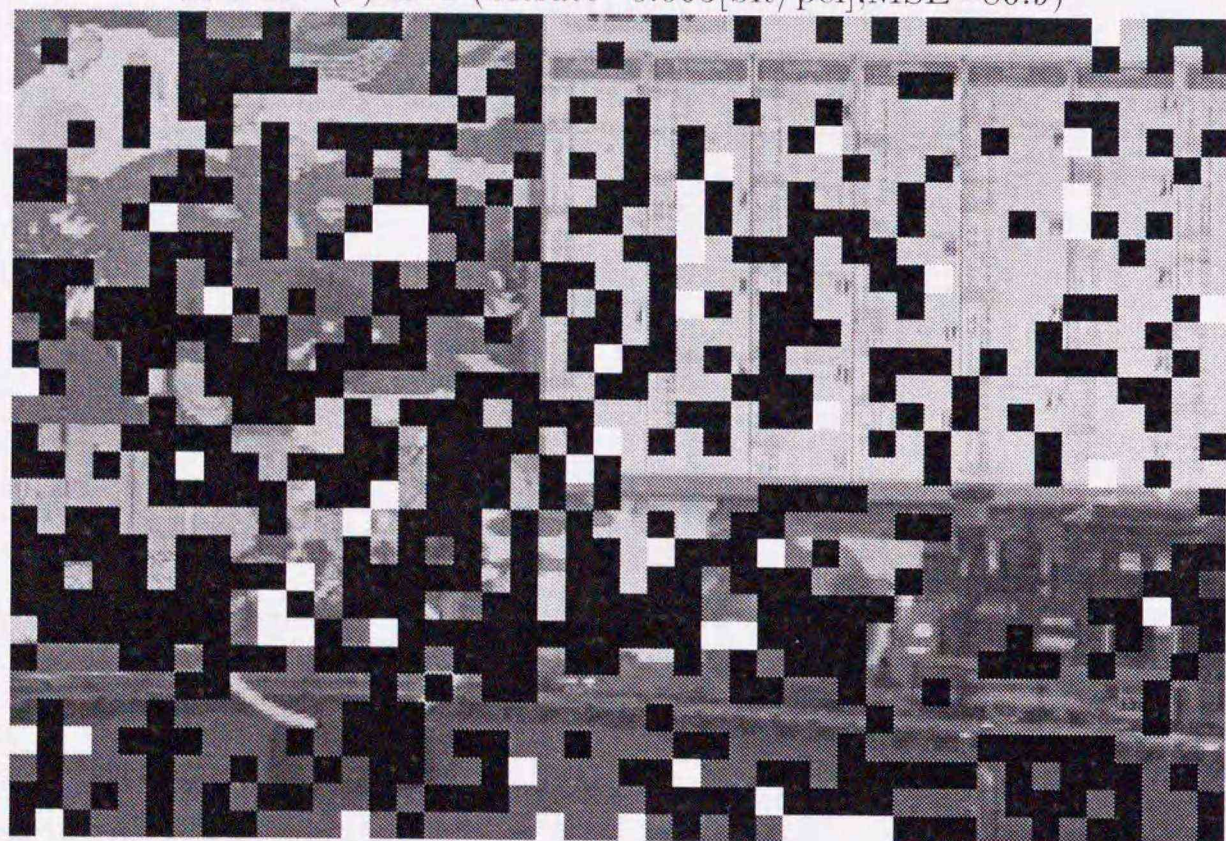


図 5.26: (d):Difference of prediction error between Freq and GA

図 5.27: 動き補償予測画像 (Moblie & Calendar)

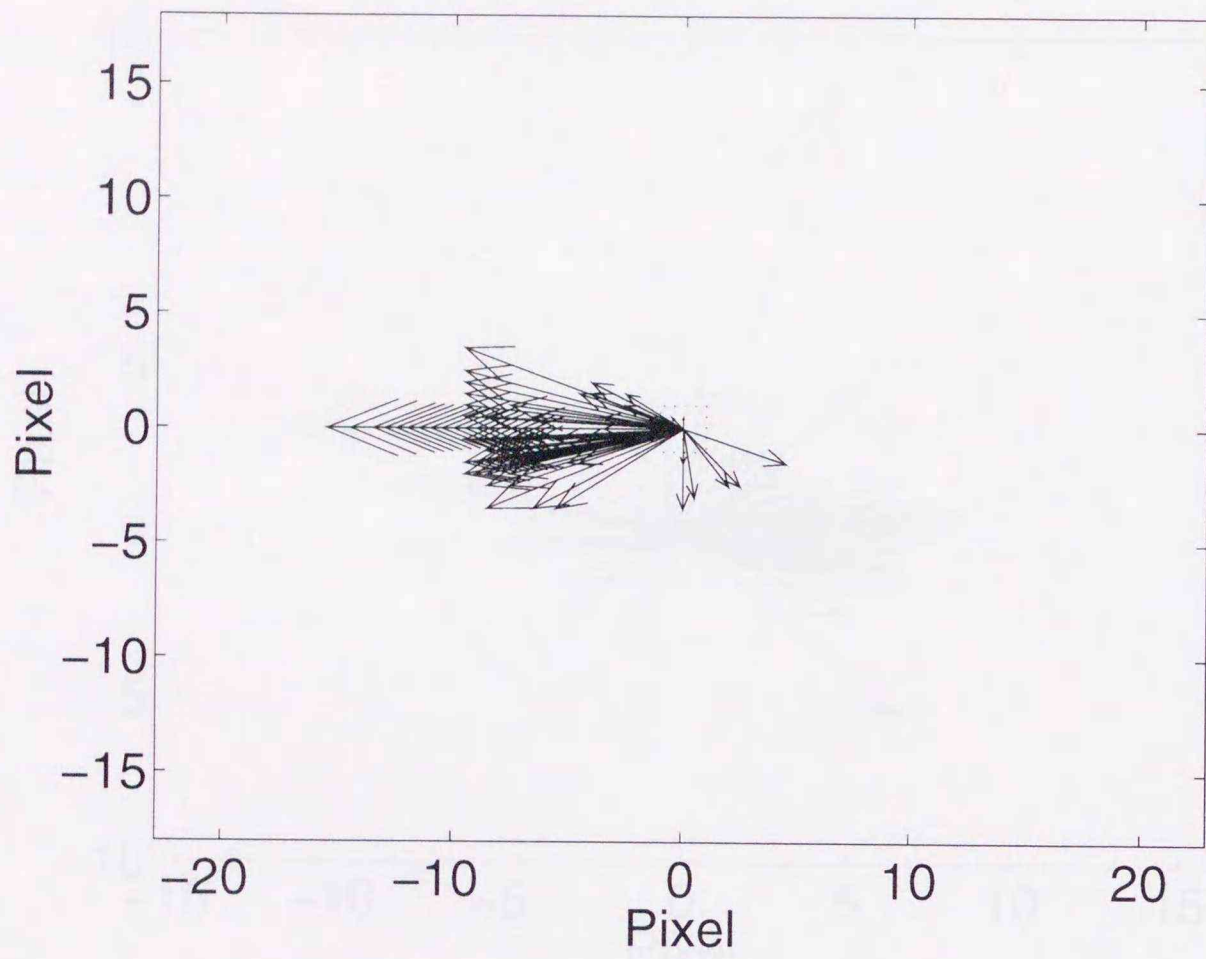


図 5.28: (a): Vectors in motion table of **Freq.** (64 vectors)

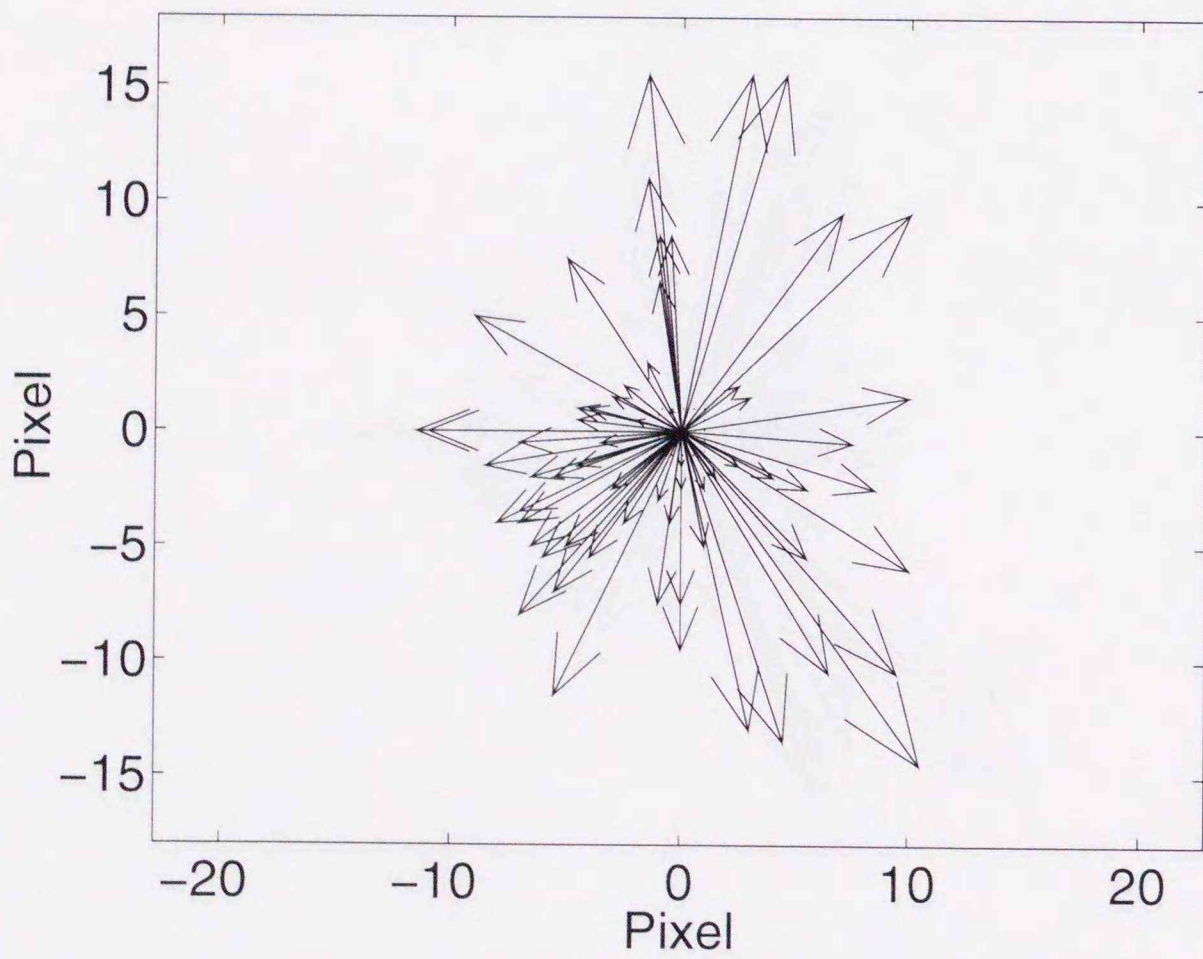


図 5.29: (b): Vectors in motion table of **GA.** (64 vectors)

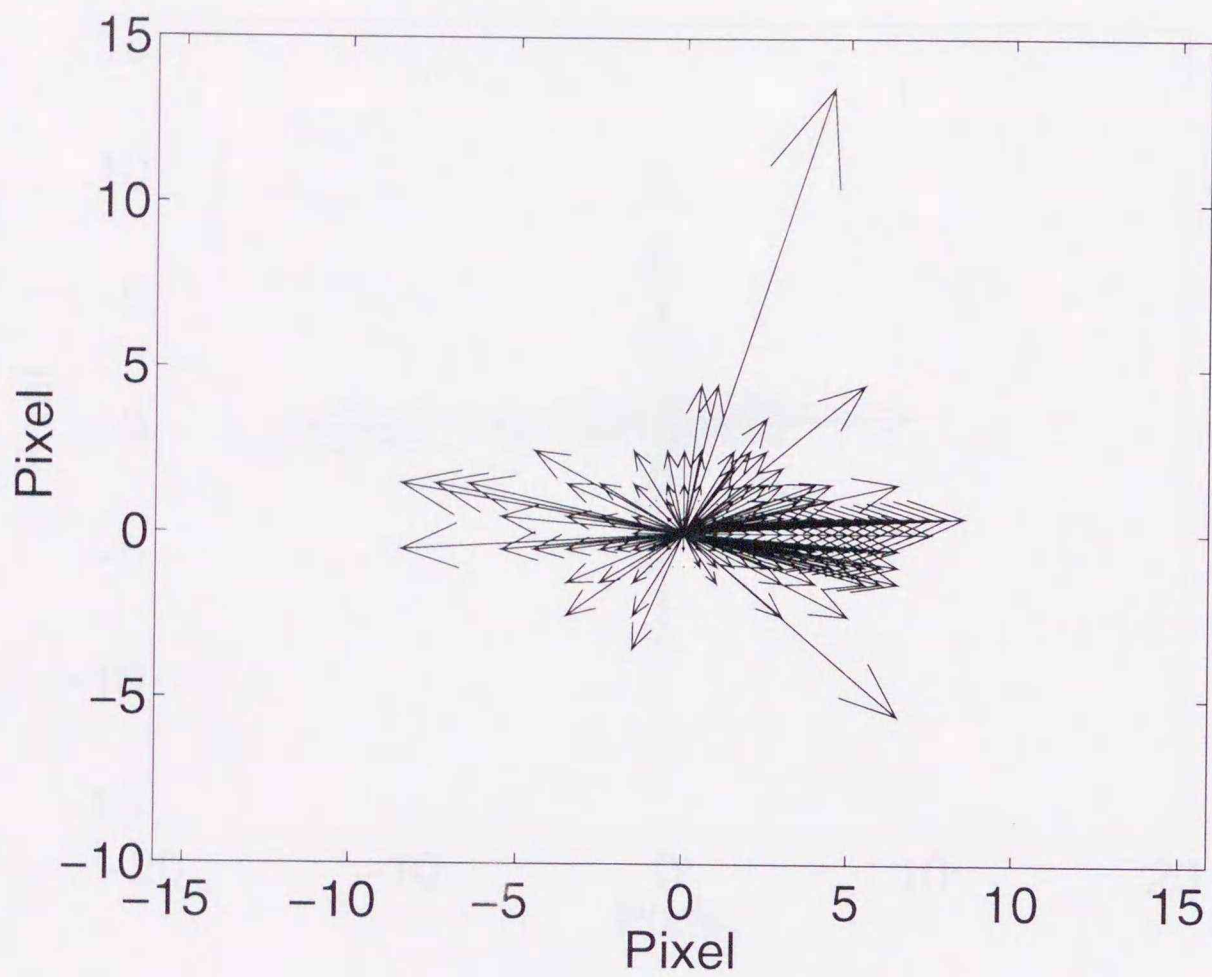


図 5.30: (a): Vectors in motion table of **Freq**, Flower Garden. (128 vectors)

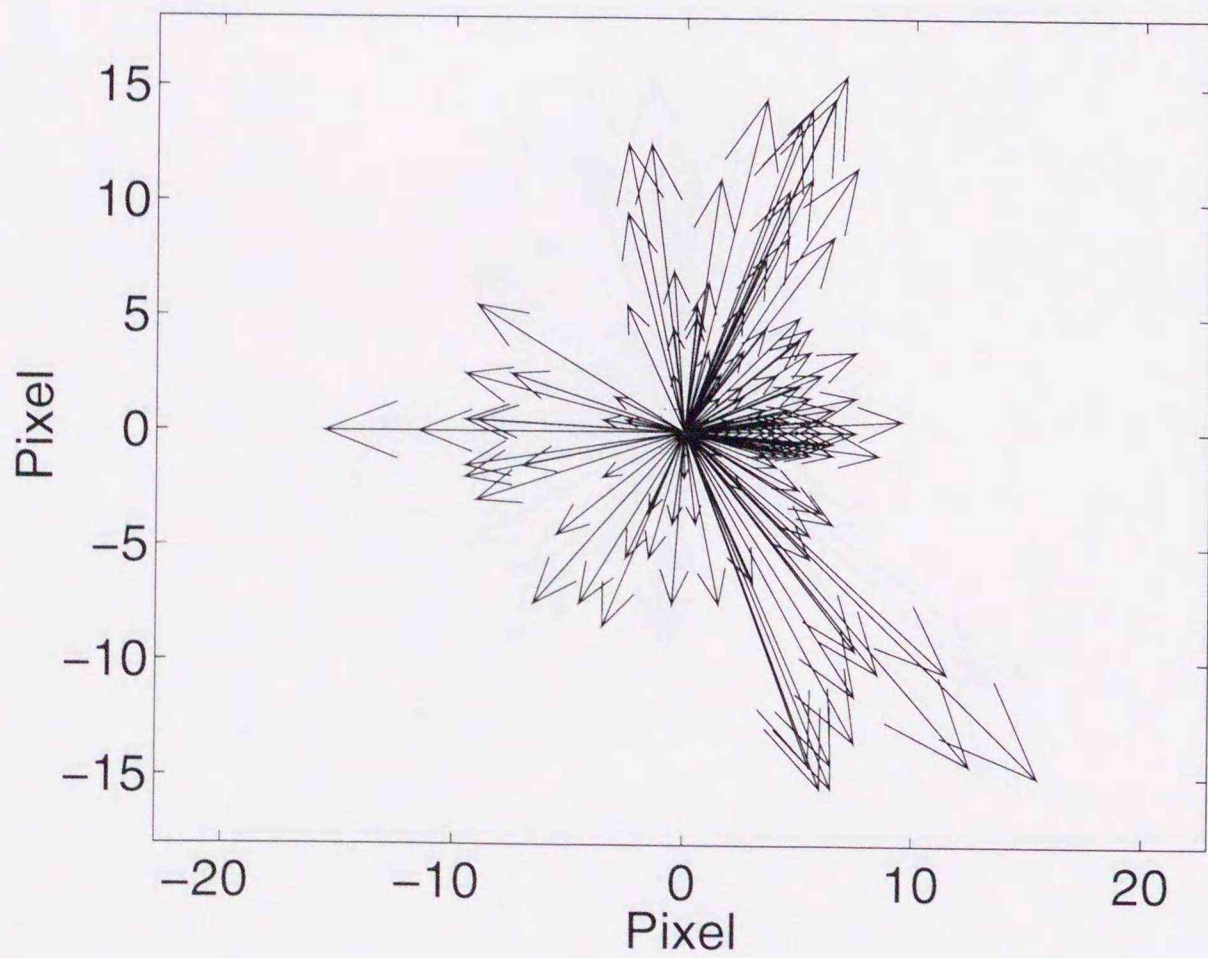


図 5.31: (b): Vectors in motion table of **GA**, Flower Garden. (128 vectors)

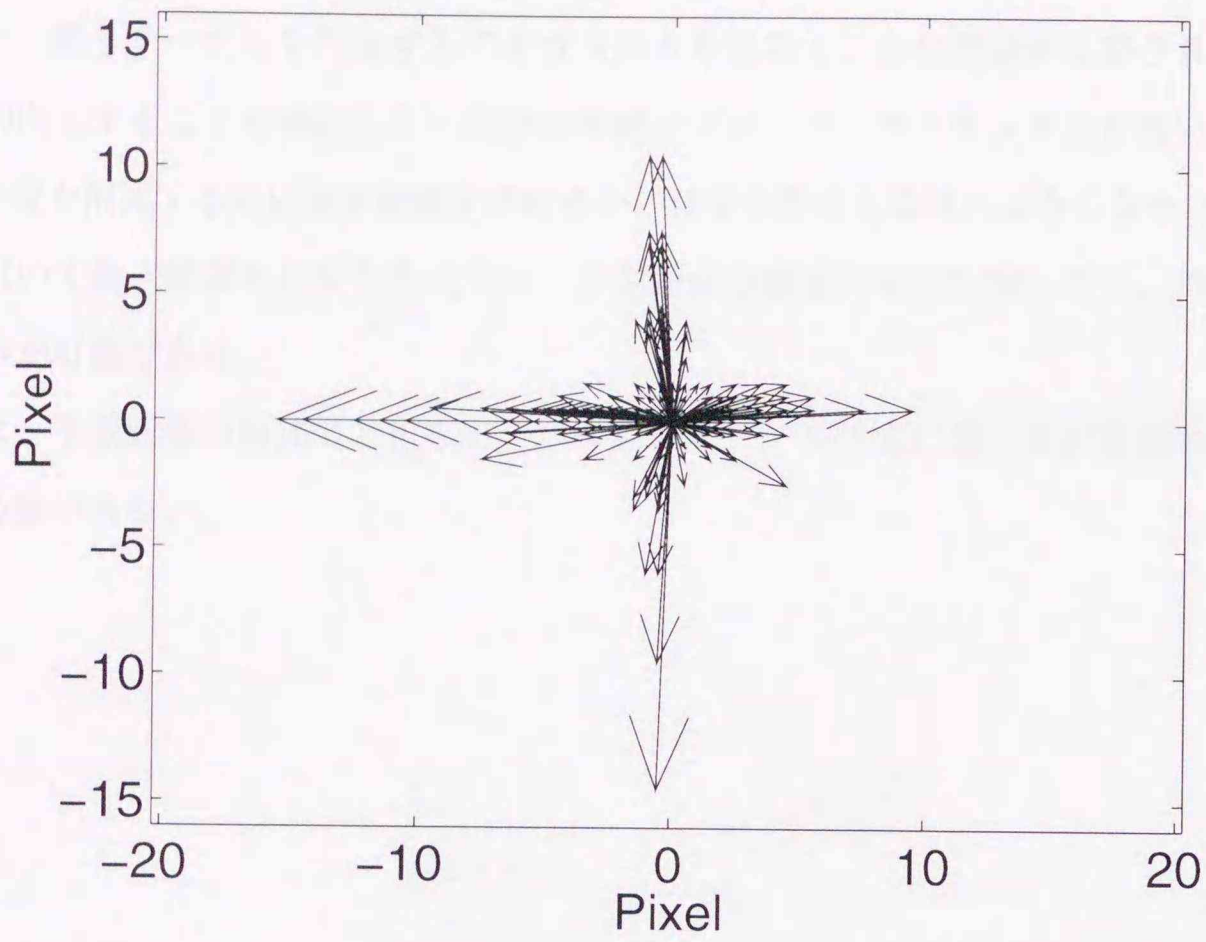


図 5.32: (a): Vectors in motion table of **Freq**, Moblie & Calendar. (128 vectors)

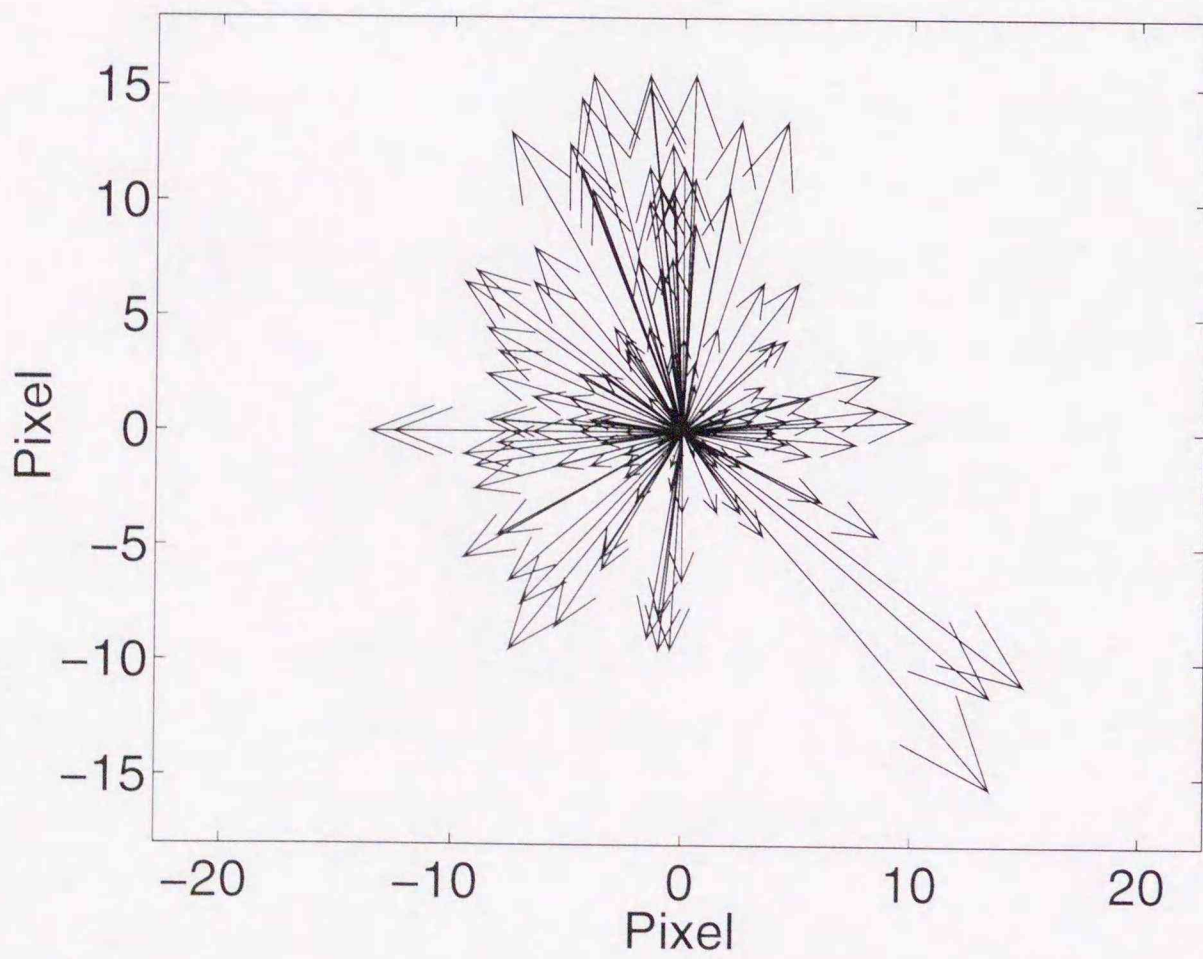


図 5.33: (b): Vectors in motion table of **GA**, Moblie & Calendar. (128 vectors)

するアルゴリズムではない。そこで、GA を用いて動き補償予測誤差が最小になる評価関数により、動きテーブルを作成するアルゴリズムを提案し、この問題が改善され、動き補償精度が向上することを確認した。従来の単純なブロック・マッチング法を用いている場合、符号量を削減するには探索範囲を狭めるか、精度を悪くしなければならなかったが、本手法を用いて動き補償を行なう方式では、予測画像の画質の劣化を抑えつつ、符号量を削減する事が可能である。

今後は、予測画像の画質の劣化を抑えつつ動きテーブル作成に要する計算量の削減を考慮する必要がある。

## 第 6 章 テーブル参照による動き補償 のための動きモデルの検討

### 6.1 はじめに

前章で説明したように、現在通常に行われている動き補償はブロック単位の処理である。この方法は、この後に行われる適応処理や符号化処理がブロック単位に行われることが多いため、それらとの親和性が高い。更に、ブロック・マッチング法による動きベクトルの検出法の性能の高いことと、伝送する動きベクトルに対するビット消費量を考えると、ブロック単位で処理を行なうのもやむを得ないという事情もある。しかしこの方法を用いる場合、隣接ブロックの動きベクトルが異なる時に予測誤差がブロックの境界部で非常に大きくなるという欠点がある。ブロック毎に処理を行なうのはそのブロック内だけの問題ではあるが、ブロック境界部分での予測誤差が原因となる高周波成分の多い予測誤差信号を符号化しなければならない。この問題を解決するため、回転・伸縮を表現できるアフィン動きモデルや透視投影動きモデルがある。これらを用いた動き補償が、従来の平行移動動き補償に対して大きく補償特性を向上させることが検証されている [52] [53][54]。

本章では、動きテーブル参照による動き補償を用いた、動きモデルの比較について述べる。このテーブル参照による動き補償は、符号化システムの設計の時点で、様々な動きを格納したテーブルを作成する。動き補償予測を行なう時には、対象ブロック毎にテーブルを探索し、各ブロック内の画素値の動き補償予測誤差を最小にする動きを選択し動き補償予測を行なうことである。

テーブルを参照して動き補償を行なうこの方式では、テーブルに登録する動きの表現方法は、自由に選択できるという特徴ある。

動きモデルには小領域でのアフィン変換によりブロック境界部分の動きベクトルを推定し動きベクトルの補正を行なう手法 [49] を提案する. また, 画素毎の動きをブロック毎に DCT(discrete cosine transform) を行ない, その変換係数の使用する個数を決定し, IDCT を行なうことにより, ブロック毎の動きを表現する手法を提案 [50] する. これらの動きモデルをテーブル参照における動き補償に適用し, その有効性を検討する.

## 6.2 アフィン動きモデル

### 6.2.1 動きのパラメータ表現

画像中の物体は平行移動のみならず, 回転・伸縮等の動きを持っている. これらを近似表現するために, 各ブロック内の動きを式 (6.1) のアフィン式により近似する.

画像内の位置  $(x, y)$  における動きベクトル  $(u, v)$  が位置  $(x, y)$  によって決定できるものとする. 動きベクトルが  $(u, v)$  は, 1組のパラメータ  $a_0 \sim a_2, b_0 \sim b_2$  を用いて表現できる.

$$\begin{cases} u = a_0 + a_1x + a_2y \\ v = b_0 + b_1x + b_2y \end{cases} \quad (6.1)$$

画素の動きが画素位置  $(x, y)$  のアフィン式で近似できると仮定する. ここで  $a_0 \sim a_2, b_0 \sim b_2$  を動きパラメータと定義する. この動きパラメータより, ズーム・回転・縦横歪み・斜め歪み・パニングの動きが表現できる [36].

### 6.2.2 アフィン変換

画面上の位置  $(x, y)$  における動きベクトル  $(v_x, v_y)$  が位置  $(x, y)$  によって決定できるものとする. 動きベクトル  $(v_x, v_y)$  は1組のパラメータ  $a \sim f$  を用いて以下のように表現できる [36] [44] [45].

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \quad (6.2)$$

(6.2) 式を次のように分解する.

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \frac{a+d}{2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \frac{c-b}{2} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \frac{a-d}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

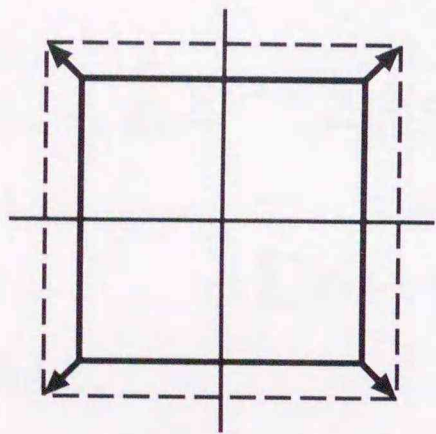


$$\begin{aligned}
& + \frac{b+c}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \\
= & A_1 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + A_2 \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + A_3 \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\
& + A_4 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \tag{6.3}
\end{aligned}$$

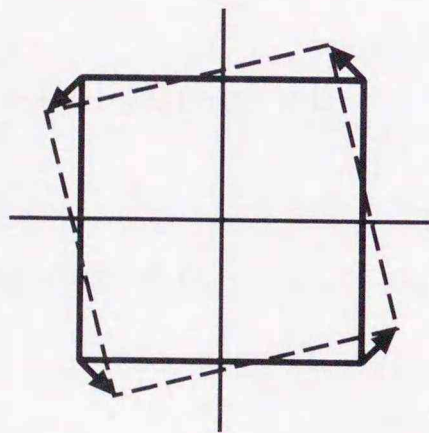
(6.3) 式より, 各項はそれぞれ以下のことを表していることが分かる [44].

- 第1項: ズーム
- 第2項: 回転
- 第3項: 縦横歪み
- 第4項: 斜め歪み
- 第5項: パニンク

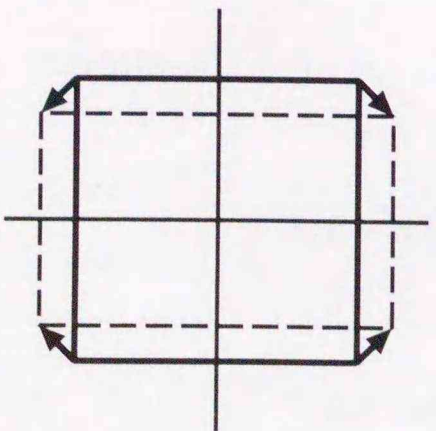
従ってパラメータ  $a \sim f$  あるいは  $A_1 \sim A_4$  を用いてグローバルな動きを表現することができる. 図 6.1 に各パラメータ  $A_1 \sim A_4$  に対応する動きを示す.



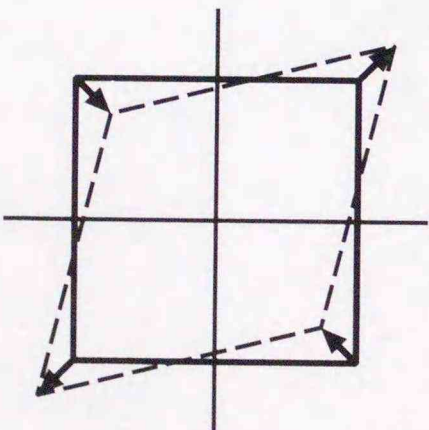
$A_1$ : ズーム



$A_2$ : 回転



$A_3$ : 縦横歪み



$A_4$ : 斜め歪み

図 6.1: 各パラメータに対応する動き

## 6.2.3 パラメータの推定

パラメータ  $a \sim f$  あるいは  $A_1 \sim A_4$  を推定する方式として最小2乗誤差法を用いることができる [44]。(6.2) 式より,

$$v_x = ax + by + e \quad (6.4)$$

$$v_y = cx + dy + f \quad (6.5)$$

(6.4) より,

$$r = v_x - (ax + by + e) \quad (6.6)$$

と置き,

$$\sum r^2 = \sum \{v_x - (ax + by + e)\}^2 \quad (6.7)$$

$$\frac{\partial}{\partial a} \sum r^2 = \sum 2\{v_x - (ax + by + e)\}(-x) = 0 \quad (6.8)$$

$$\sum (v_x x - ax^2 - bxy - ex) = 0 \quad (6.9)$$

平均値で置き換えて,

$$\overline{v_x x} - a\overline{x^2} - b\overline{xy} - e\overline{x} = 0 \quad (6.10)$$

同様にして  $b, e$  での偏微分により,

$$\overline{v_x y} - a\overline{xy} - b\overline{y^2} - e\overline{y} = 0 \quad (6.11)$$

$$\overline{v_x} - a\overline{x} - b\overline{y} - e = 0 \quad (6.12)$$

ここで領域として原点对称な領域を考えると,  $\overline{x} = 0, \overline{y} = 0, \overline{xy} = 0$  であるから,

$$a = \overline{v_x x} / \overline{x^2} \quad (6.13)$$

$$b = \overline{v_x y} / \overline{y^2} \quad (6.14)$$

$$e = \overline{v_x} \quad (6.15)$$

$v_y$  に関しても同様の操作を行い,

$$c = \overline{v_y x} / \overline{x^2} \quad (6.16)$$

$$d = \overline{v_y y} / \overline{y^2} \quad (6.17)$$

$$f = \overline{v_y} \quad (6.18)$$

### 6.3 動きテーブル参照による動き補償

従来の動画像符号化における動き補償方式では、画素ごとに検出された動きベクトルの量子化誤差が最小のパラメータを選択していたが、ブロック内の予測誤差を小さくするようにパラメータを選択した方が良好な予測画像が得られることが確認されている [51].

本文では、動きテーブル参照による動き補償の動きパラメータを動き補償予測誤差が最小なるように動きテーブル内を探索する。動きテーブルとは、ブロック内の動きを表す動きパラメータが格納されているものである。動きテーブル内の動きパラメータの組には識別番号を与え、このエントリー数が符号量に影響する。

動きテーブルの中から、個々の入力画像ごとに各ブロック内の動き補償予測誤差が最小になる代表動きパラメータを動きテーブル内より選択することにより動き補償を行う。この方式を、動きテーブル参照による動き補償方式と呼ぶ。動きテーブル内に収められている動きのみを使用して動き補償を行なう。このため動きテーブルの作成法により、予測画像の画質に大きく影響を及ぼす。

#### 6.3.1 画素ごとの動きを用いた動きテーブル作成法

動画像の画素ごとの動きベクトルの検出結果から動きテーブルを作成する [51].

### 6.3.2 動きベクトルの階層的検出法

動きベクトルを階層的に検出するこの方式の特徴は，上位階層では大きな動きを把握でき，下位階層では細かい動きを捉えられることである．この性質から，上位階層での動きベクトルの検出結果を下位階層の動きベクトル検出に利用することによって雑音や画像の性質に関係なく高い精度で動きベクトルを検出できる [37].

画素ごとの動きを用いた動きテーブル作成法では，階層画像を用いた画素ごとの動きベクトル階層的検出法を用いる．本文では，3階層， $\pm 16$ 画素を探索範囲とした．

### 6.3.3 動きパラメータ推定

動きパラメータ  $a_0 \sim a_2$ ,  $b_0 \sim b_2$  の決定法について述べる．階層画像を用いて検出された画素ごとの動きベクトルの実測値  $(u, v)$  から最小2乗法を用い，6.2.3に述べたように動きパラメータを決定する．

$$\begin{aligned} a_0 &= \bar{u}, & a_1 &= \frac{\overline{ux}}{x^2}, & a_2 &= \frac{\overline{uy}}{y^2} \\ b_0 &= \bar{v}, & b_1 &= \frac{\overline{vx}}{x^2}, & b_2 &= \frac{\overline{vy}}{y^2} \end{aligned} \quad (6.19)$$

ただし， $\bar{\cdot}$  はブロック内での平均値を表し，画像を分割するブロック (正方形) 内の座標  $(x, y)$  の原点をブロックの中心にとる．推定した動きパラメータの結果を図 6.2 に示す．

### 6.3.4 ベクトル量子化

画素ごとの動きを用いた動きテーブル作成法では，ブロックごとの動きベクトルをベクトル量子化を行ない代表ベクトルを求める．この代表ベクトルが動きテーブルである．

動きパラメータを性質の異なる (図 6.2 参照) 2つのグループに分け，動きパラメータの平行移動を表現している  $a_0, b_0$  を 2次元ベクトルとして，また，回転・伸縮等を表現している  $a_1, a_2, b_1, b_2$  をそれぞれを 4次元ベクトルとみなしベクトル量子化器を作成する．

$$\begin{cases} u = \boxed{a_0} + \boxed{a_1x + a_2y} \\ v = \boxed{b_0} + \boxed{b_1x + b_2y} \end{cases} \quad (6.20)$$

ベクトル量子化器の作成は，LBG アルゴリズム [39] により行なう．その際，動画像データ数フレームから得た動きパラメータをトレーニング系列として用いる．ベクトル量子化

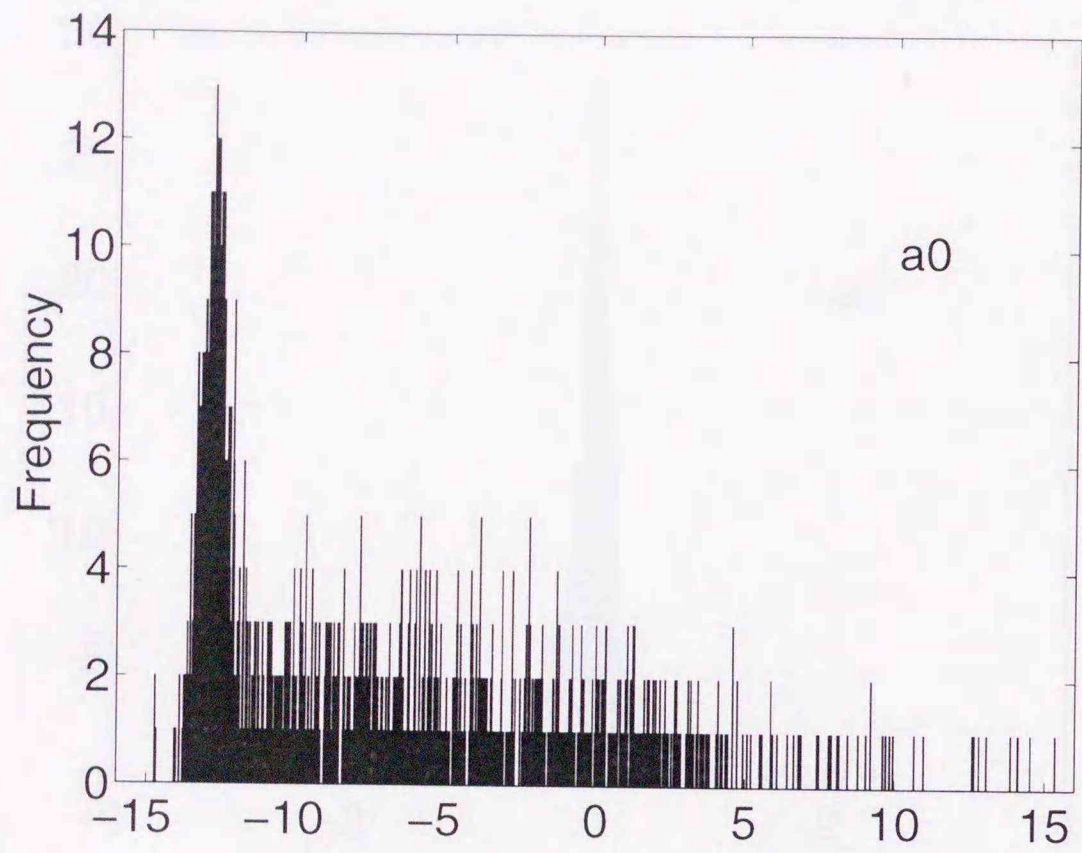


図 6.2: 動きパラメータのヒストグラム  $a_0$ (Football)

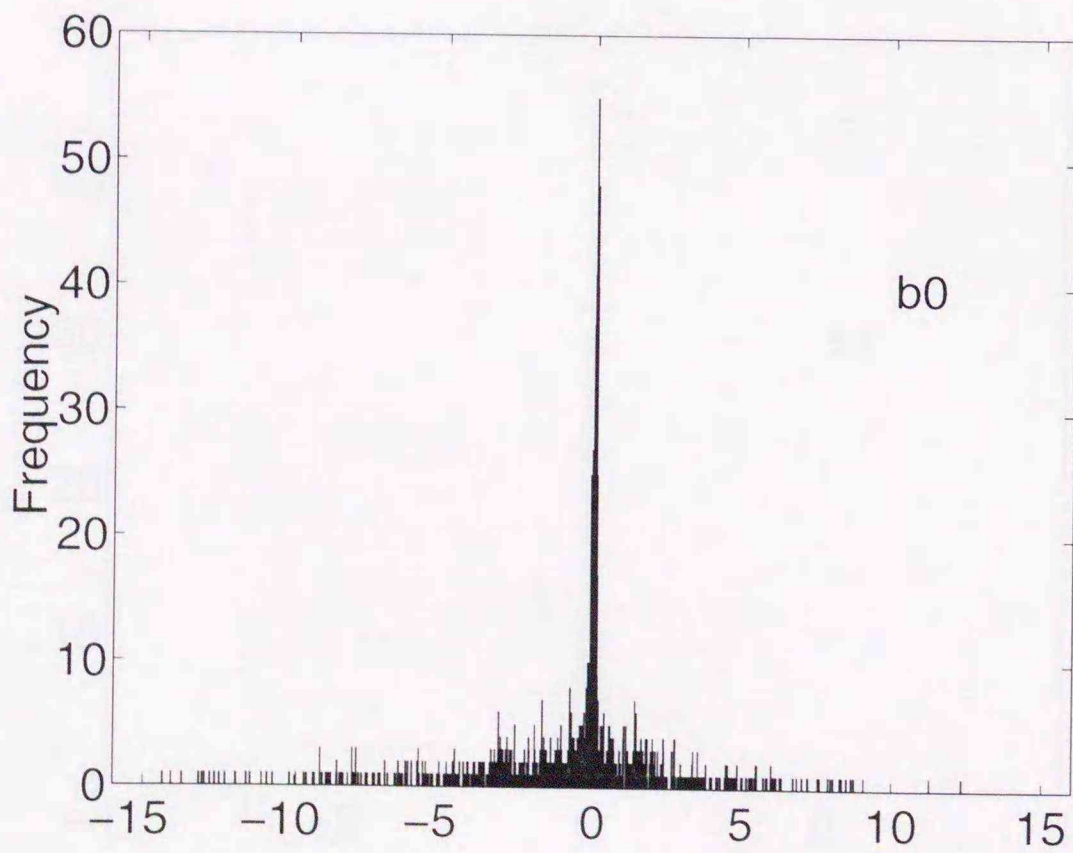


図 6.3: 動きパラメータのヒストグラム  $b_0$ (Football)

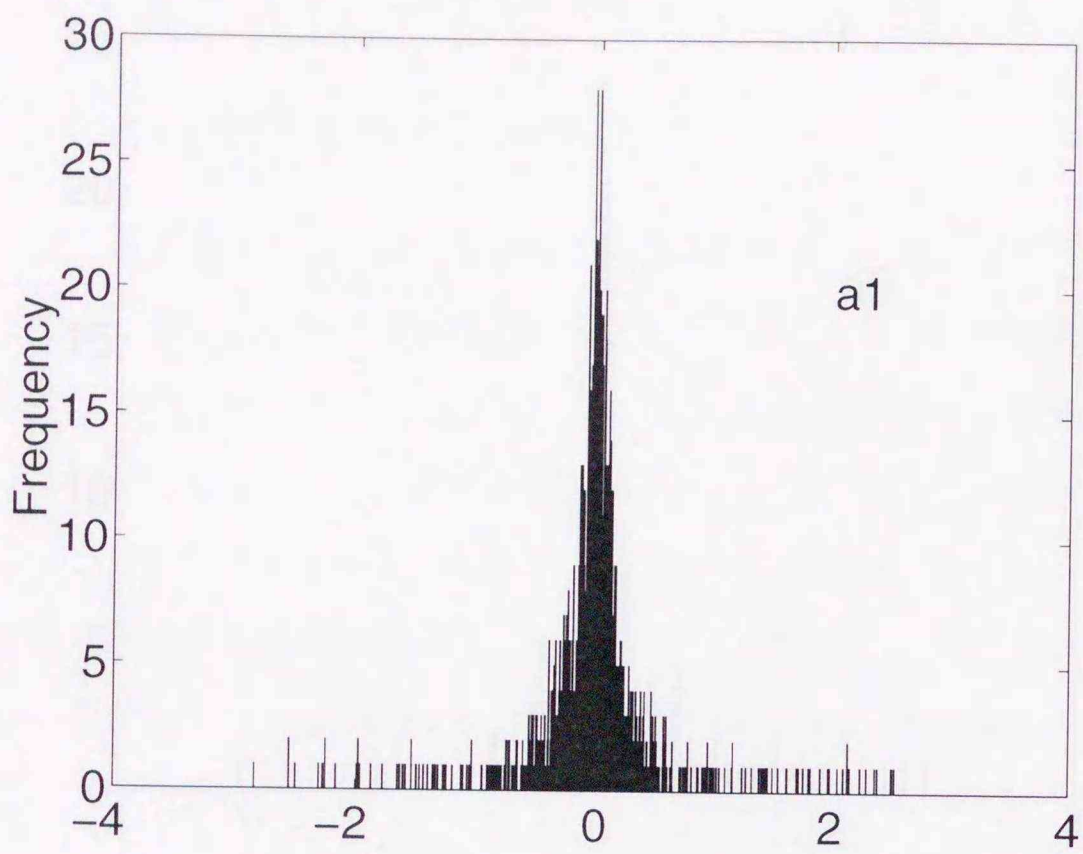


図 6.4: 動きパラメータのヒストグラム  $a_1$ (Football)

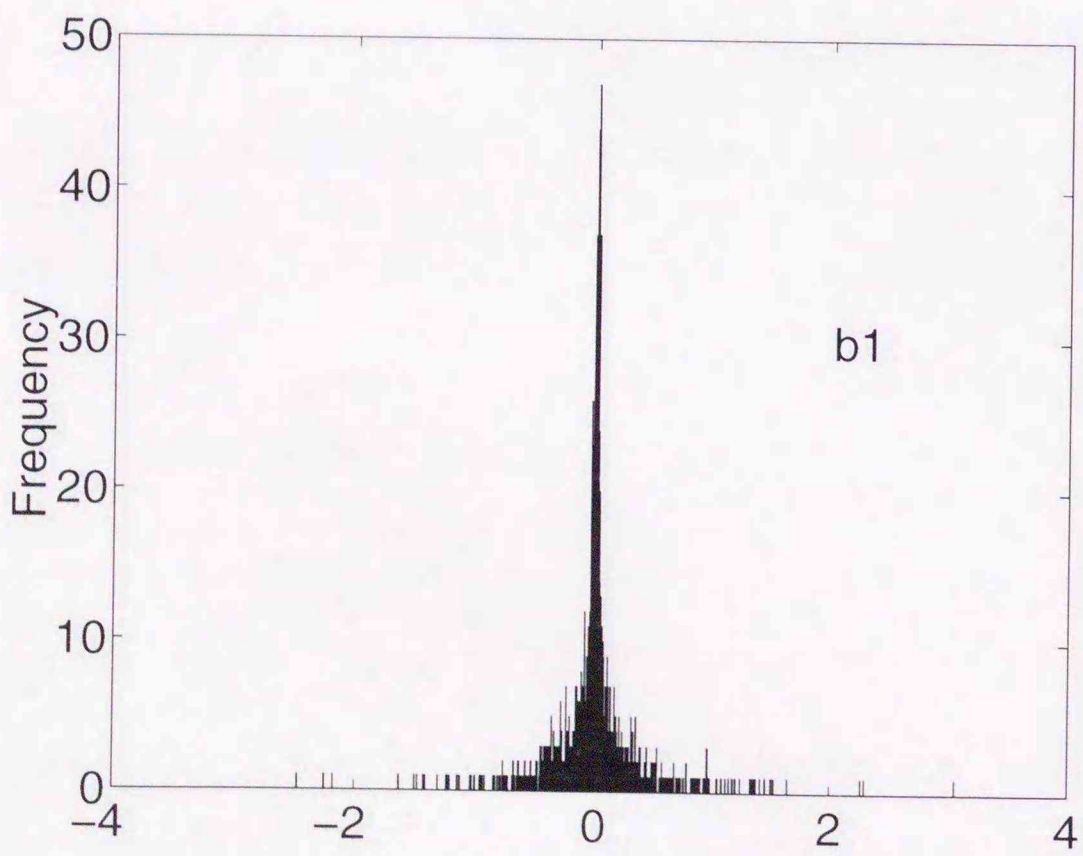


図 6.5: 動きパラメータのヒストグラム  $b_1$ (Football)

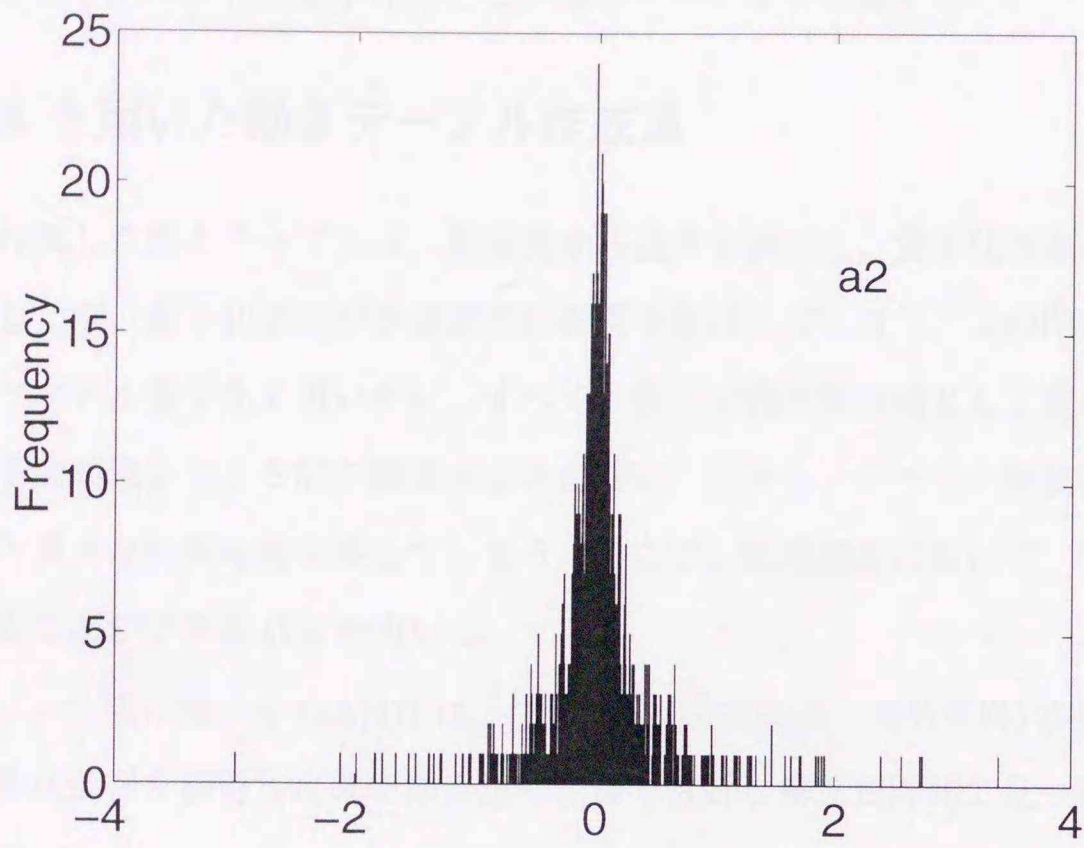


図 6.6: 動きパラメータのヒストグラム  $a_2$ (Football)

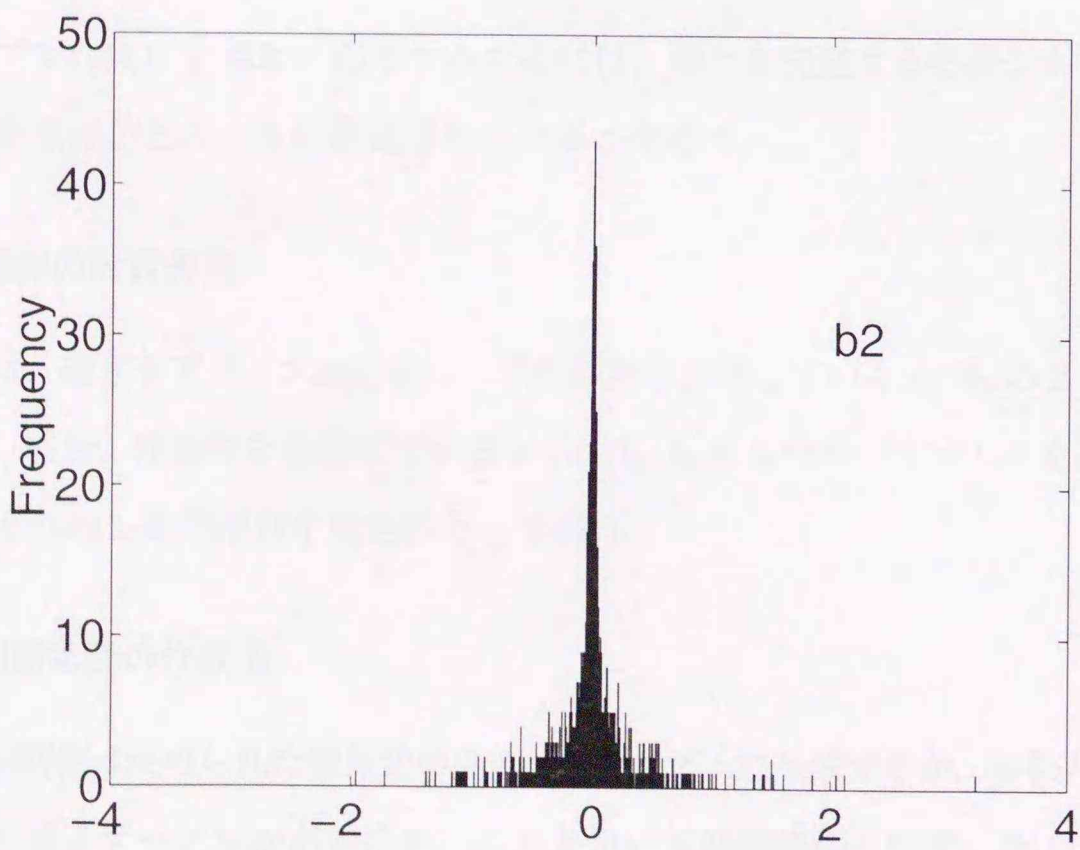


図 6.7: 動きパラメータのヒストグラム  $b_2$ (Football)

によって得られた動きパラメータを動きテーブルと呼ぶ。 $a_0, b_0$ を Table-1,  $a_1, a_2, b_1, b_2$ を Table-2 として、ベクトル量子化を行ない動きテーブルを作成する。

## 6.4 GA を用いた動きテーブル作成法

6.3.1. で作成した動きテーブルは、動画像から動きを検出し、量子化誤差が最小になるように作成したが、量子化誤差が予測誤差に影響を及ぼしてしまう。この問題を解消するためには、ベクトル量子化を用いずに、すべての量子化値を解候補として定義し、その中から良好な予測画像を与える解を探索する方が良い。しかし、すべての解候補について探索を行なうと多大な計算時間を要してしまう。そこで、広域探索において、実用解を短時間で探索することができる GA を用いる。

動きテーブル作成に用いる GA[47] は、生物進化(選択淘汰・突然変異)の原理に着想を得ており、多点情報を利用した確率探索法の一種で良好な解を短時間で見つける事ができるという利点がある。

GA は初期集団の作成を行なった後、適応度の評価、選択、交叉および突然変異という一連の処理を終了条件まで繰り返す。

### 6.4.1 提案手法のための GA の設計

動きテーブル作成に GA に応用するためには、項目を決定する必要がある。提案手法について、各項目がどのように設定されているかを示す。

#### 6.4.1.1 個体の形質表現

本手法では、動きをアフィン式近似し、平行移動を表現している  $a_0, b_0$  の2つのパラメータを、また、回転・伸縮等を表現している  $a_1, a_2, b_1, b_2$  を4つのパラメータを遺伝子として用い、それぞれの1次元配列を染色体として扱う。

#### 6.4.1.2 初期集団の作成法

一般には初期値は決められた個体数の染色体をランダムに生成するが、本手法では、6.3.1. より作成した動きテーブルがあるため、これを用いて初期集団を作成した。



#### 6.4.1.3 適応度の評価

本手法で用いる適応度は得られた動きテーブルを用いて動き補償を行ない、その予測画像の平均2乗誤差 (Mean Square Error:MSE) により評価する。

$$fitness = MSE \quad (6.21)$$

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \hat{x}_{ij})^2 \quad (6.22)$$

ここで、フレームサイズを  $M \times Npels$  とし、また、 $x_{ij}$ ,  $\hat{x}_{ij}$  はそれぞれ原フレーム、動き補償予測画像の画素濃度を示す。

#### 6.4.1.4 選択淘汰

選択には適応度の高い20%の個体において、適応度比例戦略にエリート保存戦略 [12] を組み合わせて用いる。

この方式では、適応度の大きい個体が次世代の個体として選ばれる可能性が高いが、適応度が低い個体でも次世代の個体として選ばれる可能性が残されることになる。これは、適応度が大きい個体だけを残すと、生物集団の収束性が高まり、局所的な最適解にとらわれやすくなるためである [16]。

エリート保存戦略とは、集団中で適応度の最も高い個体をそのまま次世代に残す方法である。この方法は、その時点で最も良い解が遺伝的操作で破壊されないという利点がある。

#### 6.4.1.5 交叉

交叉は、二つの親の染色体を組み替えて子の染色体を作る操作である。交叉の一般的なものとして、一点交叉、多点交叉、一様交叉 [12] などが存在する。本手法では、解候補が全解空間に及ぶように一様交叉と内分点、外分点を組み合わせて用いた。

一様交叉とは、交叉時にマスクをかけてそれによってどちらかの親の遺伝子を受け継ぐかを決定する方法である。

#### 6.4.1.6 突然変異

突然変異は、遺伝子を一定の確率で変化させる操作である。突然変異はあまり大きな変異確率に設定するとスキマタが破壊されるためランダム・サーチと化してしまうが、突然変異がない場合は、初期の遺伝子の組み合わせ以外の空間を探索することができない。

本手法では突然変異確率は3%とした。正規分布の乱数を用いて変化させ遺伝子を決定した。係数の変化量は、動きパラメータに対して、実験を行ない動きパラメータの取り得る値を調べ、その $\pm 3\sigma$ を上限値とした。ここで、 $\sigma$ は標準偏差を示す。

#### 6.4.1.7 探索終了条件

探索終了時期は実験を行ない最大適応度が30世代改善されない場合はそれ以上進化しなくなる様子を確認した。本手法では各世代における個体の最大適応度が30世代改善されない場合終了とした。

#### 6.4.1.8 個体数

個体数は、多い程探索点が多くなるため解候補の多様性が高くなり、良い結果が得られるが、探索時間を長く要する。本手法では個体数を40に定めた。

### 6.5 実験および考察

提案方式の有効性を確認するため、実際に動画像を用いて実験を行なう。以下に示す4方式についてそれぞれ各隣接フレーム間において実験を行なった。画像を分割するブロックサイズは、 $8 \times 8$ の固定サイズで行なう。用いる画像は、Football, Flower Garden, Mobile & Calendarの3種類(画像サイズは $352 \times 240\text{pels}$ )のY信号(8bit階調)である。

各画像の0~1フレームを用い、各々の画像に対して動きテーブルを前述したように作成した。Table-1とTable-2の与えるビット数は、それぞれ使用する比率を変化させ実験を行ない、MSEが最も良い比率を使用した。

#### 6.5.1 動き補償予測誤差による予測性能の比較

各方式を用いて実験を行なった。(i)BM: 平行移動モデルによるブロック・マッチング法を用いて動き補償(精度1画素)

(ii)Affine: ブロック・マッチングにより、平行移動パラメータを1画素単位で探索し、確定したの後に、ブロックを範囲 $\pm 18$ 度、ステップ6度で回転させ、その中から予測誤差の最小の角度を用いて動き補償を行なう。

(iii)LBG: 画素ごとの動きから作成した動きテーブルを用いて、動きテーブル参照による動き補償(6.3.1.参照)

(iv)GA : GA を用いて評価関数に予測画像の MSE として, 動きテーブルを作成し, 動きテーブル参照による動き補償 (6.4. 参照)

符号化シミュレーションは, (i)BM と (ii)Affine では, 平行移動ベクトル  $(t_x, t_y)$  を, 水平1次元 DPCM により符号化する. (ii)Affine の回転角  $\theta$  については, 60~80% のセグメント "0" が選ばれている. このため,  $\theta$  には, 可変長符号化を割り当てる. (iii)LBG, (iv)GA では, 動きパラメータ識別番号  $(i)$  を可変長符号を割り当てて符号化する.

シミュレーション結果を図 6.8(a)~(c) に示す.

従来法である (i)BM, (ii)Affine より動きテーブル参照による動き補償を用いた (iii)LBG, (iv)GA の方が全テスト画像で MSE が減少した. この実験結果からブロック・マッチング法より動きテーブル内から動き補償予測誤差が最小になる動きを選択して動き補償した方が良い結果が得られることがわかった.

### 6.5.2 GA による動きテーブルの特色

図 6.8 より, 動きテーブルを作成するアルゴリズムの違いにより, (iv)GA の全テスト画像で (iii)LBG より MSE が減少している. (iii)LBG は, 画素ごとの動きから求められる量子化誤差が最も小さい動きテーブルを用いている. しかし, 画素ごとの動きベクトルから求めた動きパラメータをベクトル量子化しているため, 予測画像に量子化誤差の影響があるものと考えられる. この量子化誤差の影響を軽減するように (iv)GA では, 予測画像に量子化誤差を与える影響を考慮しながら, GA を用いて評価関数に予測画像の MSE を用いて動きパラメータを決定し, 動きテーブルを作成している. この動きテーブルを用いて動き補償を行なうことにより (iii)LBG よりも, 予測画像の MSE が減少し予測誤差が軽減されていることがわかる.

図 6.11(b)~(d) に動き補償予測画像を示す.

図 6.11(d) は, (iii)LBG と (iv)GA でブロックごとの原画像とのそれぞれの予測誤差を求め, (iv)GA が小さい方が白く, 大きい方が黒で示してある. この時の白いブロックは 106 個, 黒いブロックは 58 個である. また, 予測誤差が同じブロックは予測画像を示す. この図 6.11(d) より, 提案手法 (iv)GA を用いることによって, 動き大きなブロックの予測誤差が (iii)LBG を用いて動き補償を行なうよりも減少していることがわかる. つまり, (iv)GA を用いた動き補償行なうことにより, 予測性能の向上し, 予測誤差の大きなブロックが減

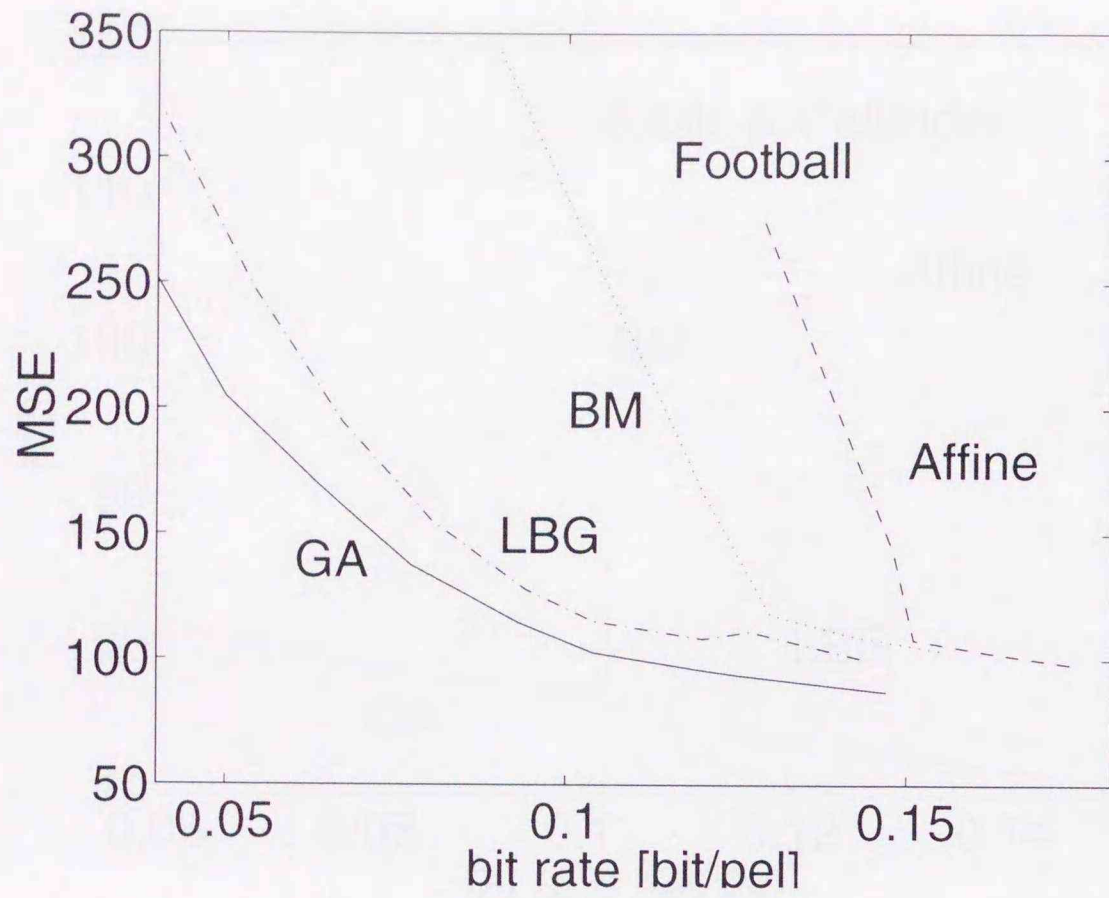


図 6.8: 符号量と MSE(Football)

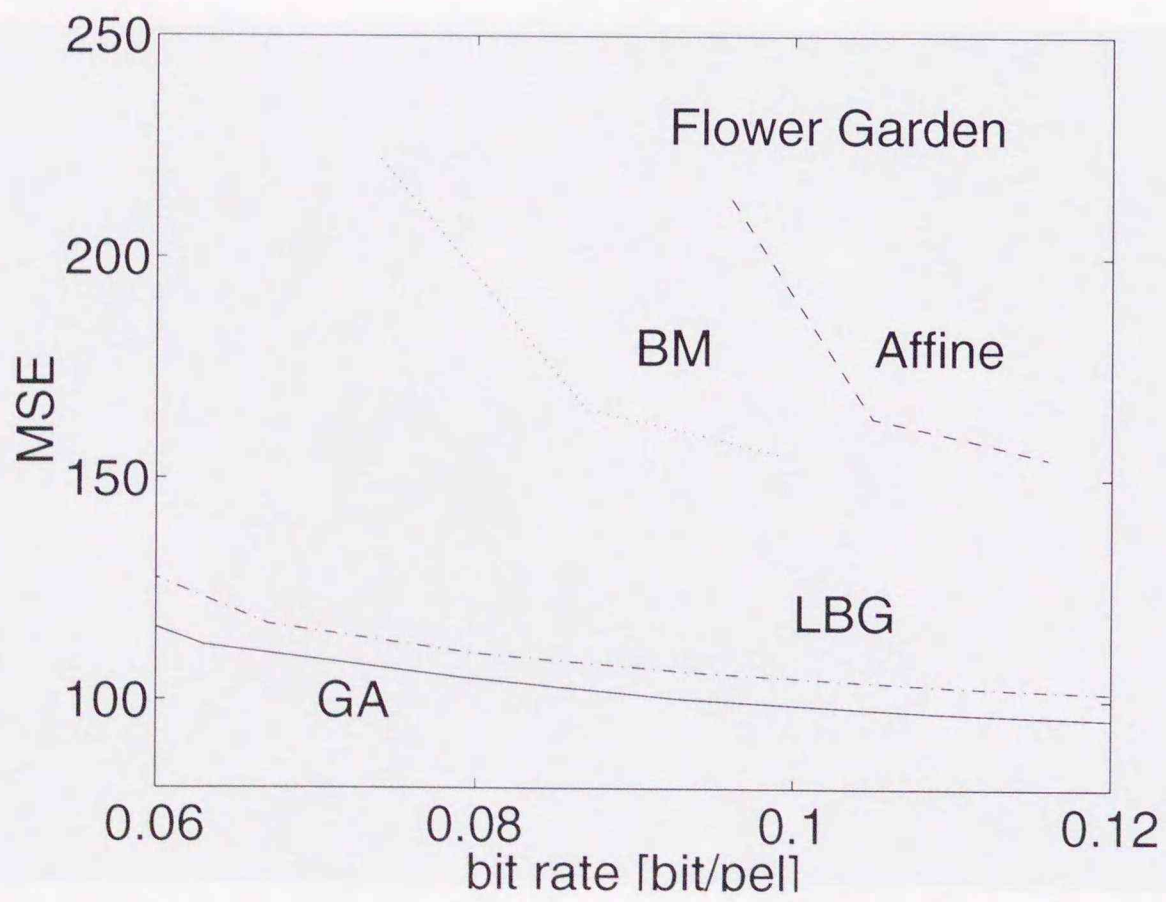


図 6.9: 符号量と MSE(Flower Garden)

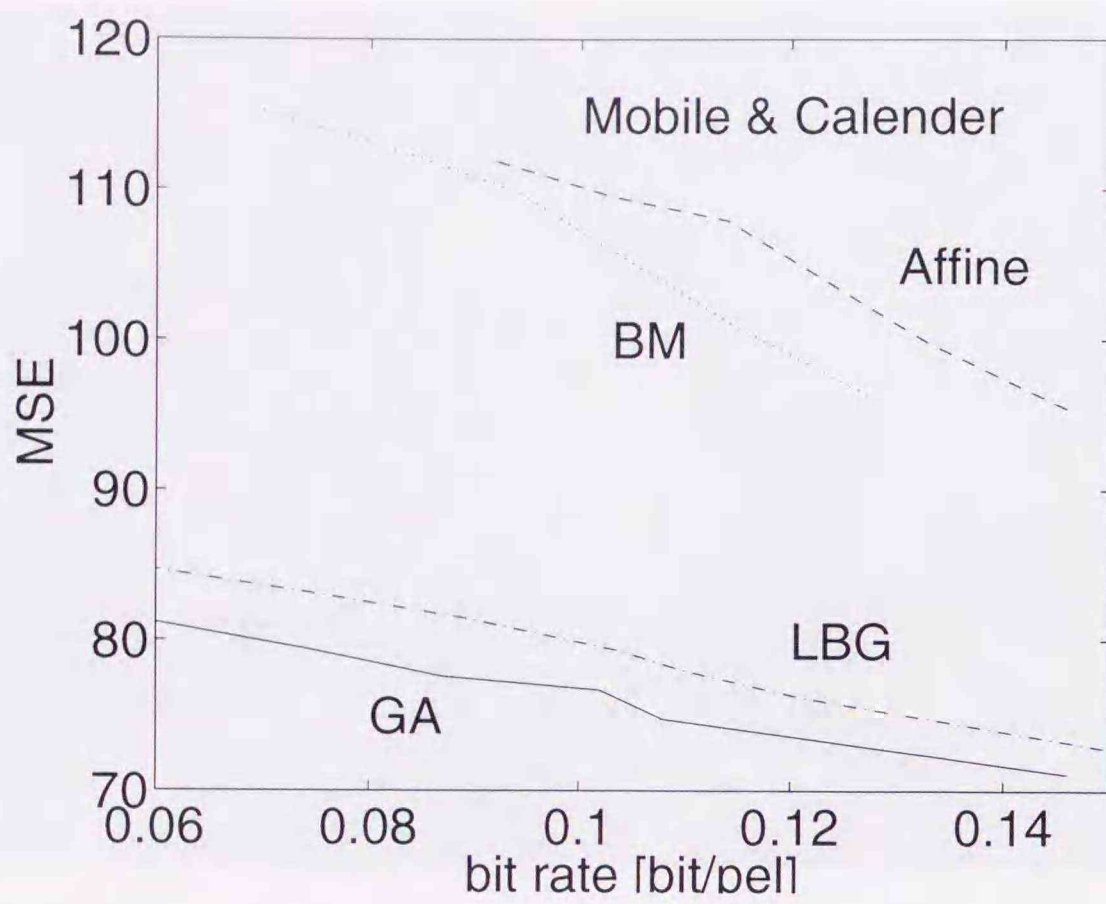


図 6.10: 符号量と MSE(Mobile & Calendar)

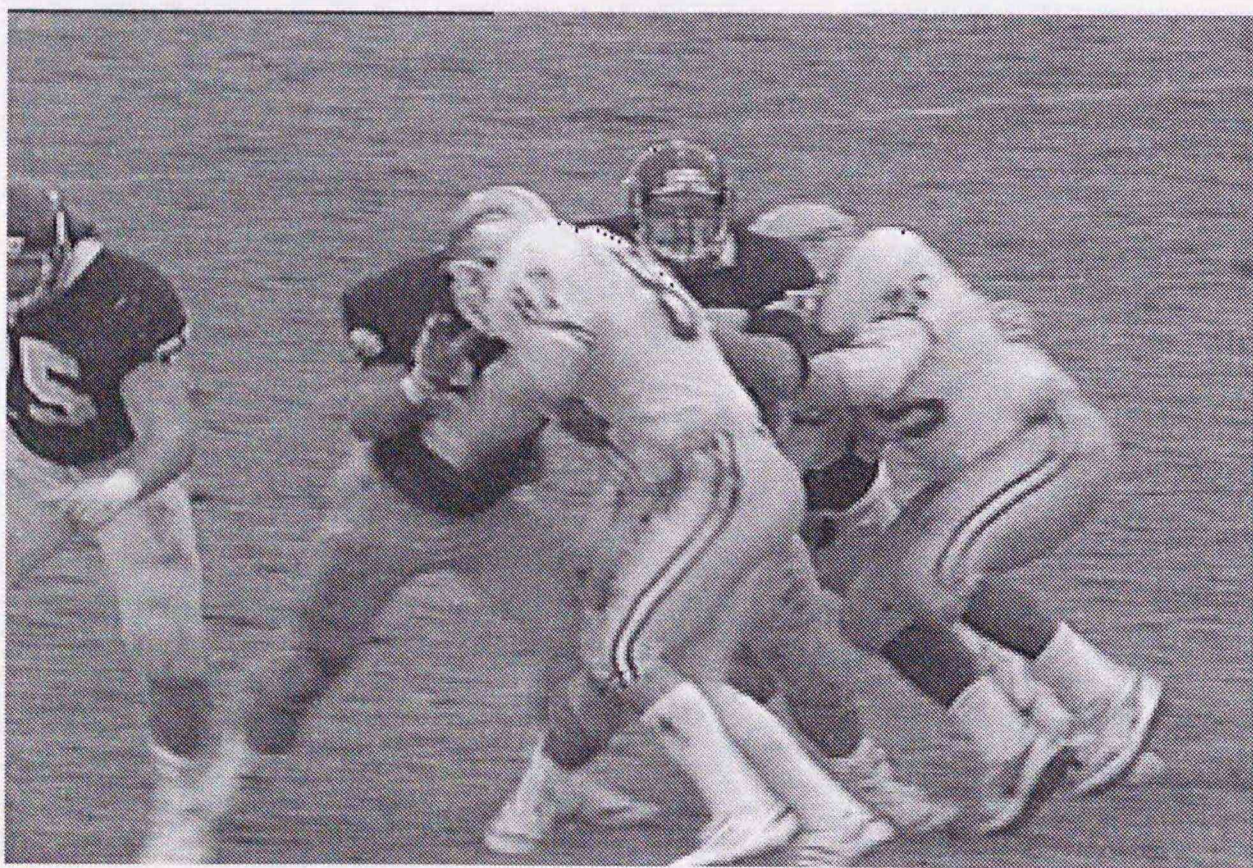


図 6.11: 原画像 (Football)



図 6.12: 動き補償予測画像 (b):(iii)LBG(bit rate=0.104[bit/pel],MSE=115.3)(Football)

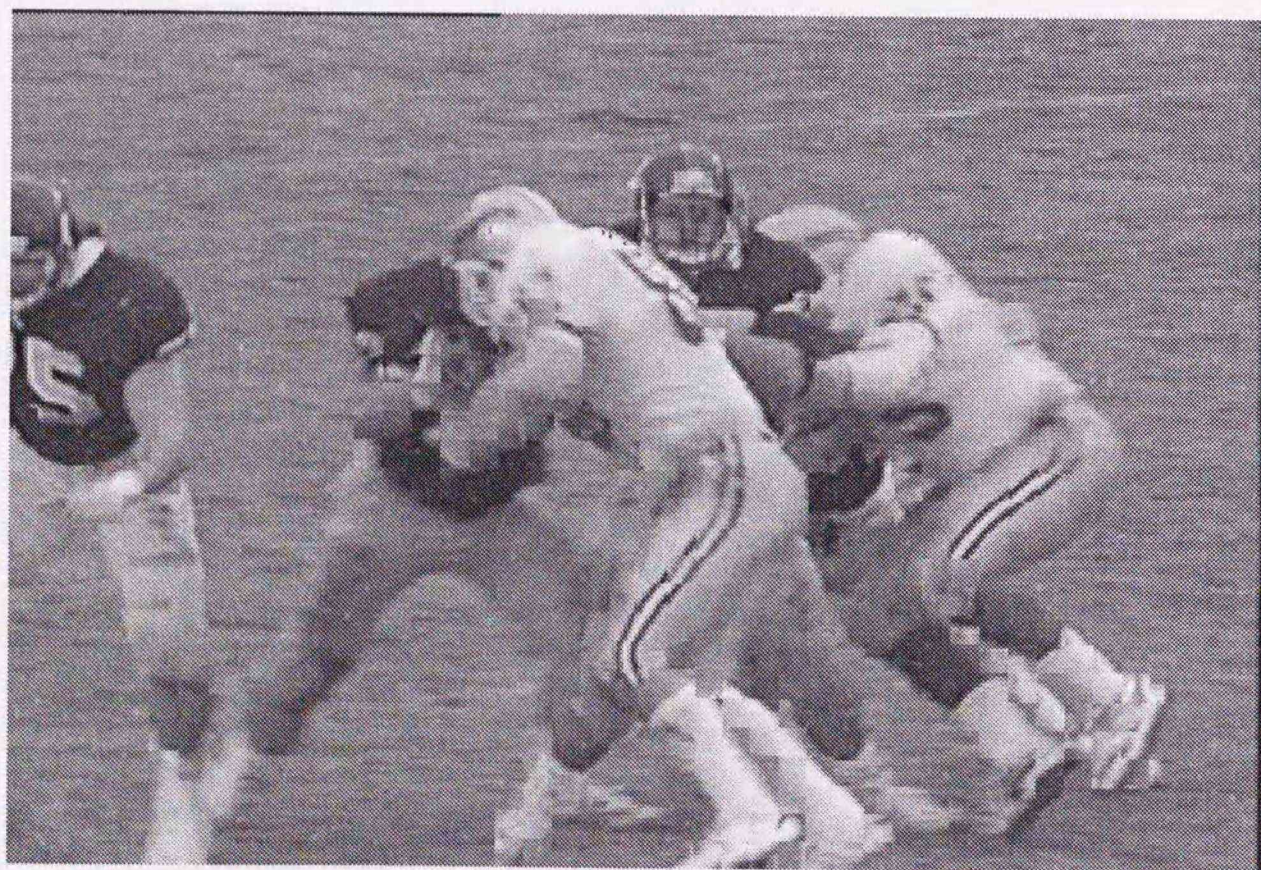


図 6.13: 動き補償予測画像 (c):(iv)GA (bit rate=0.104[bit/pel],MSE=102.6)(Football)

少し、予測画像全体の MSE も減少している。

平行移動を表現している  $(a_0, b_0)$  を、原点からのベクトル表示した結果を図 6.15 に示す。

図 6.15(a) は、画素ごとの動きを、ブロックごとにアフィン式を用いて近似 (6.3.3. 参照) したベクトルである。図 6.15(b) は、ブロックごとに求めた図 6.15(a) の動きパラメータを LBG アルゴリズムを用いて、ベクトル量子化したベクトル ((iii)LBG) である。図 6.15(b) は、図 6.15(a) のブロックごとのベクトルを量子化誤差が最小にベクトル量子化されていることがわかる。図 6.15(c) は、(iv)GA の動きテーブルのベクトルである。(iv)GA から得られたベクトルは、動き検出を行なうことなく、予測誤差が最小になるように求められたベクトルである。

動画像符号化における動きテーブル参照による動き補償方式では、画像内の動きを精度良く検出しベクトル量子化し、動きテーブルを作成することより、動き補償予測画像の予測誤差が小さくなるように、動きテーブルを作成を行なう方が画質の向上につながる事がわかった。

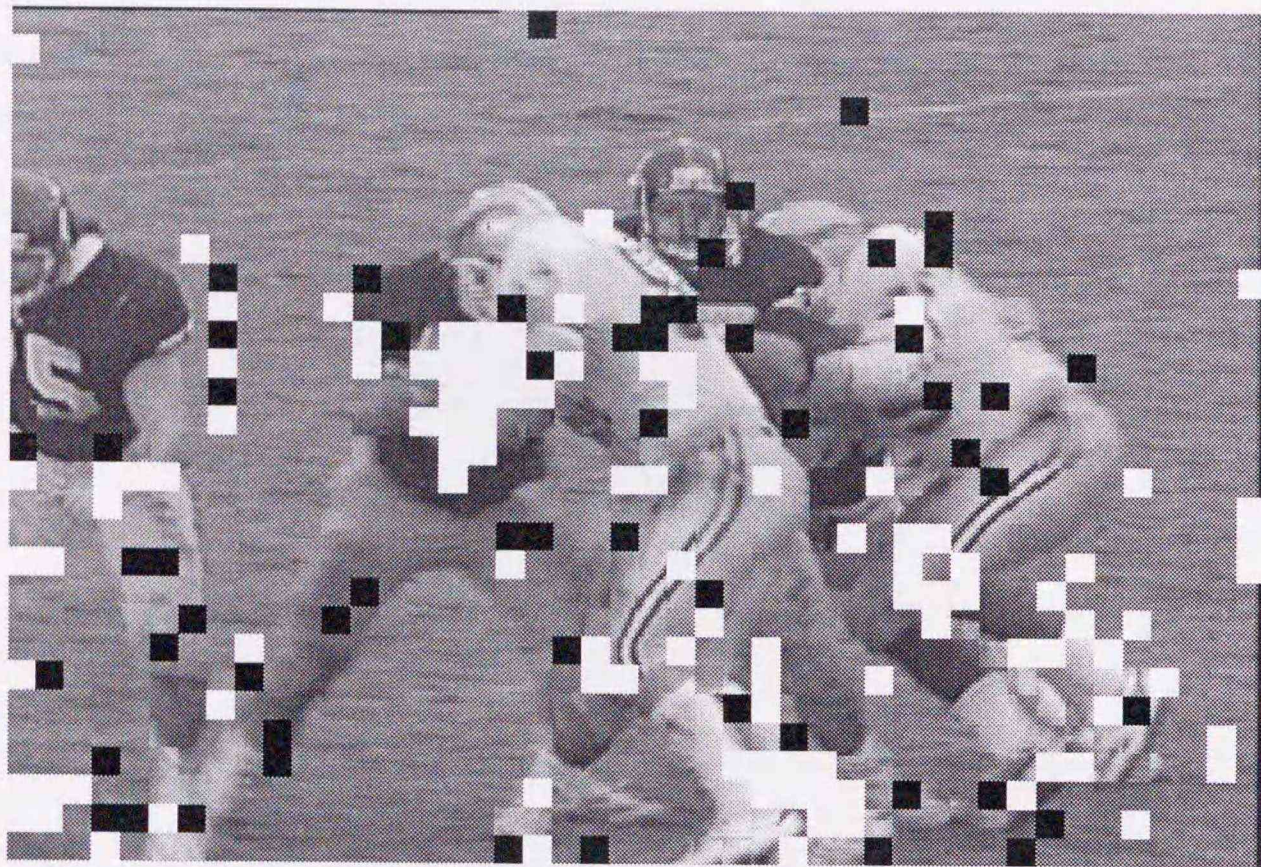


図 6.14: 動き補償予測画像 (d):(iii) LBG と (iv) GA の差分画像 (Football)

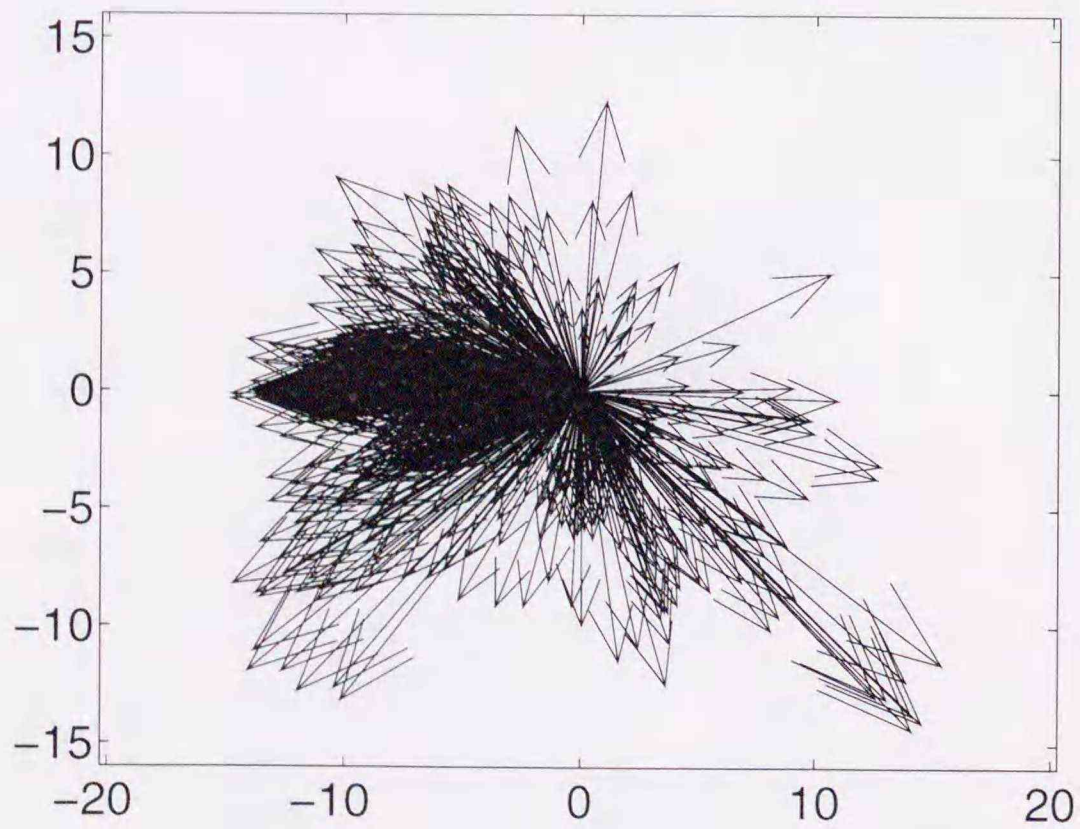


図 6.15: (a):画素ごとの動きを  $8 \times 8$  ブロックでアフィン式近似したベクトル  $a_0, b_0$  (1320本) (Football)



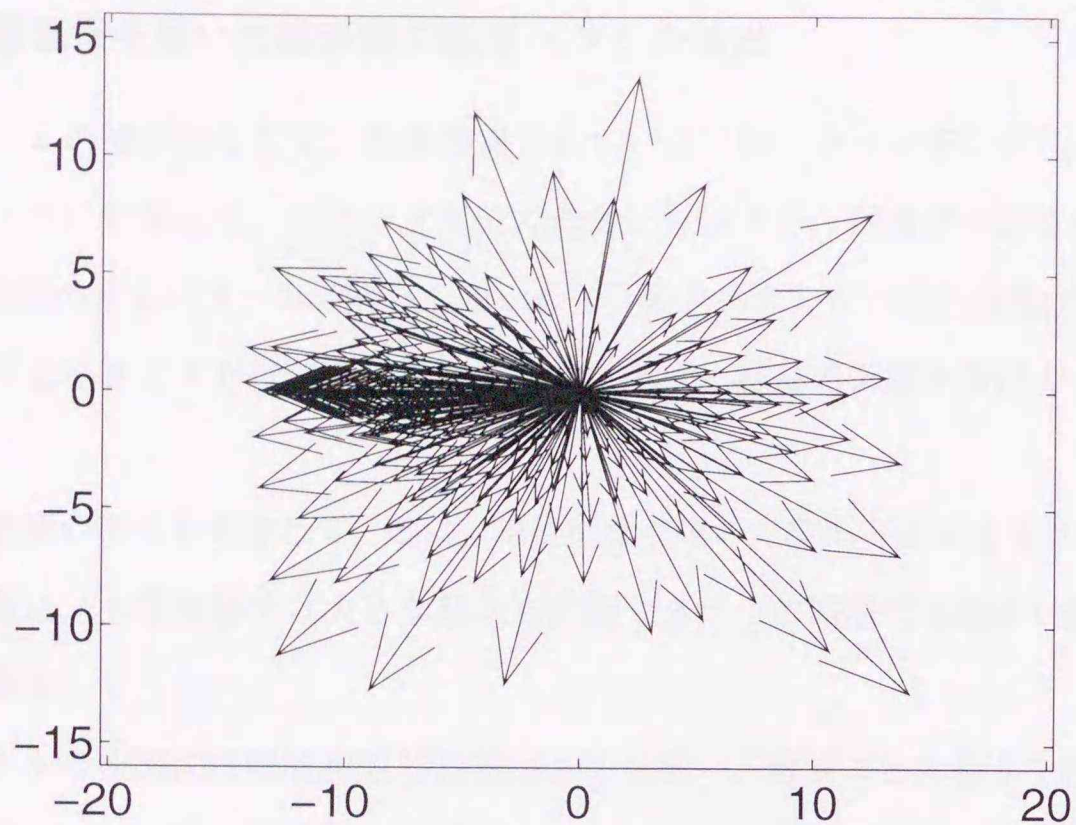


図 6.16: (b):(iii) LBG の動きテーブル内のベクトル  $a_0, b_0$  (256 本)

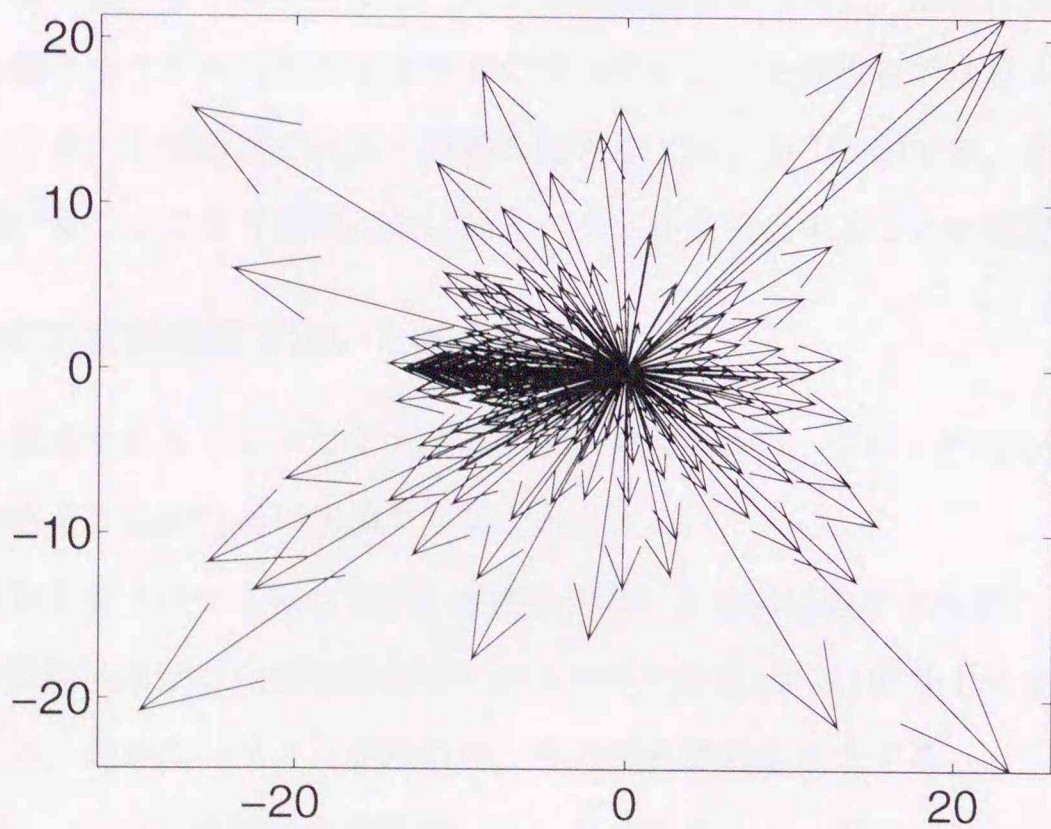


図 6.17: (c):(iv) GA の動きテーブル内のベクトル  $a_0, b_0$  (256 本)

## 6.6 DCT を用いた動きモデル

### 6.6.1 階層画像を用いた画素毎の動きベクトル検出

動きベクトルの検出法として、従来提案されているブロックマッチング法は画像をいくつかのブロックに分割して、ブロック単位で動きを検出する。輝度が平坦な領域では、ブロック内に特徴のあるパターンが存在しないので、動きベクトルの検出精度が低下する。ブロックサイズを小さくすればこの問題を回避できるが、雑音の影響を受けやすく検出誤りが発生する。

こうした画像の特性を考慮して、既に、階層画像を用いた検出方式が提案されている。この方式の特徴は、上位階層では大きな動きを把握でき、下位階層では細かい動きを捉えられることである。

動きベクトルの検出には階層画像 [37](図 6.18) を用い、画素ごとに動きを検出する [38]。階層を  $h$ 、現フレームでの時間を  $t$ 、階層  $h+1$  における画素位置を  $(x, y)$  とすると、以下のように表現することができる。

$$f_{t, h+1}(x, y) = \frac{1}{4} \sum_{i=0}^1 \sum_{j=0}^1 f_{t, h}(2x+i, 2y+j) \quad (6.23)$$

前フレームと現フレームに対してそれぞれ階層画像を作成し、最上位階層から最下位階層へ向けて動きベクトルの検出を逐次的に行っていく。上位階層では動きを大局的にとらえることができ、下位階層ではより詳細な動きをとらえることができ、画素間において線形内挿を用いることにより画素ごとに動きベクトルを検出することが可能になる。

### 6.6.2 DCT 変換係数を用いた動きモデル

動画像を構成する各フレームをブロックに分割した時の、ブロック内の画素の動きには、高い相関があることが容易に推測することができる。

画素毎の動きをブロック毎に DCT 変換を行ない、変換係数を求める。パラメータ数は、変換係数を低周波成分から高周波成分へとジグザグ走査 (図 6.19) をしながら、パラメータ数を決定する。定めたパラメータ以外は、その変換係数を 0 とする。

予測画像は、ブロック毎に変換係数に対し IDCT (inverse discrete cosine transform) を行ない、得られた動きベクトルから、動き補償を行なうことにより得ることができる。

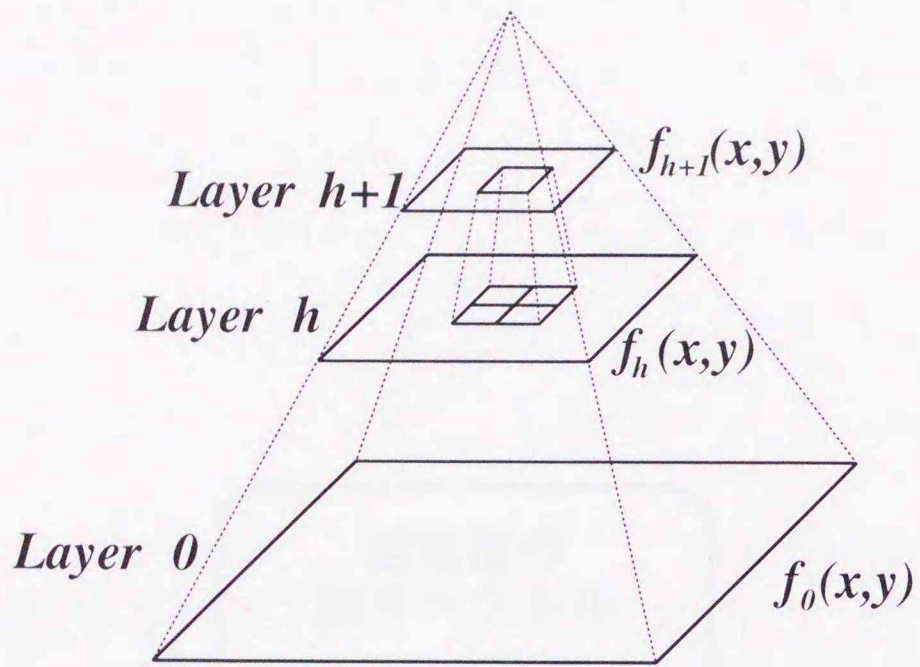


図 6.18: 4 分木階層画像の生成

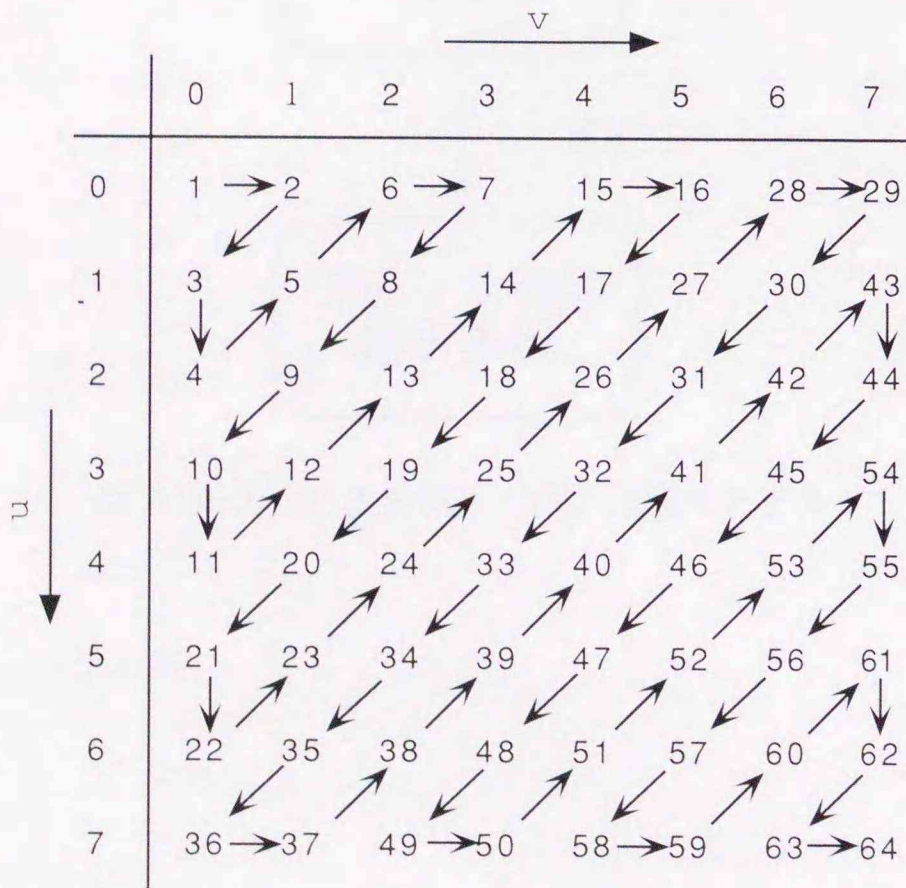


図 6.19: ジグザグ走査

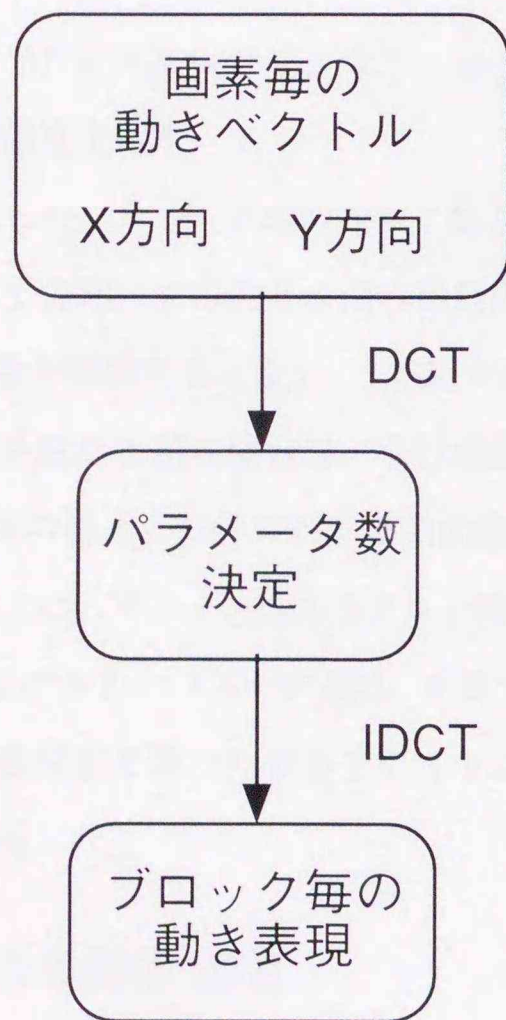


図 6.20: DCT 変換係数を用いた動きモデル

### 6.6.3 実験

各動きモデルを用いて、実際に動画像の動き補償予測実験を行なった。画像分割のブロックサイズは、 $8 \times 8$ の固定サイズで行なった。用いる画像は、図中に示す5種類(画像サイズは $352 \times 240\text{pels}$ )のY信号(8bit階調)である。各画像の0, 1フレームを用いた。予測画像の平均2乗誤差値(Mean Square Error : MSE)により評価する。

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (org_{ij} - MC_{ij})^2 \quad (6.24)$$

ここで、フレームサイズを $M \times N\text{pels}$ とし、また、 $org_{ij}$ ,  $MC_{ij}$ はそれぞれ原フレーム、動き補償予測画像の画素濃度を示す。

図6.21にDCT係数の使用したパラメータ数の変化と動き補償予測画像のMSEを示す。使用する変換係数の個数は、X方向、Y方向それぞれ同数用いた。

図6.21より、DCT変換係数が増加するに従い、MSEが減少していることがわかる。

また、使用するDCT変換係数が1個の場合は、平行移動モデルと等価で、使用する変換係数が64個の場合は画素毎の動きから得られる予測画像となる。

図中に示す、「o, x, +, \*, .」は、アフィン動きモデルを用いた動き補償予測画像のMSEを示している。アフィン動きモデルのパラメータ数は、3個である。同パラメータ数の、アフィン動きモデルとDCT変換係数を用いた動きモデルでは、ほぼ同様なMSEの予測画像が得られていることがわかる。

## 6.7 テーブル参照による動き補償

DCT変換係数を用いた動きモデルを、テーブル参照による動き補償[001]に適応する。以下にテーブル作成法を説明する。

### 6.7.1 画素毎の動きを用いたテーブル作成法

画素毎の動きベクトルの検出結果から以下の手順でテーブルを作成する。

手順1 :画素毎の動きベクトル検出。

手順2 :手順1で求めた画素毎の動きベクトルをブロック毎に各動きモデルを用いて動きパラメータ近似。

手順3 :手順2で求めた動きパラメータをベクトル量子化し、テーブルを作成。

### 6.7.1.1 ベクトル量子化

手順 2 で求めた動きパラメータを、選択したパラメータ数次元ベクトルとみなしベクトル量子化器を作成する。ベクトル量子化器の作成は、LBG アルゴリズム [39] により行なう。その際、動画像データ数フレーム分 (以下のシュミレーションでは各テスト画像の第 1 ~ 第 8 フレーム) から得た動きパラメータをトレーニング系列として用いる。ベクトル量子化によって得られた代表ベクトル (動きの代表組) をテーブルと呼び、テーブル参照による動き補償に利用する。

## 6.8 予測性能の比較

### 6.8.1 動きモデルの違いによる比較

異なる動きモデルを用いた、作成されたテーブル参照による動き補償により予測画像を得た。

以下に示す各方式を用いて実験を行ない、その結果を表 6.1 に示す。ただし、

(i)Affine : アフィン動きモデルによる、テーブル参照による動き補償

(ii)DCT : DCT 変換係数を用いた動きモデルによる、テーブル参照による動き補償

である。

### 6.8.2 実験

提案方式の有効性を確認するため、実際に動画像を用いて実験を行なった。(i) Affine, (ii) DCT 方式についてそれぞれ各隣接フレーム間において実験を行なった。画像分割のブ

表 6.1: 予測画像の MSE

Test Image	Football	Flower garden	Mobile Calendar
(i) Affine	118.112	100.317	76.104
(ii) DCT-1	120.695	102.987	80.066
(ii) DCT-3	122.252	100.816	76.462
(ii) DCT-6	125.760	100.582	75.739
(ii) DCT-10	126.717	100.930	74.626
(ii) DCT-16	130.386	100.260	75.915

ブロックサイズは、 $8 \times 8$ の固定サイズで行なう。用いる画像は、表6.1に示す3種類(画像サイズは $352 \times 240\text{pels}$ )のY信号(8bit階調)である。

各画像の0～1フレームを用い、各々の画像に対してテーブルを前述したように作成した。求めたテーブルより画像を再生する際に、現フレームを予測するには原画像を用いている。

テーブル内に格納されている動きの代表数は、256個とした。

表6.1に予測画像のMSEを示す。

(ii) DCT-xのxは、それぞれ使用したDCT変換係数の数を示している。

テーブル参照による動き補償におけるDCTを用いた動きモデルの場合、使用するDCT変換係数を増やしても、MSEの減少は見られなかった。これは、テーブル参照方式による動き補償では、テーブルを作成する際にベクトル量子化を行なうため、量子化誤差の影響が大きいと考えられる。

## 6.9 性質の異なる変換係数毎にテーブル作成

変換係数の使用するパラメータ数を増加させて、動き補償を行なった場合、その予測画像のMSEは減少する事を3.3節で示した。しかし、使用するパラメータ数を増加させて、テーブル参照による動き補償を行なってもMSEの減少は見られなかった。

これは、テーブル作成の際に、性質の異なる変換係数も、一緒にベクトル量子化を行ない、それをテーブルとしているからと考えられる。

そこで、図6.22に示すように性質の異なる変換係数を分けてベクトル量子化を行ない、それぞれのテーブルを作成した。

また比較として、アフィン動きモデルを用いたテーブルは以下のように作成した。

$$\begin{cases} u = \boxed{a_0} + \boxed{a_1x + a_2y} \\ v = \boxed{b_0} + \boxed{b_1x + b_2y} \end{cases} \quad (6.25)$$

$a_0, b_0$ をTable1,  $a_1, a_2, b_1, b_2$ をTable2として、ベクトル量子化を行ないテーブルを作成し、テーブル参照による動き補償を行ない、予測画像のMSEを表6.2に示す。

表6.2に予測画像のMSEを示す。なお、Affine-x-x, DCT-x-x-x-xのxはそのテーブル内の動きの代表数を示している。

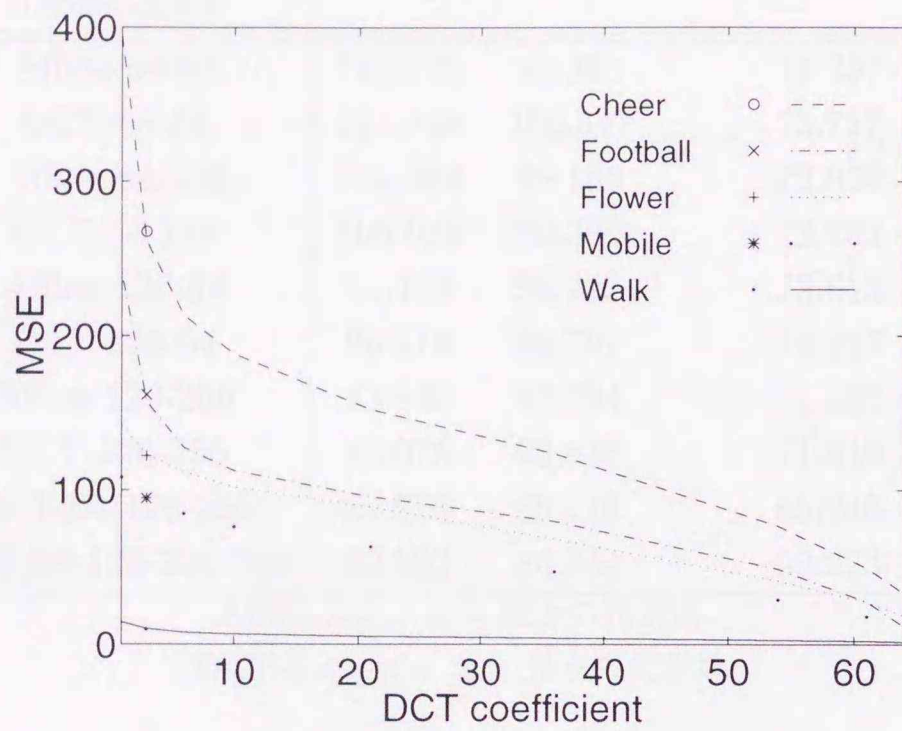


図 6.21: DCT 変換係数と予測画像 MSE

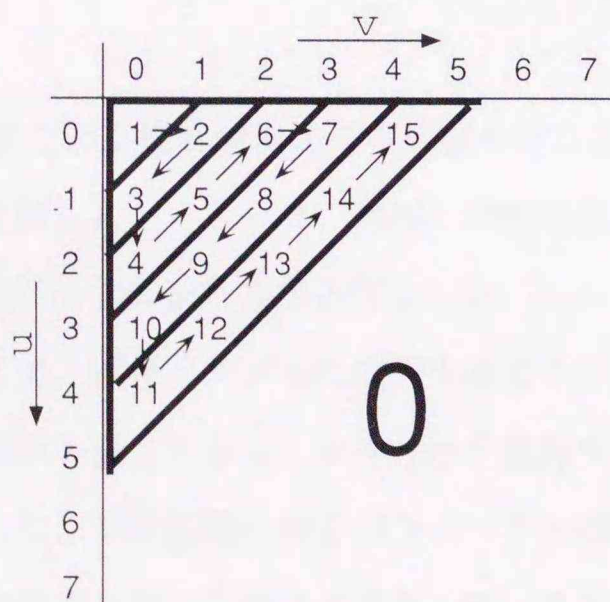


図 6.22: 性質の異なる変換係数



表 6.2: 予測画像の MSE

Test Image Table1-2-3-4	Football	Flower	Mobile & Calendar
Affine-64-64	112.272	99.881	74.397
DCT-64-64	111.819	100.347	73.717
Affine-64-128	106.784	98.186	72.856
DCT-64-128	105.099	98.309	72.763
Affine-128-64	94.139	96.493	73.613
DCT-128-64	98.618	96.797	73.227
Affine-128-256	83.840	92.794	71.525
DCT-128-256	89.078	93.836	71.849
DCT-64-128-256	84.976	89.718	65.246
DCT-64-128-256-256	80.992	84.044	59.223

Affine-x-x : x = 動きの代表数

DCT-x-x-x-x : x = 動きの代表数

DCT 動きモデルの性質の異なる変換係数毎にテーブルを作成し、パラメータを増加させた場合、変換係数を一つのテーブルとした場合と異なり MSE が減少した。

また、アフィン動きモデルと DCT を用いた動きモデルでは、同数の動きの代表数の場合、ほぼ同じ MSE を得た。しかし、DCT を用いた動きモデルは、さらにパラメータ数を増加させる事ができ、高精度な動き補償が可能である。

## 6.10 まとめ

本章では、テーブル参照による動き補償に平行移動モデル以外の動きモデルを導入し、その動き補償予測性能を比較した。一つ目は、回転・伸縮等を表現できるアフィン動きモデルを用いて、テーブル参照による動き補償を行なった。シミュレーションの結果より、6つのパラメータをそのまま、動きテーブルに登録するより、性質の異なる平行移動を表すパラメータと回転・伸縮等を表すパラメータを分けて登録することにより、同符号量での予測画像の画質が向上した。平行移動を表すパラメータと回転を表すパラメータはその取り得る範囲が違うことを確認した。このことより、テーブルに登録する場合は、それぞれ性質の異なる別々のテーブルに分け登録することにより、テーブル参照による動き補償において精度が向上することが分かった。

二つ目は、画素毎の動きより、ブロック毎に DCT 変換係数を用いた動きモデルによる

動き補償では，使用する変換係数を増加させることにより，MSE が減少することを確認した．DCT 変換係数を用いた動きモデルによるテーブル参照による動き補償を行なった場合は，そのテーブル作成の際の量子化誤差による影響により，使用する変換係数を増加させても，MSE の減少は見られなかった．アフィン動きモデルと同様に，性質の異なる変換係数を分けてベクトル量子化を行ない，テーブルを作成すると，使用するパラメータ数，動きの代表数の増加とともに，MSE の減少を確認した．

## 第 7 章 結論

本論文においては動画像符号化における動き補償の各種方式を詳細に検討し、新たな提案をし、これに要する符号量の大幅削減を実現する具体策を示した。これを通じて動画像における時間方向の冗長性を除去し、符号化効率の向上に貢献できた。

動き補償予測は動画像の高効率符号化のためには欠くことのできない技術であり、今後も更なる高効率化に向けて研究を続けていくべき事柄である。動き補償法について、いくつかの問題点を指摘した。そして、それらの問題点を解決するための動き補償法の符号化法を提案した。また、本論文の提案した方法について、実際の動画像データを用い検証した。そして、提案方式の優れた特性や残された課題について明らかにした。以下に各章で論じた事や得られた結果について箇条書にしてまとめる。

### 第 2 章の要約

1. 動画像の符号化について国際標準を中心に概説した。
2. ITU-T によりテレビ会議/電話等の通信用動画像符号化の国際標準として、H.261 が勧告され、蓄積用動画像符号化の国際標準として MPEG1 と MPEG2 が登場した。
3. 移動体通信のための超低ビットレート符号化方式とマルチメディアデータを自由に扱うための汎用符号化の国際標準として MPEG4 が登場した。
4. 動画像の伝送システムを広く普及させ、そのシステムを十分に活用するためには標準化が必要である。
5. 研究や開発を行なう際も、標準化を十分に踏まえて行なう必要がある。

### 第 3 章の要約

1. 動画像の符号化に関する各種技術の中で特に重要だと思われる動きベクトルの検出法について説明した。
2. 動きベクトルはフレーム間の相関を取り除く最も有効な手段である動き補償に用いられ、動画像の“動き”を表現する重要なベクトルである。
3. 検出に際しては、動きベクトルの検出精度と検出時間が目的によって考慮され、最も適した検出法が選択される。
4. 検出精度と検出時間は相反するものであり、その両方のバランスを取った検出法を目的に従って選択する必要がある。
5. ブロック・マッチング法ではブロック歪が顕著に現れ、これに対して対策を施す必要がある。

### 第 4 章の要約

1. 最適化・探索問題に適応可能な枠組である遺伝的アルゴリズムの概要について説明した。
2. 簡単な遺伝的アルゴリズムを用いた例を示しながら、具体的な処理手順を説明した。
3. 遺伝的アルゴリズムの各種の規則やパラメータの設定等について説明した。
4. 遺伝的アルゴリズムの基本定理であるスキーマ定理について説明した。この定理は、遺伝子型の中の注目している遺伝子の並びが、進化シミュレーションにおいてどのような生存の確率を持つのかを求める指針になることを説明した。

### 第 5 章の要約

1. ブロック毎の動きをテーブルに登録しておき、そのテーブル内から動き補償予測誤差が最小になる動きを選択して動き補償を行なうテーブル参照による動き補償を提案した。
2. 提案手法は、テーブル内に登録されている動きのみを使用するので、テーブルの作成法により、予測画像の画質に直接影響する。

3. 予備的に汎用動きテーブルを用いて動き補償を行ない、使用される動きベクトルの出現頻度の高い順に優先的に動きテーブルに登録する方式を提案した。
4. 探索範囲と精度を定め、符号化を行なう画像に対して動き補償予測誤差が最小になるように遺伝的アルゴリズムを用いたテーブル作成法を提案した。
5. 従来法であるブロックマッチングを用いた動き補償とテーブル参照による動き補償を、実際の動画像を用いてシミュレーションを行ない、その有効性を確認した。
6. 動きテーブル作成法の違いにより、予測画像の画質に影響があることを確認した。

## 第6章の要約

1. 通常の動き補償はブロック毎に行われるためにブロック歪を生じる。
2. ブロック歪はブロック境界部分での予測誤差を増加させ、高周波成分を増加させる。
3. ブロック歪を減少させるために、小領域アフィン変換を用いた動き補償法を提案し、実際のシミュレーションによりその有効性を確認した。
4. テーブル参照による動き補償方式と小領域でのアフィン変換を組み合わせて、より効率の良い動き補償を行なうことができた。
5. ブロック毎の動きをDCTを用いた動きモデルを提案し、使用するパラメータ数を変化させてシミュレーションを行なった。
6. 動き補償に平行移動以外の動きモデルを用いることにより、動き補償予測精度が向上するが、現時点ではいずれも符号量と画質の関係で従来のブロックマッチング法を用いた動き補償予測性能を著しく上回るものではないことを確認した。

近年、パーソナルコンピュータの高度化が進み、家庭にあるコンピュータで MPEG1 の符号化画像を容易に見ることができるようになっている。また、ハードウェアを用いて MPEG2 もリアルタイムエンコードができ、HD、DVD 等にデジタルで動画像を記録しておくことができるようになった。従来のアナログ電話回線や移動体通信に用いることを目的とした MPEG4 という国際標準が検討され始めている。このように、これからは動画像データの圧縮法がますます我々の生活において身近なものとなって行くであろう。MPEG4

のような超低レート (64 kb/s 程度) での動画データ圧縮は現存の方法よりももっと高度な技術を要求する。DCT 中心であった符号化法にもウェーブレット変換やフラクタル等の新しい手法が登場してきている。これらの手法が実用化されるにはまだかかりそうではあるが、MPEG4 以降の国際標準において間違いなく主役を担うことになろう。著者もこれらの新しき流れを十分に把握し、これからの時代の新しい動画データの圧縮法の開発に携わって行きたいと考える。

## 謝辞

本研究は、著者が北海道大学大学院工学研究科電子情報工学専攻に在学中に、1996年より3年間にわたって行なったものである。

本研究の遂行のみならず、あらゆる面において、終始御指導、御鞭撻を頂いた北海道大学大学院工学研究科、北島秀夫教授に深く感謝します。また、学術会議への参加など、多くの本研究に有益な活動の機会を与えて下さった事に対しても、深く感謝致します。

本研究の遂行にあたり、研究の方向を示し、終始御指導及び有益な御助言を頂いた北海道大学大学院工学研究科、長谷山美紀助教授に深く感謝致します。

本論文をまとめるにあたり、御助言を頂き、御指導を頂いた北海道大学大学院工学研究科、枅内香次教授に深く感謝致します。

本論文をまとめるにあたり、御助言を頂き、御指導を頂いた北海道大学大学院工学研究科、青木由直教授に深く感謝致します。

著者の研究期間中、御協力を頂いた北海道大学大学院工学研究科情報メディア工学講座像情報工学分野の学生の皆さんに感謝致します。

著者の長年の学生生活に、終始御支援、御協力を頂いた両親及び家族の皆さんに感謝致します。

# 文献

## 参考とした著書

- [1] 原島 博：“画像情報圧縮,” オーム社 (1991)
- [2] 安田 浩, 渡辺 裕：“デジタル画像圧縮の基礎,” 日経 BP 出版センター
- [3] 吹抜敬彦：“画像のデジタル信号処理,” 日刊工業新聞社 (1981).
- [4] 吉井貞熙, 田崎三朗, 小寺博, 渡辺裕：“ベクトル量子化と情報圧縮,” コロナ社 (1998)
- [5] 貴家仁志, 村松正吾：“マルチメディア技術の基礎 DCT 入門,” CQ 出版社
- [6] 谷萩隆嗣：“デジタル信号処理と基礎理論,” コロナ社
- [7] 谷萩隆嗣：“音声と画像のデジタル信号処理,” コロナ社
- [8] 小野文孝, 渡辺 裕：“国際標準画像符号化の基礎技術,” コロナ社
- [9] 映像情報メディア学会：“MPEG,” オーム社
- [10] K.R.Rao, J.J.Hwang：“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
- [11] 三木 弼一：“MPEG-4 のすべて,” 工業調査会
- [12] 北野 宏：“遺伝的アルゴリズム,” 産業図書
- [13] 北野 宏：“遺伝的アルゴリズム 2,” 産業図書



- [14] 北野 宏：“遺伝的アルゴリズム 3,” 産業図書
- [15] 電気学会 GA 等組み合わせ問題最適化手法応用調査専門委員会：“遺伝的アルゴリズムとニューラルネット (- スケジューリングと組合せ最適化,” コロナ社
- [16] 安居院 猛、長尾 智晴、“ジェネティックアルゴリズム”、昭晃堂 (1993)
- [17] Allen Gersho, Robert M. Gray：“Vector Quantization and Signal Compression,” Kluwer Academic Publishers

### 参考とした論文

- [18] 安田浩編著：“マルチメディア符号化の国際標準,” 丸善 (1991).
- [19] NHK 放送技術研究所編：“マルチメディア時代のデジタル放送技術事典,” 丸善 (1994).
- [20] テレビジョン学会編：“デジタル AV 規格ガイドブック,” オーム社 (1994).
- [21] 藤原洋監修：“ポイント図解式最新 MPEG 教科書,” アスキー出版局 (1994).
- [22] 渡辺裕：“MPEG2/H.262,” テレビ誌, vol.48, No.1, pp. 44-49 (Jan. 1994).
- [23] 安田浩編著：“MPEG/マルチメディア符号化の国際標準,” 丸善 (1994).
- [24] 原島博：“講座 画像情報圧縮の手ほどき (第 1 回)：画像情報圧縮の考え方,” テレビ誌, vol.43, No.6, pp. 603-612 (Jun. 1989).
- [25] 羽鳥好律, 橋本秀雄：“講座 画像情報圧縮の手ほどき (第 8 回)：動画像の符号化方式,” テレビ誌, vol.44, No.1, pp. 47-54 (Jan. 1990).
- [26] 羽鳥好律：“講座 画像情報圧縮の手ほどき (第 4 回)：画像符号化アルゴリズム (I) — 予測符号化 —,” テレビ誌, vol.43, No.9, pp. 949-956 (Sep. 1989).
- [27] 古閑敏夫, 平野昭, 大木淳一, 飯沼一元：“会議テレビ信号の動き補償フレーム間符号化,” 信学技報, vol.IE 81-54, (Jul.1981).
- [28] S. C. Kwatra, C. M. Lin, and W. A. Whyte：“An adaptive algorithm for motion compensated color image coding,” IEEE Trans. Commun., vol.COM-35, pp. 747-754 (Jul.1987).

- [29] J. O. Limb and J. A. Nurphy : "Measuring the speed of moving objects from television signals, " IEEE Trans. Commun., vol.COM-23, No.4, pp. 474-478 (Apr.1975).
- [30] C. Cafforio and F. Rocca : "Methods for measuring small displacements of television images, " IEEE Trans. Inf. Theory, vol.IT-22, No.5, pp. 537-579 (Sept.1976).
- [31] A. N. Netravali and J. D. Robbins : "Motion compensated television coding : Part I," Bell Syst. Tech. J., vol.58, pp. 631-670 (Mar.1979).
- [32] J. R. Jain and A. K. Jain : "Displacement measurement and its application in interframe image coding, " IEEE Trans. Commun., vol.COM-29, No.12, pp. 1799-1806 (Dec.1981).
- [33] R. Srinivasan and K. R. Rao : "Predictive coding based on efficient motion estimation, " IEEE Trans. Commun., vol.COM-33, No.8, pp. 888-896 (Aug.1985).
- [34] H. G. Musmann, P. Pirsch, and H. J. Grallert : "Advances in picture coding," Proc. IEEE, vol.73, pp. 523-548 (Apr.1985).
- [35] 滝保夫, 羽鳥光俊, 田中茂 : "動きに追従するフレーム間符号化," 昭 49 信学総全大, 1260, (1974).
- [36] 太田 睦 and 宮本 義弘 and 清水 孝 : "拡張されたグローバル動きベクトルの検出方式," 信学秋期全大,D-224,6,pp226,Oct.1990.
- [37] 富永 英義 and 小松 尚久 and 宮下 壮史 and 花村 剛 : "階層画像情報を用いた動画像における動き量検出方式," 電子通信学会論文誌 (D-II),J72-D-II,3,pp395-403,Mar.1989.
- [38] 金丸 利文 and 白川 智昭 and 北島 秀夫 and 小川 吉彦 : "画素毎の動きベクトル検出の一手法," 信学'92 春大,D-298,Mar.1992.
- [39] Y. Linde and A. Buzo and R.M. Gray : "An Algorithm for Vector Quantizer Design," IEEE Trans. Communications,COM-28,1,pp84-95,Jan.1980.

- [40] 如澤：“アフィン変換を用いた動き補償予測に関する検討,” 信学技報, **IE94-36**, pp. 1-8 (1994).
- [41] 横山, 宮本, 太田：“適応 warping 予測を用いた超低ビットレート動画像符号化,” 信学技報, **DSP96-5, IE96-5**, pp. 33-39 (1996).
- [42] J. Nieweglowski, T. G. Campbell and P. Haavisto: “A novel video coding scheme based on temporal prediction using digital image warping,” *IEEE Trans. on Consumer Electronics*, **39**, 3, pp. 141-150 (1993).
- [43] 上倉, 渡辺：“動画像符号化におけるグローバル動き補償法,” 電子情報通信学会論文誌, **J76-B-I**, 12, pp. 944-952 (1993).
- [44] 太田睦, 宮本義弘, 清水孝：“拡張されたグローバル動きベクトルの検出方式,” 1990 信学秋季全大, D-224, (Oct. 1990).
- [45] 宮本義弘, 太田睦, 清水孝：“ズーム・回転画像におけるグローバル動き補償,” 1990 信学秋季全大, D-225, (Oct. 1990).
- [46] 内藤, 川田, 松本：“各種動き補償方式の実画像による比較検証,” 映像情報メディア学会誌, **51**, 9, pp. 1560-1567 (1997).
- [47] D. E. Goldberg: “Genetic algorithm in search and optimization and machine learning,” Addison Wesley Publishing Company (1989).
- [48] “Iso/iec 11172 information technology: coding of moving pictures and associated audio for digital storage media at up to about 1.5mbit/s, part1: Systems; part2: Video; part 3: Audio; part4: Conformance testing” (1993).
- [49] 浅水 仁, 長谷山美紀, 北島秀夫：“動き補償のための動きテーブル作成法,” 電子情報通信学会技術研究報告 **IE98-24**, pp63-70 (1998-06).
- [50] 浅水 仁, 長谷山美紀, 北島秀夫：“テーブル参照による動き補償のための動きモデルの検討” 電子情報通信学会技術研究報告 **EID97-119, IE97-144**, pp7-12 (1998-02).
- [51] 浅水, 吉田, 長谷山, 北島：“動きテーブル参照による高精度動き補償方式”, 電気関係学会北海道支部連合大会, 284, pp. 323 (1997).

- [52] V. Seferidis: "Three dimensional block matching estimation", *Electronics Letters*, **28**, 18, pp. 1770-1772 (1992).
- [53] V. Seferidis and M. Ghanbari: "General approach to block matching motion compensation", *Optical Engineering*, **32**, 7, pp. 1464-1474 (1993).
- [54] Y. Nakaya and H. Harashima: "Motion compensation based on spatial transformation", *IEEE CSVT*, **4**, 3, pp. 339-356 (1994).
- [55] 木村 祥勝, 北島, 白川, 小川: "動画像符号化のための動きパラメータを用いた可変ブロックサイズ動き補償", *電子通信学会論文誌 (D-II)*, **J77-D-II**, 7, pp. 1273-1281 (1994).
- [56] H. Kitajima: "A symmetric cosine transform," *IEEE Trans. Comput.*, vol.C-29, pp. 317-323 (Apr. 1980).
- [57] Z. Wang: "Fast algorithms for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Acoust., Speech, and Signal Process.*, vol.ASSP-32, pp. 803-816 (Aug. 1984).
- [58] N. Ahmed, T. Natarajan, and K. R. Rao: "Discrete cosine transform," *IEEE Trans. Comput.*, vol.C-23, pp. 90-93 (Jan. 1974).
- [59] N. Ahmed and M. D. Flickner: "Some considerations of the discrete cosine transform," *16th Asilomar Conf. on Circuits, Systems and Computers*, Pacific Grove, CA, pp. 295-299 (Nov. 1982).

# 著者の研究業績

## 論文

### (A) 学会誌

- [001] 浅水 仁, 長谷山美紀, 北島秀夫: “動画像符号化のためのテーブル参照による動き補償” 電子情報通信学会 論文誌 (D-II), (採録決定)

### (B) 電子情報通信学会技術報告等

- [003] 浅水 仁, 長谷山美紀, 北島秀夫: “遺伝的アルゴリズムを用いた動画像符号化のための動きパラメータ決定” 電子情報通信学会技術研究報告 IE97-18 (1997-06)
- [004] 浅水 仁, 長谷山美紀, 北島秀夫: “動きテーブル参照による動き補償方式に関する考察” 電子情報通信学会技術研究報告 CS97-126, IE97-107 (1997-12)
- [005] 浅水 仁, 長谷山美紀, 北島秀夫: “テーブル参照による動き補償のための動きモデルの検討” 電子情報通信学会技術研究報告 EID97-119, IE97-144 (1998-02)
- [006] 浅水 仁, 長谷山美紀, 北島秀夫: “DCT を用いたテーブル参照による動き補償” 平成9年度 北海道大学知識メディアラボラトリー (VBL) 研究成果報告 (1998-03)
- [007] 浅水 仁, 長谷山美紀, 北島秀夫: “動き補償のための動きテーブル作成法” 電子情報通信学会技術研究報告 IE98-24 (1998-06)

## 講演発表

## 学会における研究発表

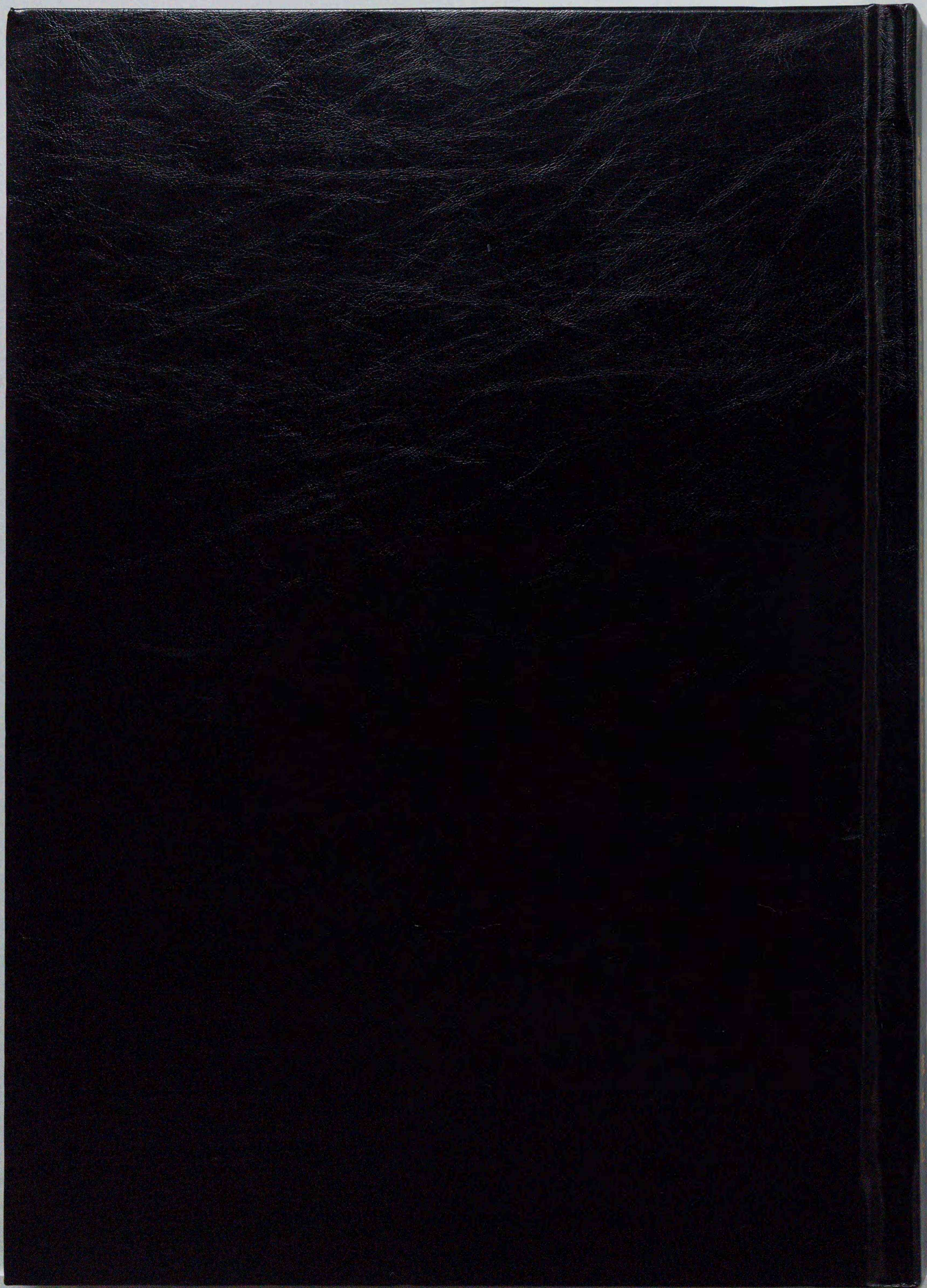
- [011] 浅水 仁, 向井田 健一, 佐藤 守行: “マグネシアに硫酸イオンを担持したときの固体酸塩基の挙動,” 北海道支部 1994 年夏季研究発表会, 日本化学会 (1994)
- [012] 浅水 仁, 向井田 健一, 佐々木 真: “複合酸化物上への硫酸イオン担持効果:  $SO_4^{2-}/SiOMgO$  の場合,” 北海道支部 1995 年冬季研究発表会, 日本化学会 (1995)
- [013] 浅水 仁, 向井田 健一, 佐々木 真: “ $SiO_2$  及び  $MgO$  の酸・塩基触媒特性に関する複合効果,” 北海道支部 1996 年冬季研究発表会, 日本化学会 (1996)
- [014] 浅水 仁, 秋保 政一, 長谷山 美紀, 北島 秀夫: “GA と SA を用いたフィルタ形状の変換について,” 電気関係学会北海道支部連合大会, p.353 (1996-334)
- [015] 浅水 仁, 吉田 仁, 長谷山 美紀, 北島 秀夫: “動きテーブル参照による高精度動き補償方式” 電気関係学会北海道支部連合大会, p.323 (1997-284)
- [016] 浅水 仁, 長谷山 美紀, 北島 秀夫: “動きテーブル参照による動き補償を用いた符号量削減” 電気関係学会北海道支部連合大会, p.17 (1998-17)

---

動画像符号化のための高能率  
フレーム間動き補償予測に関する研究

発行年月 1999年2月  
著者 浅水 仁 (Satoshi Asamizu)  
著者連絡先  
E-Mail [asamizu@media.eng.hokudai.ac.jp](mailto:asamizu@media.eng.hokudai.ac.jp)  
研究室 C350  
電話 011(706)6078 (1999年2月現在)

---





Inches 1 2 3 4 5 6 7 8  
cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

# Kodak Color Control Patches

© Kodak, 2007 TM: Kodak



# Kodak Gray Scale



© Kodak, 2007 TM: Kodak

A 1 2 3 4 5 6 M 8 9 10 11 12 13 14 15 B 17 18 19

