

平成 14 年度  
修士学位論文

# 準動画向け超軽量動画圧縮処理方式

A Super Light-Weight Compression Scheme for  
Semi-Motion Pictures

1055108 橋本 正和

指導教員 岩田 誠

2003 年 2 月 5 日

高知工科大学大学院 工学研究科 基盤工学専攻  
情報システム工学コース

# 要 旨

## 準動画向け超軽量動画圧縮処理方式

橋本 正和

昨今、高性能 LSI 技術やネットワーク技術の進展に伴い、映像メディアが様々な用途に活用され、MPEG や Motion-JPEG2000 などの標準方式も普及している。しかし、これらの標準方式は多様なアプリケーションを想定しているため、用途によっては過剰な仕様になり、高コストになる恐れがある。

そこで本研究では、遠隔監視や遠隔講義など、比較的動き領域が小さい動画、すなわち、準動画を対象とした簡易で高速な映像処理システムの構築を目的としている。準動画は人間の認知特性と酷似しており、これに認知特性を利用することにより、最適な映像圧縮が可能となる。本稿ではカメラアングル固定時に、指定した注視領域を基準として圧縮する手法により処理の軽量化を図った。さらに領域ごとの圧縮手法として、加減算のみで簡単に計算可能なピクセルベースの圧縮・伸張方式を採用し、より軽量化を図った。これを並列処理能力に優れたデータ駆動型プロセッサ [1] を用いて実装し、VGA サイズの画像を約 36dB の品質、39frame/sec. で圧縮可能であることを確認した。

キーワード 準動画, ピクセルベース, データ駆動

# Abstract

## A Super Light-Weight Compression Scheme for Semi-Motion Pictures

HASHIMOTO, Masakazu

With rapid progress of highly efficient LSI technology and network technology, image media are widely utilized for various applications. Furthermore, several video compression standards such as MPEG 2/4, Motion-JPEG2000 and so on have been proliferating. However, most of these standards may involve superfluous functions for a specific cheap application such as a simple remote surveillance system or a simple e-learning system because they are general-purpose compression schemes.

This thesis research aims at establishing a super light-weight compression scheme for semi-motion pictures in which there are a few motion objects. In the scheme, the close relationship between semi-motion pictures and human cognitive characteristics on vision are utilized. That is, motion areas in every picture frame is compressed in a high-quality mode and static background areas in periodical picture frame is done in a low-quality mode. Furthermore, in order to reduce computational complexity, a pixel-value-based compression scheme augmenting the block truncation coding is introduced as a picture compression scheme for each area in place of DCT or DWT. The scheme has also a novel method for detecting motion-areas and changing the area size in every picture frame. The scheme was implemented on a data-driven media processor and achieved 39 frames/ second for VGA pictures at about 36 dB picture quality.

*key words*    Semi-Motion Picture, Pixel Base, Data-Driven

# 目次

第 1 章	序論	1
第 2 章	準動画の特徴と既存の映像処理の概要	6
2.1	緒言	6
2.2	準動画の特徴	7
2.3	人間の映像認知特性	8
2.4	既存の領域抽出手法	9
2.4.1	ブロックマッチング法	9
2.4.2	領域分割法	10
2.4.3	問題点	11
2.5	既存の画像圧縮手法	11
2.5.1	フレーム間圧縮による映像符号化	12
2.5.2	フレーム圧縮による映像符号化	14
2.5.3	問題点	15
2.6	結言	15
第 3 章	準動画向け超軽量映像処理	17
3.1	緒言	17
3.2	超軽量映像処理	17
3.2.1	領域抽出手法	18
3.2.2	ピクセルベース圧縮伸張	19
3.3	注視領域圧縮処理	21
3.4	周辺領域圧縮処理	22
3.5	領域別アルゴリズムの性能評価	24

3.6	領域指定による問題点 . . . . .	25
3.7	通常時の映像圧縮処理 . . . . .	26
3.8	動き領域の適応的伸縮 . . . . .	27
3.9	結言 . . . . .	28
<b>第 4 章</b>	<b>性能評価</b>	<b>30</b>
4.1	緒言 . . . . .	30
4.2	シミュレータの原理 . . . . .	30
4.3	シミュレーション手法 . . . . .	33
4.4	実装手法 . . . . .	35
4.5	測定手法 . . . . .	36
4.6	性能評価 . . . . .	38
4.7	結言 . . . . .	41
<b>第 5 章</b>	<b>結論</b>	<b>42</b>
	<b>謝辞</b>	<b>46</b>
	<b>参考文献</b>	<b>47</b>
<b>付録 A</b>	<b>画質評価手法</b>	<b>48</b>
A.1	映像の客観評価 . . . . .	48
A.1.1	PSNR . . . . .	48
A.1.2	RMS . . . . .	50
A.1.3	コントラスト誤差 . . . . .	51

# 目次

2.1	準動画	7
2.2	認知特性	8
2.3	基本映像圧縮伸張処理	12
3.1	BTC の基本アルゴリズム	20
3.2	基本 BTC 処理結果画像	20
3.3	注視領域における符号・復号化アルゴリズム	22
3.4	周辺領域における符号・復号化アルゴリズム	23
3.5	注視領域用処理結果画像	24
3.6	周辺領域用処理結果画像	25
3.7	通常時の映像圧縮	26
3.8	動き領域の適応的伸縮	28
4.1	DDMP のナノプロセッサ構成	31
4.2	DDMP の並列概念図	32
4.3	DDMP10A の内部構成	33
4.4	注視領域用フローグラフ概念図	36
4.5	周辺領域用フローグラフ概念図	37
4.6	スループット算出手法イメージ図	38
4.7	超軽量映像圧縮方式の性能	40
A.1	映像の構成	50

# 表目次

2.1	MPEG1, 2, 4の比較 . . . . .	13
3.1	量子化/逆量子化 . . . . .	21
3.2	ピクセルベース圧縮手法の特性 . . . . .	24
4.1	動き領域の割合に対するスループット性能 . . . . .	39
4.2	動き領域の割合に対する圧縮率性能 . . . . .	39

# 第 1 章

## 序論

昨今，高性能 LSI 技術やネットワーク技術の進展に伴い，映像メディアが様々な用途に活用され，MPEG や Motion-JPEG2000 などの標準方式も普及している [2] ．

MPEG とは，ISO/IEC の動画像音声符号化グループ (Moving Picture coding Expert Group) のことであり，このグループにより標準化された動画符号化方式の名称にもなっており，現在までに MPEG1, MPEG2, MPEG4 が標準化されている ．

Motion-JPEG2000 とは，静止画符号化方式の JPEG(Joint Photographic Experts Group) を改良し，2001 年に標準化された静止画圧縮技術である JPEG2000 を，連続再生することにより動画として表示する技術である [3] ．

MPEG4 を例にとると，その想定用途は多種多様で，用途に応じたプロファイルとレベルが定義されている [4] ．ビジュアル・プロファイルにおいては，9 つのプロファイルとそれに応じたレベルが階層構造で定義されており，さらにオーディオ・プロファイルには 4 つのプロファイルとレベルが定義されている ．このように，これらの標準方式は多様なアプリケーションを想定しているため，用途によっては過剰な仕様となる可能性がある ．また，それを実現するために高いハードウェアパフォーマンスが要求され，LSI 技術が発展したとはいえ，高コストになる恐れがある ．ネットワーク利用を想定した場合でも，過剰な仕様であれば，余分な付加情報までも必要となり，無駄にネットワーク帯域を利用しなければならない ．無駄なく用途に応じた映像圧縮処理の実現も，映像圧縮処理技術の重要な課題である ．

既存の圧縮アルゴリズムとして，フレーム毎圧縮に DCT(離散コサイン変換) や DWT(離散ウェーブレット変換)，量子化，ランレングス符号，ハフマン符号，算術符号，フレーム間圧縮に動き推定，動き補償が用いられる ．さらに MPEG4 では圧縮率を向上させる手法と



して、オブジェクト符号化が利用されている。既存の標準方式ではこれらの手法を組み合わせることにより、汎用性に富んだ画質・圧縮率・速度性能を保証している。しかし、これらの標準方式は、浮動小数点演算など、高負荷の計算が必要であったり、多くのメモリ領域が必要であったりする。さらにフレーム間圧縮のために画像解析が必要となるため計算量は膨大となり、圧縮システムの軽量化には適していない。

また、用途に応じた映像圧縮処理といっても、その用途は MPEG4 の例にあるように多種多様である。そこで、前述した各種圧縮アルゴリズムを縮小可能であるような映像として、準動画を選択し、最適な映像処理システムの構築を目標とする。

準動画とは、フレームレートが少ない、または、通常は静止状態で、ある時刻だけ動画として見えるような映像を指し、例としては遠隔監視や、遠隔講義・会議などがあげられる。また、これらの映像は動きのある領域が局所的であり、通常は特定の領域内のみで動きが生じるという特徴がある。

準動画を対象として圧縮処理を検討する上で、日常ごく普通に行っている人間の生活に視点を置き、人間が行う情報処理やその特性から最適な映像圧縮伸張処理を考案する。

人間のコミュニケーションは、聴覚、視覚、触覚など、通常、五感のほとんど総てを使って表現し、理解している。その中で映像に関係のある視野に注目すると、視野は「見る」という行動から二種類の状態で構成されているといえる。一つは意識的に「見ている」という状態であり、人間は被写体の形状・状態など多くの情報を元に認識する。二つ目は無意識に「見えている」という状態で、意識していないため、少ない情報だけで認識する。つまり、「見ている」領域（注視領域）の情報伝達が保証されれば、「見えている」領域（周辺領域）の情報量は少なくとも認識可能であるといえる。これは、人間には周辺視野の認識率が低いという認知特性のためである。映像を見る人間の視覚認識がそのような特徴を持つのであれば、映像も視覚と同じように注目する領域と、周辺領域とで情報量を変化させることにより、的確な映像処理が実現可能となる。

そこで人間の認知特性を活かした映像圧縮処理を検討する。映像を注目領域と、それ以外の周辺領域に分離するために、以下の二つの重要な問題がある。

1. 領域を抽出するための最適な手法を考慮しなければならない
2. 抽出した領域ごとに最適な圧縮処理を考慮しなければならない

1. に関しては，既存の領域抽出手法としてブロックマッチング法や，領域分割手法がある．ブロックマッチング法は，平行移動で画像内の動きを近似可能であるという見地から誕生したアルゴリズムである．画像を適当なブロックに分割し，ブロック単位で平行移動量分を求めるというアルゴリズムで，MPEG の動き検出などで用いられる．領域分割法はリモートセンシング画像の解析過程などで行われ，古くからその方式として統合法，分離・統合法，弛緩法など数多く提案されている．しかし，これらを用いた領域抽出法はいずれも計算量が膨大で，準動画向けの簡易圧縮・伸張には適していない．そこで本研究では，カメラアングル固定時に利用者が一度だけ，動きがあると推測される任意領域を指定する手法を取り，領域抽出時にかかる演算負荷を軽量化することで対応する．

2. に関しても，様々な映像圧縮処理が考案されており，先ほど述べた標準方式も利用可能な一手段である．本稿では圧縮時の演算にかかる負荷の軽量化を考慮し，映像圧縮・伸張処理を行うための最適な方式として，ピクセルベース圧縮手法を採用した．ピクセルベース圧縮方式は，1979 年に発表された BTC(Brock Truncation Coding)[6] を基本方式とする．BTC は DCT や wavelet 等の周波数変換のような高負荷な演算を使用せず，加減算とシフト演算のみで簡単に計算可能で，かつ，ピクセル単位であるため，並列処理が容易な圧縮・伸張処理方式である．さらに BTC は静止画圧縮用に提案された手法であり，動画を対象とした場合に，フレーム間情報を独立に扱えるという利点がある．この方式を採用することにより，圧縮時における演算負荷の軽量化を図った．

以上のことを考慮し，本研究では，準動画を圧縮する上で最適な映像処理アルゴリズムとして，以下の手法を考案した．

- カメラアングル固定時に，動きがあると推測される領域（注視領域）を利用者に指定してもらおう
- 注視領域符号時に情報量を最小のまま，復元時に表現可能な色数を拡大するために，画

## 質を重視した圧縮伸張処理方式の提案

- 周辺領域符号時の情報量を少なくするために，圧縮率を重視した圧縮伸張処理方式の提案

上記手法では，通常時には利用者が指定した領域のみの圧縮処理となり，指定した領域から物体がはみ出した場合に映像が切れた状態となってしまう．本研究では，このような映像の不具合に対しても，適応的に動き領域を伸張して映像圧縮する方式を提案する．具体的には，画面を  $N \times N$  のタイルに分割し，動き領域の境界タイルと画像全体の境界タイルで，毎フレーム単位に輝度値の変化により動きの有無を検査し，変化があれば圧縮対象タイルとする．これにより，動き領域の形状を更新しながら適応的な圧縮が可能となる．

本手法は，任意領域の位置情報における計算が最低一回は必要となる．また，動き領域の検出のために，毎フレーム単位で境界タイルを検査する必要がある．これら位置情報の演算処理によるオーバーヘッドをできるだけ最小にするために，利用ハードウェアを DDMP(Data Driven Multi-Media Processor) に想定した．DDMP は画素の位置情報を世代番号として識別しており，多数の連想処理が，並列に実現可能であるという特徴を有する．また，組み込み型・極省電力・超高速といった利点もあり，既存の標準映像圧縮方式を専用ハードウェアで実装する場合に比べて，本システムに適用し易いといえる．

これらの提案方式を，10 個のプロセッサを搭載した DDMP のシミュレータ (シャープ製データ駆動型プロセッサシミュレータ) 上で実装し，シミュレーションで，画面全体に占める動き領域の割合におけるスループット性能，並びに圧縮率を算出した．また，本方式と同じく，動き推定などフレーム間情報の圧縮を行わない，既存の標準方式である Motion-JPEG2000 との比較を行った．

第 2 章では，準動画の特徴と人間の認知特性の共通点をあげ，既存の領域抽出手法，標準映像圧縮処理手法のしくみを述べる．また，標準映像処理の汎用性に富むことから起因する，演算時の高負荷における問題を述べる．

この問題を受けて第 3 章では，準動画向けの超軽量映像処理全体の概要を述べ，領域抽出

時の演算をできるだけ軽量化する領域抽出手法を提案し，領域抽出後の領域別の圧縮処理手法として，単純な演算で実現可能なピクセル・ベース圧縮手法の提案をする．次いで，提案した手法を準動画に適用する場合のアルゴリズムを提案し，領域抽出から起因する，指定した領域からのみ出しに対応するために，指定領域を適応的に伸縮して，画像を補間する手法を提案する．

第4章で，シミュレータの原理とシミュレーション手法，スループット性能，圧縮率に関する評価結果を述べ，既存の Motion-JPEG2000 との比較を行い，第5章で本研究の成果，今後の課題を考察する．

## 第 2 章

# 準動画の特徴と既存の映像処理の概要

### 2.1 緒言

映像メディアが様々な用途に活用されるようになり、圧縮伸張処理の重要性が高まっている。多様なアプリケーションにおける数々の要求に応えるべく、映像・音声における標準化が盛んとなり、MPEG や Motion-JPEG が誕生した。

しかし、総ての用途において標準技術が有効とはいえず、例えば動きのある時間・領域が少ない監視や講義・会議などにおいても、動き補償などの過剰な処理を行わなければならない。このような映像は準動画とよばれており、動きのある領域は映像全体に対して局所的で、フレームレートも要求されないという特徴を有する。

一方、人間の視野の特性を考慮した場合、視野は注視領域と周辺領域に分離できる。注視領域は情報量が多く、周辺領域は情報量が低い状態で、物体または背景を認識している。つまり、注視領域の情報が保証されれば、それ以外の周辺領域の情報量は少なくともよい。これは人間には周辺視野の認識率が低いという認知特性があるためである。この特性は準動画の特性と一致しており、準動画の圧縮・伸張に認知特性を活かせば、周辺領域(動きの少ない領域)に対しては計算量を抑えることができ、高速な映像処理が可能となる。

これを実現するために、注視領域と周辺領域を抽出するための技術が必要であり、また、抽出した領域に対して行う適切な映像処理を選択しなければならない。既存の領域抽出手法としては、ブロックマッチング法や、領域分割法があり、抽出領域に対して適用する代表的

な映像処理として、MPEG など代表される標準技術がある。

本章では、準動画を対象として映像を圧縮・伸張するシステムを実現するために、人間の認知特性と準動画の類似点を述べ、既存の領域抽出法、並びに既存のフレーム圧縮処理の概要と問題点を述べる。

## 2.2 準動画の特徴

準動画とは、通常時は静止画と区別がつかない映像で、ある瞬間だけ動画のように見える映像のことである。図 2.1 の例は監視状況を図示している。誰もいない個所を監視した状態ではほぼ静止画状態であり、誰かが通過するなどすると動画のように見える。正確に定義されていないが、このような映像を準動画または準静止画と呼ぶ。また、これらの映像はフレームレートをほとんど要求しないため、フレームレートが 30fps にも満たない映像に対して準動画と呼ぶ場合もある。

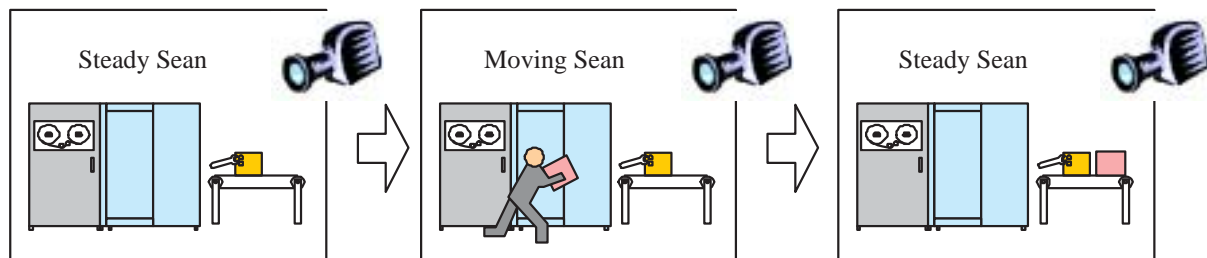


図 2.1 準動画

準動画はほとんど変化が無いという映像の性質上、ある一場面においてほとんどの領域が背景であり、時間的に見ても背景はほとんど変化しない。この観点からすれば、遠隔講義の映像も準動画といえる。講義風景において、講師や白板の文字などは絶えず動画として動くが、それ以外は背景としてほぼ静止状態である。

準動画に対して映像圧縮をする場合、動画として見える部分と、静止画として見える部分とで分離して圧縮するほうが、見たい部分の情報を保証したままで、高圧縮が実現可能となる。人間は普段の生活の中で、映像内で部分ごとに無意識に情報量を変化させて認識してい

る．次節では人間の認知特性の仕組みと準動画との共通点を述べる．

## 2.3 人間の映像認知特性

人間は雑音（ノイズ）と共に生活を営んでおり，ノイズから得られる情報を無意識に吸収している．ノイズは人間の五感総てに影響を及ぼし，時折有益な情報となりうるが，通常ほとんどの情報は意味をなさない．人間は，ノイズを含む情報の中で，吸収したい情報のみを集中して取り入れることにより，認識を実現している．

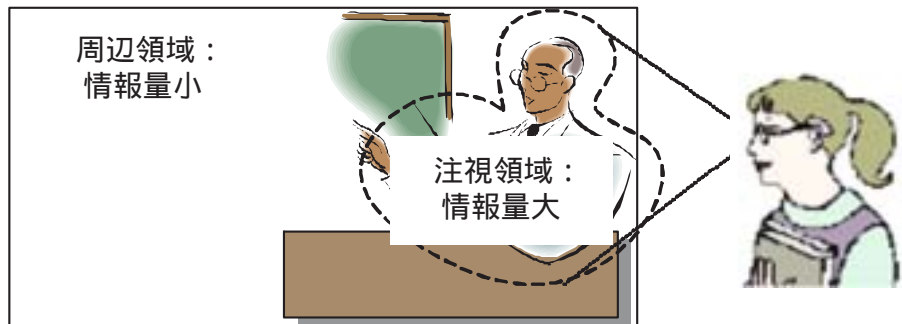


図 2.2 認知特性

人間のコミュニケーションは，聴覚，視覚，触覚など，通常，五感のほとんど総てを使って表現し，理解している．その中で映像に関係のある視野に注目すると，視野は，図 2.2 に示すように，「見る」という行動から二種類の状態で構成されている．一つは意識的に「見ている」という状態であり，人間は被写体の形状・状態など多くの情報を元に認識する．二つ目は無意識に「見えている」という状態で，意識していないため，少ない情報だけで認識する．つまり，「見ている」領域（注視領域）の情報伝達が保証されれば，「見えている」領域（周辺領域）の情報量は少なくとも認識可能であるといえる．これは人間には周辺視野の認識率が低いという，映像の認知特性のためである．つまり，注視領域の情報伝達が保証されれば，それ以外の周辺領域の情報量は最小限でよいといえる．

また，ある一点を見ている状態で，周りで何かが動くと，動いたと思われる個所に視点を移すという特性もある．つまり，視野における変化（動き）に対しては敏感であるという特性がある [5]．

動画像における視野は画像全体である。動画像が静止状態のとき、意識的に一個所を注視したり、画面全体を漠然と眺めたりする。この静止状態からある一個所に変化がおきた状態に遷移すると、人間は変化のあったものに注目する。アニメーションや、グラフィックスなどの映像によっては画像が不連続に変化することもありうるが、自然画像を記録したような映像においては、画面上で連続的に変化が起こり、その変化に対して人間は注目し、物体を認識している。また、変化のある一個所が映像の一部分でのみ発生するような映像の場合、ほぼ一定して映像の一部しか見ないことになる。例えば遠隔講義の場合、カメラが固定されると講師と白板はほぼ一箇所のみで変化する。

講師や白板を注視領域、それ以外を周辺領域とすると、準動画は人間の認知特性と同じ性質を持っているといえる。認知特性を動きのある領域が小さい準動画の圧縮に活用すれば、周辺領域、つまり動きの少ない領域に対しては計算量を抑えた圧縮・伸張ができ、高速な映像処理が可能となる。

## 2.4 既存の領域抽出手法

領域の抽出手法は様々な手法が提案され利用されている。例としては、MPEG などによく用いられるブロックマッチング法や、画像におけるオブジェクト抽出で利用される領域分割法がある。これらの特徴を次節より述べる。

### 2.4.1 ブロックマッチング法

ブロックマッチングは、基本的にパターンマッチングであり、MPEG において動きを検出するために利用され、動き推定などで用いられる。画像内の動きは、ほぼ並行移動で近似可能であると仮定し、画像を適当なブロックに分割し、ブロック単位で並行移動分を求める、というアルゴリズムである。具体的な手順を以下に述べる。

1. 対象フレームを適当なブロックに分割する。MPEG では、縦×横 =  $16 \times 16$  のブロックが用いられる。



2. 一つのブロックに着目し、これを前フレーム上でスライドさせ、最も適合する部分を探し出す。ブロック間の差分評価関数<sup>\*1</sup>を定義し、この距離が最小になるブロックを見つける。
3. 前フレーム上で対象ブロックのシフト量をベクトルとして求める。これを「動きベクトル(MV)」と呼ぶ。
4. 対象フレーム上のすべてのブロックに対するMVを求める。

## 2.4.2 領域分割法

画像に対して、それぞれの持つ特徴が一致する領域に分割する手法を領域分割とよぶ。この領域分割は、様々な手法が提案されており、小さな領域から一様な領域を統合する統合法、弛緩法を用いる手法などがある [7]。

統合法には、単純統合法、反復型領域拡張法などがあり、単純統合法はラスタ走査である一点を探し、その点から閾値以下の隣り合う画素を統合してラベルを付加する。以降領域が広げられなくなるまで繰り返す。特徴として以下の点がある。

1. 領域間の濃度レベルの変化がなだらかな場合や、領域間のエッジに隙間が1点でもあると、2つの領域は統合されてしまう。
2. 統合の程度は閾値によって調節可能。閾値を小さくすればするほどこの領域の面積が小さくなり、領域数が増加する。

反復型領域拡張法は初期値、増分の値は画像に存在するため、前もっての決定が困難である。また、この方法は上記1.の欠点がある程度防ぐ。

分割手法として分離・統合法があり、中間レベルの分割画像から出発し、まず各領域を定数(平均濃度)で近似するアルゴリズムである。近似精度の低い領域は分離し、近似した隣接領域は統合する。

---

<sup>\*1</sup> MSE(平均二乗誤差) や、MAE(平均絶対値誤差) がよく用いられており、最も一致したものを参照するために、相互相関関数とよばれる関数を用いる場合もある [2]。

画素結合法は、隣接する画素間に何らかの関係を導入し、その関係の成り立つ画素対を順次結合する。この方式の利点は、結合の停止基準が明確であることであるが、画像のぼけや、雑音に弱いという特性がある。このため、前処理として平滑化を行うのが好ましい。

画像の不明瞭さ、高低周波ノイズのために、特徴量に曖昧さが存在すると、領域の細分化、過併合が発生する。弛緩法を用いた領域分割は、その曖昧さを局所的な矛盾が少なくなるように、並列反復的に除去する方法である。分類法として、確率的弛緩法や勾配弛緩法がある。

### 2.4.3 問題点

これまでに述べた領域抽出手法はごく一部ではあるが、このような領域抽出手法は、繰返しによる計算がほとんどで、計算量が膨大となる。また抽出画像の正確さなどの問題があり、これを解決するための手法の提案もなされているが、信頼度が高くなればなるほど、演算負荷が高くなる。

ブロックマッチング法の探索方法として、全検索、3段階総検索、最数検索、共役方向探索などがあるが、映像符号化における最適なアルゴリズムは現在も研究中である [2]。

また、領域抽出の自動化において、領域の分割や統合をどこで停止させるかという問題は、対象とする画像にも影響する。そのため分割数を自動的に決定することは困難であるといわれている [7]。

## 2.5 既存の画像圧縮手法

既存の標準映像圧縮処理には、MPEG や Motion-JPEG2000 がある。これらの圧縮伸張処理は図 2.3 に示すように、画像前後フレームの画面の違いを検出する動き検出・動き補償 (Motion-JPEG2000 には無い)、画像情報を周波数情報へ変換する周波数変換、視覚的に不要な高周波成分の情報を削減する量子化、生起確率の高い符号に短い符号を割り当て低い符号に長い符号を割り当てるエントロピー符号化、という手順で符号化され、復号化はその逆手順により実現する。

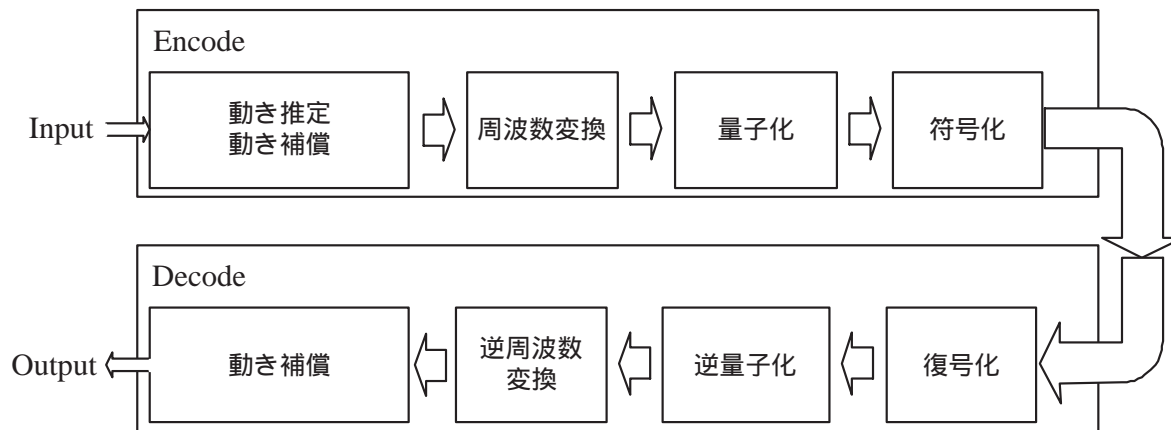


図 2.3 基本映像圧縮伸張処理

次節から標準映像圧縮処理である，MPEG と Motion-JPEG2000 について解説する．

### 2.5.1 フレーム間圧縮による映像符号化

フレーム間情報を圧縮する映像符号化の例として MPEG がある．MPEG とは，ISO/IEC(国際標準化機構/国際電気標準会議)の動画像音声符号化グループ (Moving Picture coding Experts Group) の名称である [2][4]．動画・音声のデジタル圧縮方式の標準化を目的として，世界のデジタル映像技術者が 1988 年に設立した．同グループによって標準化された動画像圧縮符号化方式にこの略称が用いられ，現在までに MPEG1，MPEG2，MPEG4 を標準化しており，現在 MPEG7，MPEG21 を標準化に向けて進行中である．

表 2.1 は現在までに標準化されている MPEG1，2，4 の用途や性能などを簡単にまとめたものである．

MPEG1 は，CD-ROM や DAT(Digital Audio Tape) などの蓄積メディアを対象としており，転送レートは最大 1.5Mbps，画質は VHS ビデオ程度の 45dB であり，Video CD やカラオケとして，ネットを介した動画通信用として広く使用されている．

MPEG2 は，より高画質を実現し，放送分野での応用を主目的として，より広範囲な応用を目指しており，転送レートは最大 100Mbps，画質は S-VHS ビデオ程度を実現している．

表 2.1 MPEG1, 2, 4 の比較

	MPEG1	MPEG2	MPEG4
標準化年	1992	1994	1999, 2000
目的	1 時間の映画を CD-ROM に保存	放送・通信を中心に広範囲に拡張	マルチメディアオブジェクトを時系列で表現
用途	ビデオ CD, カラオケ, VOD	DVD-Video, デジタル放送, VOD, 放送局内	携帯電話, Internet, デジタル放送, VOD
bit-rate	1.5Mbps (数 100kbps~1.8Mbps)	2Mbps~100Mbps	5kbps~38.4Mbps
解像度	352×240, 320×240	720(704・640)×480, 1920×1080, 1920×1152	多様, 最大 1920×1088
利用技術	JPEG (DCT 等), I/P/B ピクチャ(動き予測)	インタレース, I/P/B ピクチャ	オブジェクト合成, 合成映像, 合成音声

また, MPEG1 で対応していなかった放送やビデオに利用されるインタレース技法に対応し, 各種アプリケーション用に複数のプロファイルが定義されている。

MPEG4 は, MPEG1 よりも低ビットレートに対応した, 高圧縮率符号化方式である。主な用途は携帯電話, PHS, 電話回線, インターネットなど, 低画質・高圧縮の映像配信を目的としている。画像と音声を合わせて 64kbps 程度の転送レートを実現している。MPEG4 はこれまでのように動画や音声を信号ではなく, 内容の記述によって扱う, オブジェクト符号化方式を取り入れている。オブジェクトとしてデータを分別し, データを扱いやすくしている。

MPEG7 は, これまでのように映像圧縮方式を規定したものではなく, 検索などを目的とした記述方法の標準技術で, 正式にはマルチメディアコンテンツ記述のインターフェース (Multimedia Content Description Interface) とよぶ。コンテンツ内容や付加情報など

の記述についての標準技術となっている。コンテンツの特徴量を抽出・記述する技術と検索エンジンの技術が融合し、マルチメディアが利用し易くなる。EPG(Electronic Program Guide) \*<sup>2</sup> やホームサーバでの利用を想定している。

MPEG21(Digital Audio-Multimedia Framework) は、マルチメディアコンテンツの配信サービスと、コンテンツの利用について必要な、情報データ(メタデータ) \*<sup>3</sup> 形式に関する技術規格の標準化団体である。

### 2.5.2 フレーム圧縮による映像符号化

フレーム間情報の圧縮無しにフレーム毎に圧縮する映像符号化の例として Motion-JPEG2000 がある。JPEG とは、1990 年に ITU-TS(国際電気通信連合:旧 CCITT) と ISO(国際標準化機構) が共同で定めた、カラー静止画像の圧縮・伸張の標準化規格である。MPEG と同様に、規格を制定した標準作成委員会 (Joint Photographic Expert Group) が、そのまま規格の名称となっている。JPEG は静止画像に対する標準圧縮伸張方式であるが、この静止画像を映像の 1 フレームとして 1 枚ずつ圧縮し、高速に連続再生することで動画像として表示する方式として、Motion-JPEG がある。

JPEG をさらに高画質・高圧縮率にする符号化方式として、JPEG2000 が 2001 年に規格化された。従来の JPEG 方式では画像を離散コサイン変換 (DCT:画像を小さなブロックに分割して周波数成分係数を量子化・符号化する方式) で変換するが、JPEG2000 ではウェーブレット変換 (ウェーブレット関数で画像全体を周波数帯域に分けた縦横それぞれの周波数成分を量子化・符号化する方式) で変換する。これにより、JPEG で解決できなかったブロックノイズを解消し、可逆圧縮にも対応している。

Motion-JPEG と同様に、フレーム単位に映像を JPEG2000 圧縮する方式を Motion-JPEG2000 とよぶ。Motion-JPEG2000 は MPEG などと違い、フレーム間の動き予測など

---

\*<sup>2</sup> 「電子番組ガイド」の略。TV に番組表を表示するシステムで、番組表を、電波の隙間から専用の端末で受信する方式と、インターネット端末を利用する方式がある。衛星放送などで利用できる。

\*<sup>3</sup> 著作権情報、ID など、メディアそのものに付随する各種情報。

を行わずに圧縮するため，MPEG が苦手といわれている動きの激しい映像でも圧縮効率が高い．

### 2.5.3 問題点

標準映像圧縮伸張方式は，用途に応じて画質・圧縮率を向上可能な，汎用的な符号化方式である．しかし，動き推定・動き補償や，DCT・wavelet などの周波数変換は，演算における負荷が高く，量子化テーブル，符号化テーブルを映像に応じて適応的に作成する場合はさらに高負荷となる．

また，ネットワークで有料配信や，セキュアな映像を送受信する場合，標準化であるため，暗号化しなければ盗聴などにより簡単に復号可能であり，その上，暗号化処理による負荷も高速化を困難にしてしまう．

これを実用化するために ASIC <sup>\*4</sup> などの専用 IC が必要となるが，ASIC の場合，特定の用途に設計・製造し，回路を固定してしまうため，要件が異なる各種サービスへの柔軟な対応が不可能となる．

## 2.6 結言

映像アプリケーションの拡大により，多様な要求に柔軟に対応するためには，用途に応じた圧縮・伸張処理システムの実現が急務である．そのためには，利用者の立場から処理方式を考案し，解決することが最重要といえる．しかし，既存の映像圧縮・伸張処理は汎用性に優れるものの，用途によっては過剰な処理が必要となり，利用者にとって使いにくいものになってしまう．

本章では，人間の認知特性と準動画の共通点から，準動画に対しては，認知特性を活かした映像処理が有効であることを述べた．また，準動画の領域を抽出し各領域に対して処理する手法として，既存の技術利用を想定した場合，既存の領域抽出法，映像圧縮・伸張処理法

---

<sup>\*4</sup> Application Specific Integrated Circuit の略．特定用途向け集積回路．

の概要と問題点を述べた。

準動画の特徴から以上をまとめると、以下の3点の解決が重要であるといえる。

1. 領域の抽出手法

領域の位置情報を求めるための計算量が膨大となるため、この演算にかかる負荷がボトルネックとならないように考慮しなければならない。

2. 領域ごとの適応的映像処理

抽出した領域に対する圧縮・伸張処理においても、できるだけ演算量が最小となるほうが望ましい。

3. 各種サービスへの柔軟な対応

専用ハードウェアの場合高速であるが、用途ごとにASICを用意しなければならない。用途に応じてソフトウェアをインストールすることで、柔軟に対応可能なシステム構成が望ましい。

本研究では、1.における演算を最小にするため、利用者が最初に一度指定する方法を採用した。2.には単純な演算のみで圧縮・伸張を実現し、並列性も活かせるピクセルベース手法を採用し、3.には、組み込み型のハードウェアを用いることで対応した。次章で具体的な提案方式を詳細に述べる。

# 第3章

## 準動画向け超軽量映像処理

### 3.1 緒言

準動画は、人間の認知特性と同じように、注視領域と周辺領域に分離が可能である。領域を抽出する既存の方式では演算量が膨大となるため、映像圧縮システムの軽量化を目指すためには、できるだけ演算量が少なくなるような手法を取らなければならない。また、抽出したそれぞれの領域に対して行う圧縮・伸張処理も、演算時の負荷を最小にする手法が望ましい。

本章では、超軽量映像処理の概要を述べ、その中で用いる分離方法(領域抽出手法)と、抽出した領域に適用する映像圧縮処理方式を述べる。

次に、提案手法を用いた映像圧縮処理のアルゴリズムと、領域抽出手法から生じる、映像の不具合に対応するための手法を提案する。

### 3.2 超軽量映像処理

準動画は動きのある領域が局所的であることから、注視領域と周辺領域とに分離が可能である。このような映像を圧縮する場合、注視領域は情報量を多く、それ以外の周辺領域は情報量を少なくすることで、映像全体の圧縮率は高くなり、見たい部分の画質は保証され、高速な映像処理が可能となる。超軽量映像処理を実現するためには、領域抽出処理の軽量化、抽出後のそれぞれの領域の圧縮処理自体の軽量化が必要である。

領域抽出における問題点は、位置情報算出のための計算量が膨大なことである。近年の画



像認識の研究により領域の自動認識が可能となり、医療やリモートセンシングなどの分野で利用されつつある。しかし、この計算量を圧倒的に少なくする有効な方式はまだ確立されていない。そこで本研究はカメラアングルを固定した時点で、動きがあると予測される任意形状領域を指定する手法を採用する。

領域ごとの圧縮手法として、標準技術を採用する場合、DCT や wavelet などの周波数変換のような高負荷な処理が必要となる。そこで加減算とシフト演算のみで簡単に計算でき、並列処理が可能なピクセルベース圧縮手法を導入し、さらに、注視領域用、周辺領域用に改良を加えたものを提案する。

しかし本方式の欠点として、通常時に動きがあると予測し指定した注視領域のみを圧縮処理対象とするが、指定は最初の一回であるため、注視領域外へ動きがはみ出す可能性がある。そこで、物体が切れた映像となる状態を防止するために、映像を適応的に補間する手法を提案する。

### 3.2.1 領域抽出手法

前章で領域抽出手法の例として2種類述べたが、どれも準動画において有効な抽出手法とはいえない。

一方、近年の画像認識の研究において、複数カメラを用いて、人間など特定の領域を自動検出する方式などが報告されている [8][9]。具体的には、映像から得られる情報から動きベクトルを抽出し、動きベクトルで得られた領域を動いた領域として自動的に検出する。しかし、2章でも述べたが映像情報には様々なノイズが含まれており、このノイズを動きベクトルとして抽出してしまい、認識精度が落ちてしまう問題がある。これを解決するために、マスキングやフィルタリング、新たな認識方式の提案もされているが、自動的に動きを抽出するために、映像処理全体が高負荷となってしまう恐れがある。また、一部の映像圧縮処理においては映像認識を取り入れたシステムも現存するが、本稿が想定するような準動画に向けて映像圧縮処理の軽量化を図る上で、有効な領域抽出手法がまだ確立されていないのが現状である。

そこで本研究では、利用者がカメラ設置時に一度だけ領域を設定し、その後は通常どおり撮影する方式を採用する。利用者にとって自分で領域を設定しなければならないわずらわしさは残るが、そのわずらわしさは最初の一回であり、領域を計算する必要性も最初の一回でよく、しかも注目したい個所を利用者自身が選択するため、正確に領域を抽出可能となる。見たい部分は見易くそれ以外は通常以下の見易さとしたほうが、認知特性の観点から、人間がモノを認識する上で重要であるといえる。

次節では上記の方法で領域を抽出した後、領域ごとに行う映像圧縮伸張方式について述べる。

### 3.2.2 ピクセルベース圧縮伸張

ピクセルベースの圧縮伸張方式として、BTC(Block Truncation Coding)がある。図 3.1 は BTC の基本アルゴリズムを図示したもので、静止画像を  $M \times M$  の MB(マクロブロック)に分割し、MB を単位として、その平均値、標準偏差を元に、濃度差を表す量子化ビットを算出することにより画像を圧縮する。

図 3.2 は、BTC 処理を実行する自作のアプリケーションで、左側が符号化前の入力画像、右側が BTC 復号後の出力画像である。BTC はマクロブロックの基数  $M$  を 4 とすることで、比較的良好な画質と圧縮率を保証している。

しかし、以下のような映像の場合も想定され、総ての映像においてこの算出方法が有効とはいえない。

1. あるブロックに対応する映像の濃度差が極端に違う境界部分などには、演算時に利用するデータ(平均値、標準偏差)量が少ない。
2. あるブロックに対応する映像が平坦な背景などの場合、平均値のみでも十分に画質を維持可能であり、演算するブロック数も削減可能である。

本稿では領域に応じて画質、圧縮率を適応的に変動させる手法をとり、BTC の拡張方式を提案する。

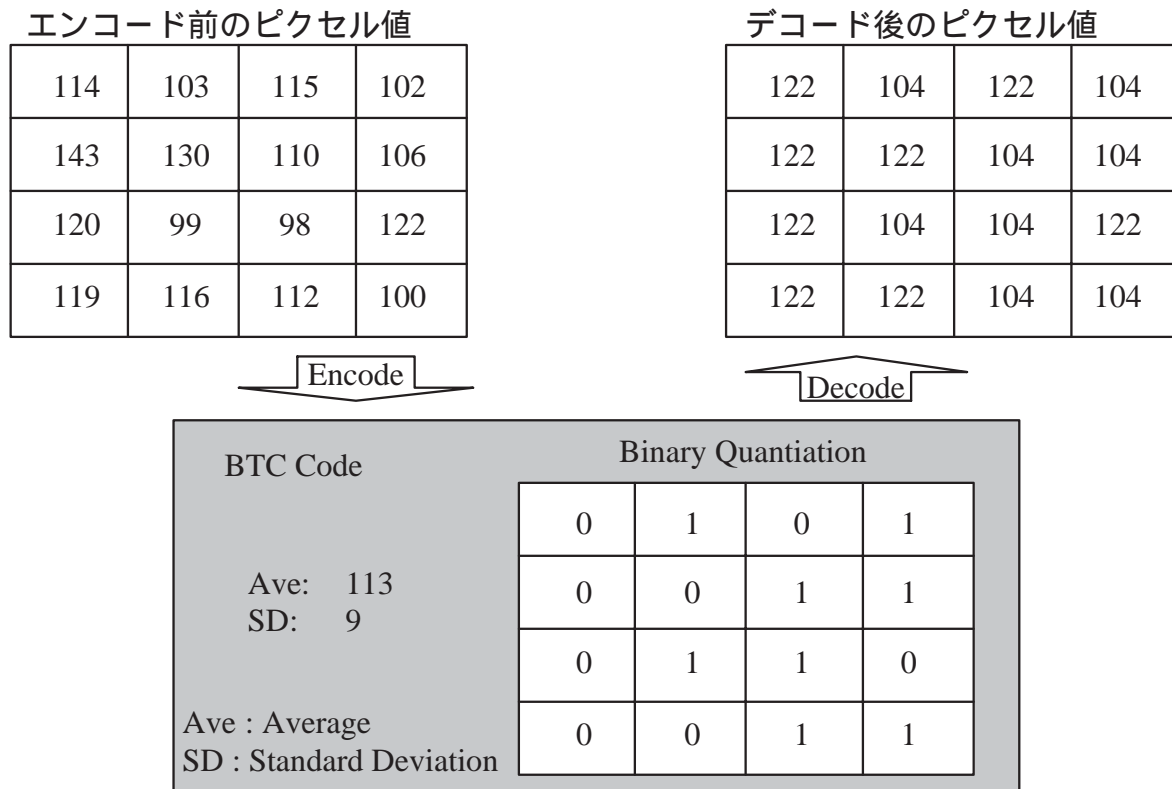


図 3.1 BTC の基本アルゴリズム

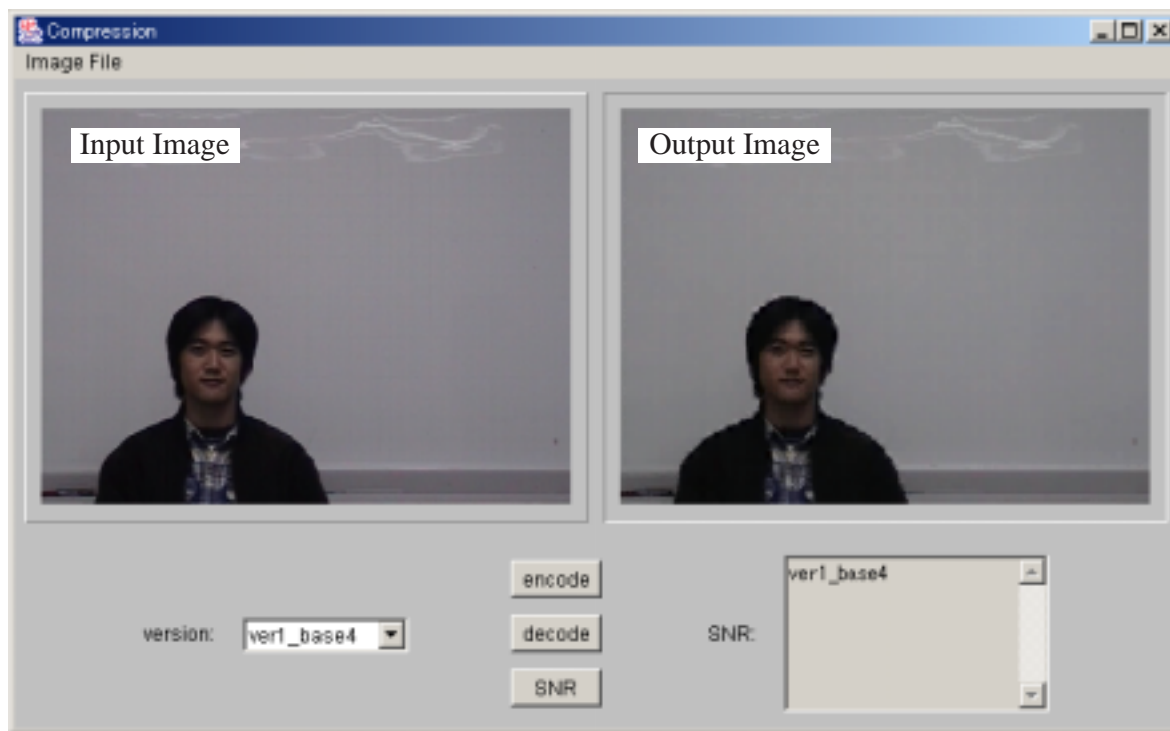


図 3.2 基本 BTC 処理結果画像

### 3.3 注視領域圧縮処理

BTC は  $4 \times 4$  の領域において、2 つの濃度しかもてないため、境界部分や、色の濃淡が激しい画像において、ぼやけた画像となる。これを防ぐために、MB において 4 つの濃度がもてるような演算手法を考案した。

つまり、 $M = 4$  の MB を用いて、4 値量子化を行う。

- 符号時

- 画素値が平均値 – 標準偏差以下であれば量子化ビットを 11 とする。
- 画素値が平均値以下であれば量子化ビットを 10 とする。
- 画素値が平均値 + 標準偏差より大きければ量子化ビットを 01 とする。
- 画素値が平均値以上であれば量子化ビットを 00 とする。

- 復号時

- 量子化ビットが 11 であれば画素値を平均値  $-(2 \times \text{標準偏差})$  とする。
- 量子化ビットが 10 であれば画素値を平均値 – 標準偏差とする。
- 量子化ビットが 01 であれば画素値を平均値 + 標準偏差とする。
- 量子化ビットが 00 であれば画素値を平均値とする。

表 3.1 量子化/逆量子化

$Q_{ij}$	量子化条件	復号式
11	$V \leq A - D$	$A - 2D$
10	$V \leq A$	$A - D$
01	$V > A + D$	$A + D$
00	$V > A$	$A$

上記アルゴリズムの符号化条件と、逆量子化時の復号式を表 3.1 に示す。ただし、MB の平均値  $A$ 、標準偏差  $D$ 、画素  $(i, j)[0 \leq i, j \leq M - 1]$  の画素値  $V_{ij}$ 、量子化ビット  $Q_{i,j}$  と

する．また，符号・復号化アルゴリズムを図 3.3 に示す．

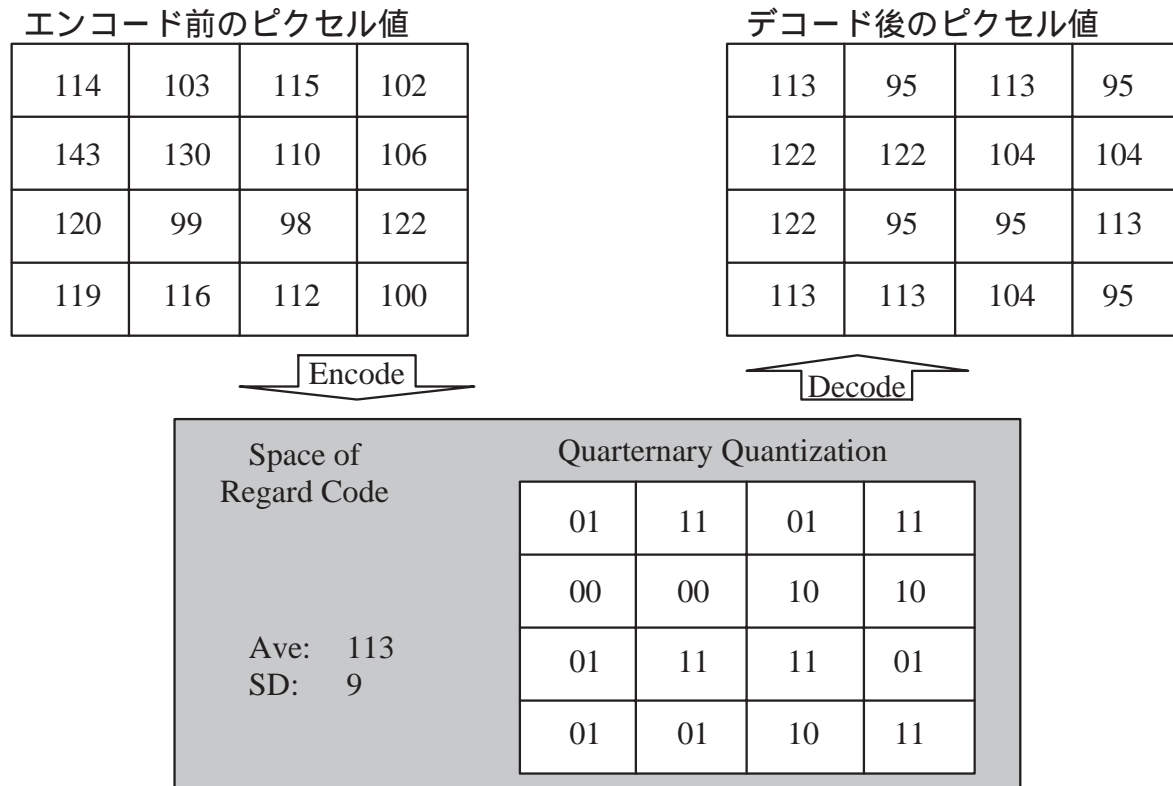


図 3.3 注視領域における符号・復号化アルゴリズム

### 3.4 周辺領域圧縮処理

周辺の領域は準動画の性質から，ほぼ静止状態であり，計算速度は高速であるほうが望ましい．また，認知特性から情報量は少なくてもよい．画質よりも圧縮率を優先した圧縮手法を提案する．

圧縮・伸張アルゴリズムを単純にするため，MB の基本単位を 8 に変更し，64 個の画素から平均値と標準偏差，2 値量子化ビットを算出する．

- 符号時

- 画素値が平均値以下であれば量子化ビットを 1 とする．
- 画素値が平均値より大きければ量子化ビットを 0 とする．

● 復号時

- 量子化ビットが1であれば画素値を平均値 - 標準偏差とする。
- 量子化ビットが0であれば画素値を平均値 + 標準偏差とする。

図 3.4 に周辺領域における符号・復号化処理のアルゴリズムを図示する。

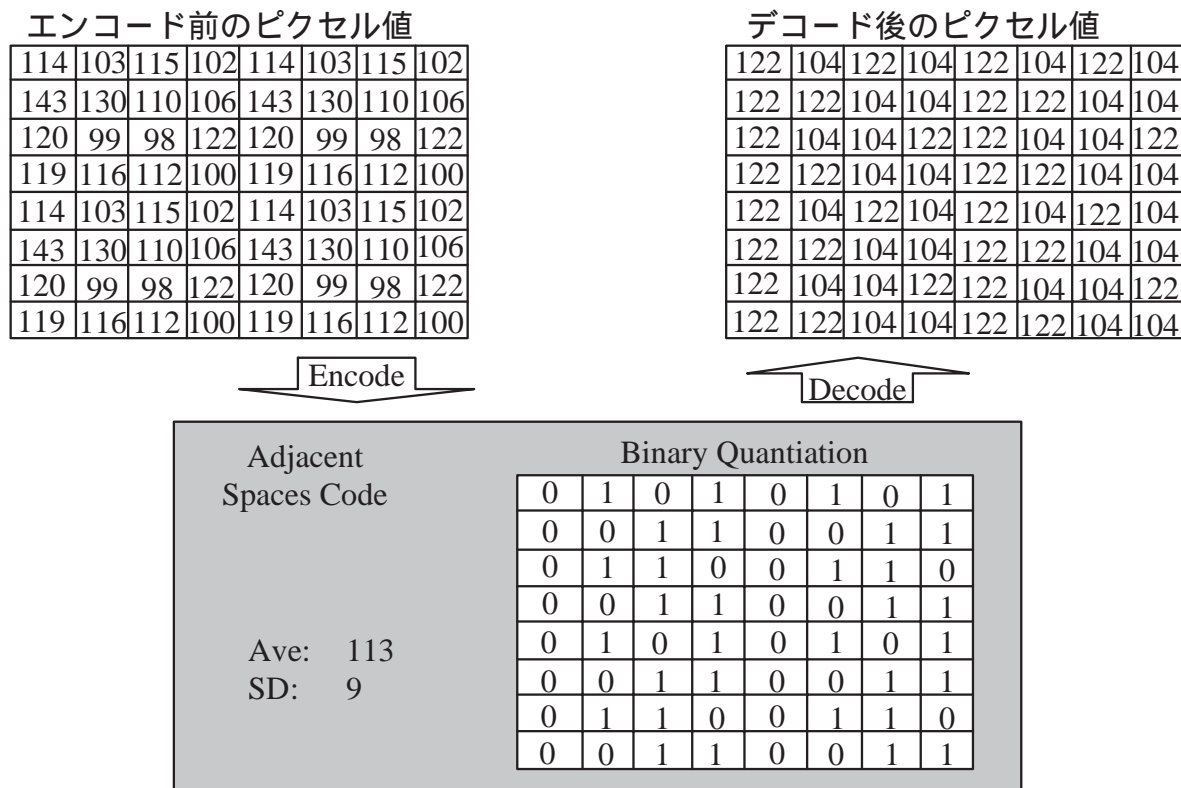


図 3.4 周辺領域における符号・復号化アルゴリズム

単純ではあるが、この方式の利点は通常の計算量で圧縮が可能なことである。準動画を対象とした場合、周辺領域は全画面に対して1/2以上は存在し、この領域における計算量がボトルネックとならないように軽量化を行っている。

### 3.5 領域別アルゴリズムの性能評価

表 3.2 は注視領域処理と周辺領域処理についての、Original BTC と比較した画質 (PSNR)<sup>\*1</sup> と圧縮率の性能評価結果である。また、主観評価のために、注視領域用提案手法の結果を図 3.5 に、周辺領域用提案手法の結果を図 3.6 に示す。

表 3.2 ピクセルベース圧縮手法の特性

	画質 (dB)	圧縮率
Original BTC	35.641	1/6
注視領域	36.462	1/4.8
周辺領域	28.312	1/13.7

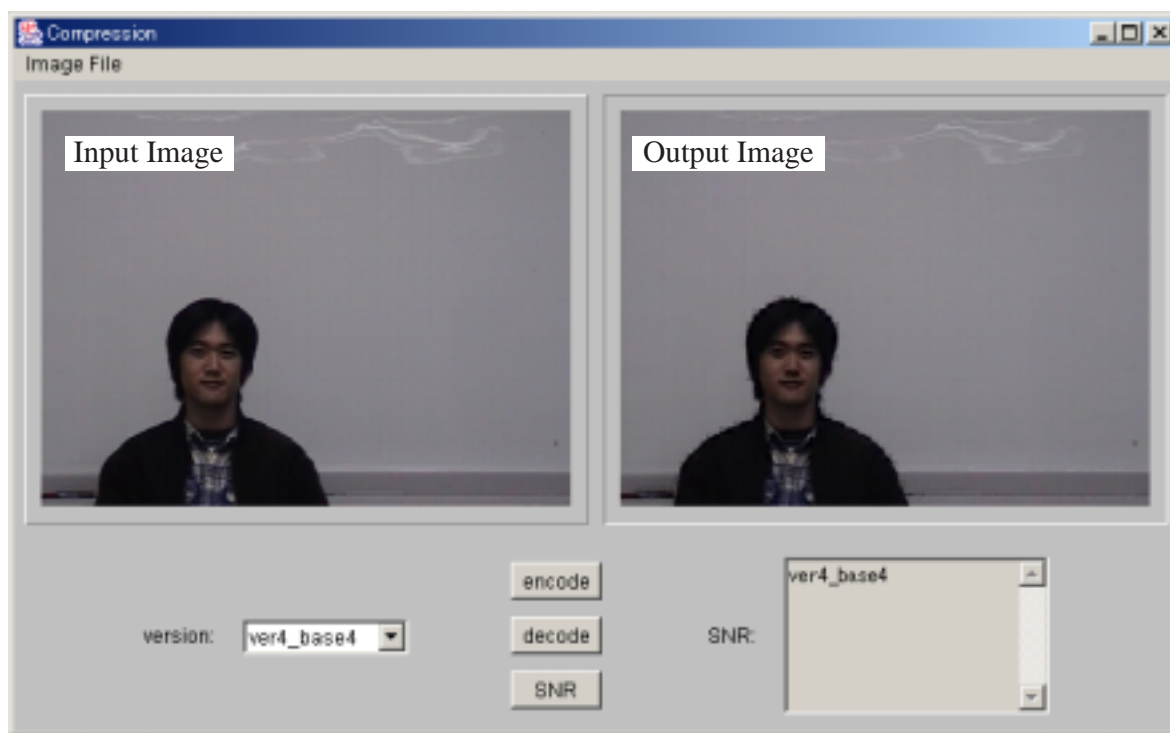


図 3.5 注視領域用処理結果画像

<sup>\*1</sup> Peak Signal to Noise Ratio の略。ピーク信号対ノイズ比。画質を客観評価するための一つの手法。付録 A 参照。

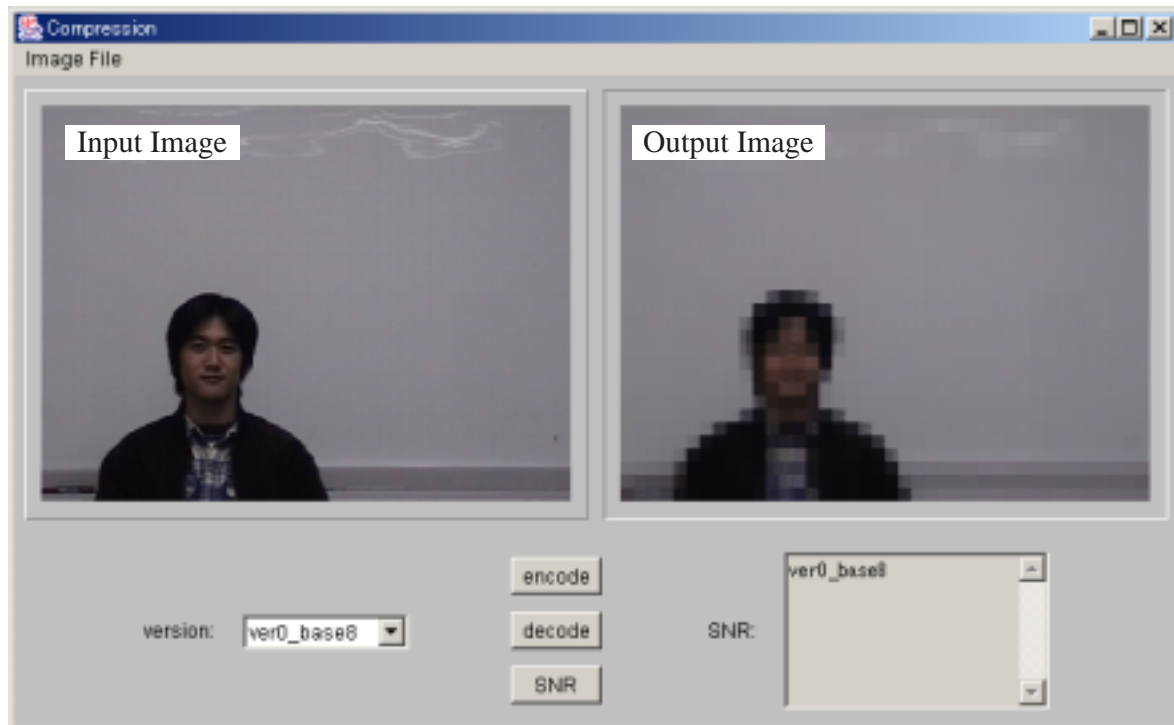


図 3.6 周辺領域用処理結果画像

本手法を用いれば，表 3.2 のように，領域に応じて，画質ならびに圧縮率を変更でき，画像に応じた適応的な圧縮が可能となる．

### 3.6 領域指定による問題点

準動画を注視領域と周辺領域に分離する手法として，カメラアングル固定時に，動きがあると予測される部分を利用者が指定する手法を採用した．これは，任意領域を演算する手間を省くことが可能となり，計算量を削減する利点がある．しかし，予測された領域から物体がはみ出してしまう可能性もあり，これに対処可能なアルゴリズムを採用しなければ，映像中に物体が切れた状態で圧縮されてしまう．

次節からは，これまでの提案手法を実現するための圧縮アルゴリズムの提案と，新たに生じた映像の不具合を解消するための圧縮アルゴリズムを提案する．もちろん，総ての映像に対してこのアルゴリズムが必要であるわけではなく，最初に指定した領域のみの情報が保証されればよいような利用法であれば，通常時の映像圧縮処理のみで実現可能である．本手法



は遠隔講義の利用を想定した場合、物体が切れたことによる、講義理解力の低下を防ぐための手法として提案する。

## 3.7 通常時の映像圧縮処理

最初のフレームのみ全画面を処理対象とし、それ以降は図 3.7 に示すように、あらかじめ指定された領域のみを通常のフレームレートで圧縮する。これにより、領域面積に反比例して、圧縮率を向上できる。



図 3.7 通常時の映像圧縮

具体的な符号化ステップを以下に述べる。

1. カメラ設置時に利用者がマウスなどの外部入力装置で注視領域 (動きがあると予測される領域) を指定する
2. 最初のフレームは画面全体を処理対象とする
3. 次のフレームからは最初に利用者が指定した動き領域 (図 3.7 の点線内部) のみを通常のフレームレートで圧縮
4. 領域外部は特定のフレーム間隔のみで処理し、領域位置情報も同時に圧縮する。

この方式を採用することにより、全画面を処理対象とするのはわずかな回数でよく、演算が高速となる。その上、毎フレーム単位で圧縮する個所は小さい領域内だけで良いため、格段に圧縮率が向上する。また、周辺領域を特定のフレーム間隔で更新することにより、周辺領域と動き領域との誤差による不自然さを小さくするメリットが生まれる。さらに、周辺領域圧縮と同時に領域位置情報も圧縮することにより、実時間通信の場合において、途中から通信に参加した場合でも復号が可能であるように考慮している。

### 3.8 動き領域の適応的伸縮

上述の手法では、図 3.8 のように、あらかじめ指定された領域の外部へ対象物が動く場合、対象物が切れた映像になる。遠隔講義の利用を想定し、学生の講義理解力の低下を防ぐ上でも、これを防止する手法が必要である。また、映像内の対象物の動きに適応して動き領域を任意に伸縮させ、領域形状を変更しなければ、速度・圧縮率が低下してしまう。この不具合を解決するために本方式では、下記のアлゴリズムを用いる。

1. 画像全体を  $N \times N$  個のタイルに分割する
2. 動き領域の境界タイルと画像全体の境界タイルで毎フレーム圧縮時に動きの有無を検査する
3. 境界タイルに属する境界 MB の左上隅の輝度データを、現フレームと直前フレームの間で比較する
4. 一致しない場合、その MB が属するタイルを動き領域に追加し、それ以降のフレームで圧縮対象タイルとする
5. 次のフレームからは、追加タイルについては差分を圧縮する
6. また、境界タイルが追加タイルの場合、3. の比較結果が一致すれば、動き領域から除外する

このように適応的に動き領域のタイルを追加・削除することによって、動き領域の形状を更新しながらの圧縮が可能になる。追加されたタイルの画質/圧縮率に関しては、用途に応

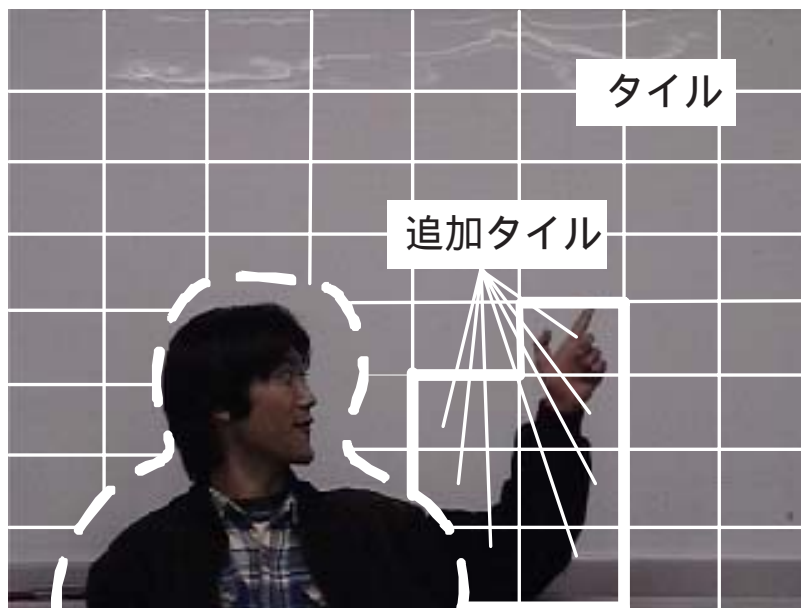


図 3.8 動き領域の適応的伸縮

じて，前章に述べたピクセルベース圧縮手法を使い分けても良い．

ここでの想定用途を遠隔講義とし，講義の理解度低下を防ぐための手法として提案したが，この処理を追加することより演算負荷が増加してしまう．遠隔監視を想定すると，重要個所（動く領域）はセキュリティ性の高い領域のみであり，それ以外に物体がはみ出しても必要性は低いはずである．そのため，この提案は用途の要求性能に応じて使い分けることが必要である．

### 3.9 結言

準動画の特徴として，動きのある領域が局所的であることから，映像を注視領域と周辺領域に分離可能であり，これにより有効な映像圧縮処理が実現可能となる．実現する上での問題は，領域抽出手法の複雑さ，各領域における圧縮処理の演算量である．

これを解決するため本章では，超軽量映像処理の概要を述べ，以下の 2 点を提案した．

- カメラ設置時のみの領域抽出手法
- 抽出後の領域における 2 種類のピクセルベース圧縮手法

ピクセルベース圧縮手法の評価結果から，領域に応じて画質並びに圧縮率を変更でき，映像に応じて適応的に圧縮可能となることを示した．

次いで，前章までの提案方式を組み込んだ準動画向けの映像圧縮アルゴリズムを提案した．また，領域抽出処理に指定することで計算量を減少させることができるが，この手法に起因する映像の不具合を想定し，物体が指定領域からはみ出した場合に物体を補間する手法を提案した．

以上の提案から本手法には以下の利点がある．

- 領域の大きさに応じて適応的に圧縮率/画質を変更可能
- 領域からはみ出し時に動き領域の形状を伸縮しながら圧縮可能

これらの利点により，超軽量映像圧縮処理方式の実現が期待できる．次節で本提案の定量的な評価を述べる．

# 第 4 章

## 性能評価

### 4.1 緒言

2 章では準動画の特徴と人間の認知特性の共通点を述べ、準動画を映像圧縮する場合は認知特性を活かした圧縮法が有効であることを述べた。3 章では準動画を注視領域と周辺領域に分離する上で必要となる、領域抽出手法と、抽出後に行う圧縮処理手法の提案を行い、アプリケーションによる性能評価を示した。また、提案した領域抽出手法から生じた問題点の解決法も含めた、映像圧縮処理方式を提案した。

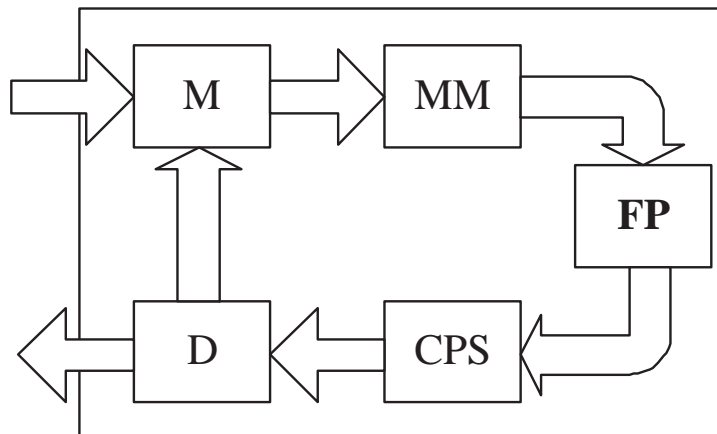
本稿では、これまで述べてきた圧縮処理方式を、準動画像圧縮処理システムに構築するために、評価用チップとして DDMP を選択した。DDMP は組み込みシステム向け汎用プロセッサであり、超高速・極省電力という利点を持つ。

本章では、DDMP シミュレータの原理とシミュレーション手法、実装手法、画面全体に対する動き領域の割合別に、圧縮率・速度の評価結果について述べる。

### 4.2 シミュレータの原理

本提案手法は任意領域の位置情報を最低一度は計算しなければならない。また、フレーム毎に動き領域の境界タイルを検査する必要がある。これらの処理によるオーバーヘッドが可能な限り低減できる実現法を採ることが望ましい。そこで、画素の位置情報による多数の連想処理が並列に実現可能な DDMP を利用し、提案方式を実装した。その他の DDMP の特徴として以下があげられる。

1. 組み込みシステム向けの汎用プロセッサであるため，本システムのプログラマブルな利用が可能となる．
2. 自己タイミング型のアーキテクチャを持つため，非同期に動作し，独立したデータはパイプライン中を並列に動作する．本方式においてピクセル単位，MB 単位，タイル単位，フレーム単位は，各々独立しており，それぞれの並列性を活かすことができる．
3. C 素子のハンドシェイクにより，パイプライン間のデータを受け渡すため，配線長が短くてすみ，省電力である．本システムを戸外で利用を想定した場合，省電力であればカメラまたはモニタの設置場所の制限が少なくなる．



MM : Matching Memory  
 FP : Functional Processing Unit  
 CPS : Cache Program Storage  
 M : Flow Merging Module  
 D : Flow Diverting Module

図 4.1 DDMP のナノプロセッサ構成

DDMP の基本アーキテクチャを図 4.1 に示す．DDMP は，従来のノイマン型プロセッサとは異なるアーキテクチャを採用することで，高速処理・低消費電力を実現している．ノイマン型プロセッサがプログラムカウンタにより逐次的に命令を実行するのに対し，DDMP は行き先情報，世代識別子などを含むパケット形式のデータを扱う．パケットの行き先情報は図 4.1 の D において参照され，データの行き先を決定する．行き先情報は処理内容に応じ

て CPS で変更され、データは独立してプロセッサ内を巡回する。

MM において、処理対象データが揃った時点でデータが読み出され、FP 部において演算処理される。

DDMP の超高性能は複数のナノプロセッサを 1 チップに集積したことによる、空間並列性と、ナノプロセッサレベルでのパイプライン処理による並列性にある。図 4.2 は、DDMP の並列性を表した概念図である。x 軸は時間を表し、y 軸はパイプライン的な並列度、z 軸は空間的な並列度を表している。

DDMP のパイプラインは自己タイミング型パイプライン機構となっている。これは各々のデータが各々の処理タイミングで動作するため、処理に必要な部分のみを動作させることができ、省電力化できる。また、複数のパイプラインを並列に動作させ、かつ、連結して長大なパイプラインを形成できるため、柔軟な処理変更が可能であり、スループットを向上できる。

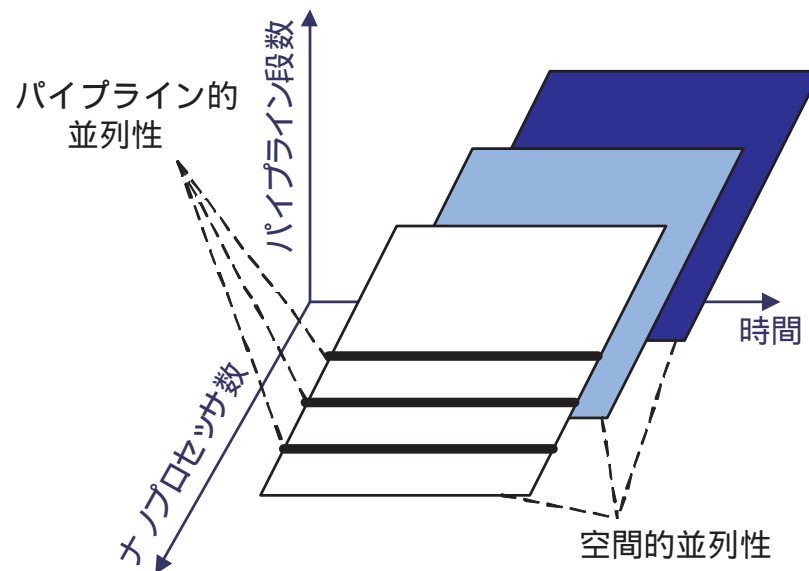


図 4.2 DDMP の並列概念図

本研究の性能を評価するために 10 個のプロセッサを搭載した DDMP チップ構成を、ソフトウェアにより再現した、DDMP10A フローグラフシミュレータを用いる。このシミュレータは DDMP チップ構成を元にしたフローグラフシミュレーションツールであり、プロ

グラムはフローグラフを記述することにより実現する。

DDMP10A のチップ構成は 10 個のプロセッサがそれぞれルータにより相互結合されており、図 4.3 に DDMP10A の内部構成を示す。なお、ALP という算術演算プロセッサが 5 個、LCP という論理演算・制御プロセッサが 3 個、CVP という外付け SDRAM アクセスプロセッサが 2 個、それぞれがルータに相互接続された構成となっている。

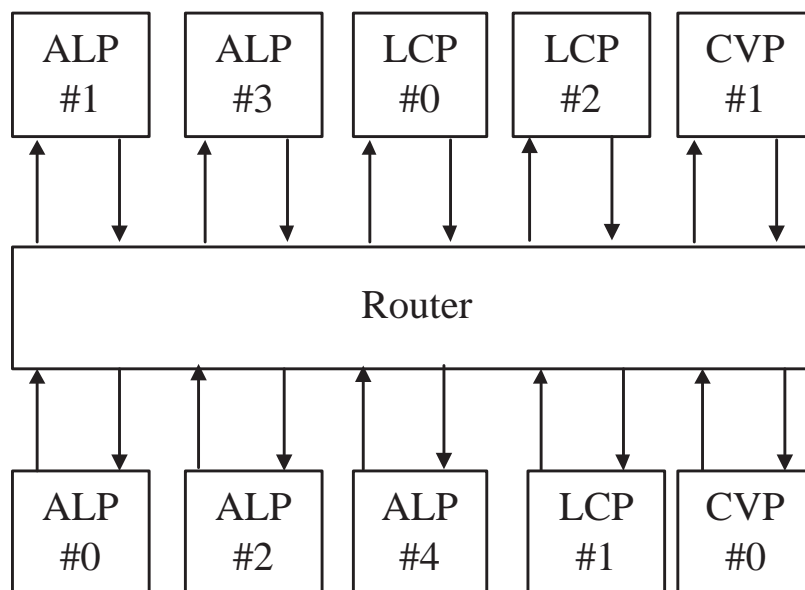


図 4.3 DDMP10A の内部構成

入力データは、1 データが投入される時刻、世代番号とよばれるデータ識別子、入力ポートを表すエントリ番号、データなどの情報を 1 パケットとしている。1 パケットには最大 2 つの 32 ビットデータを保持することが可能で、2 つのデータが互いに独立して扱える場合、二並列に演算可能となる。この仕組みを二世代化とよび、これを利用することにより、フローグラフ中を流れるパケット数を減少させることができる。

### 4.3 シミュレーション手法

シミュレーションにおける入力データの選定に際して、以下の点に留意した。

- 任意形状の動き領域の割合別に評価



本提案手法は動き領域の割合が小さいほど、圧縮率・速度の向上が図れる。逆に、動き領域が画面全体である場合、毎フレーム毎に全画面を圧縮処理対象としなければならないため、Motion-JPEG2000と同じ処理となり、高性能化は期待できない。この予測を定量的に示すために、領域の割合別に評価を行う必要がある。

- 入力データパケット数が最大時の性能を評価

DDMP のアーキテクチャより、処理の終了したデータについては実行しないため、プログラム中の負荷は、実行されているノードだけに限定されてしまう。このため、入力データ量が少ないと、プログラムにかかる負荷が均等にならないため、正確な性能の評価にはならない。そこで、データ数をプログラム中の各 PE(Processing Element) にかかる実行段数より多くしておく必要がある。つまり、最低でもデータが入力されて、最初の 1 画素が出力されるまでは、データを入力し続ける必要がある。

以上のことから、動き領域がある割合で一定になるようなデータ群を数パターン用意し、入力データ数は各 PE の命令実行段数 (速度) より多いデータを入力しなければならない。

そこで、動き領域のパターンとしては、全画面が動き領域 (割合 1)、動き領域が画像の 1/2(割合 0.50)、動き領域が画像の 1/3(割合 0.33)、動き領域が画像の 1/4(割合 0.25) の 4 パターンを用意する。この 4 パターンの入力パケットによって、動き領域の割合における性能を比較する。

入力データ数は、プログラムの実行速度と、入力データの投入速度によって決定する。例えば、 $64 \times 64(64)$  画素を圧縮処理する場合、入力パケットは以下の式により、2048 パケットとなる。

$$64 \times 64(\text{総画素数}) / 2(\text{二世代化により 1 パケットに 2 データ格納}) = 2048(\text{入力パケット数})$$

また、パケットの投入間隔が 45 ステップ\*1 とすると、

---

\*1 1 ステップ=3.5nano sec. DDMP10A は各プロセッサ内を 140MHz で動作し、ルータ内を 280MHz で動作する。

$$45(\text{入力パケット投入間隔}) \times 2048(\text{入力パケット数}) = 92160(\text{ステップ})$$

となり，最初に出力されるデータが，このステップ時刻より早い時刻で出力されなければならない。

## 4.4 実装手法

実装に際しては，MB 単位，ならびに，MB 自身の符号化処理の並列処理性を活用するために，10 個のプロセッサを搭載した DDMP チップ上で並列処理可能なプログラムを作成した．具体的には以下である．

1. 投入パケットの入力は水平スキャンライン方向順とし，RGB を 1 データとして，隣り合う 2 ピクセルのデータを 2 世代化して 1 パケットに格納した．これにより，空間的冗長性の削減を行う．
2. 色情報 (YCrCb) は各々独立で演算が可能であるため，YCrCb 変換開始後は独立して行えるため Y 値，Cr 値，Cb 値を並列に処理可能である．
3. MB 単位は各々独立で演算が可能であるため，MB の特定のピクセル情報が揃った時点で演算を開始可能とする．
4. 標準偏差と 2 値量子化あるいは 4 値量子化の演算は互いに独立で行えるため，並列に動作させ，時間的冗長性を削減する．
5. 処理中の各 PE にかかる負荷を均一にするため，PE ごとに流れるパケット数とその平均レートを算出し，割合に見合った PE の割り当てを行い，10 個の PE を最大限に活用する．

また，注視領域演算用フローグラフモジュール概念図を図 4.4 に，周辺領域演算用フローグラフモジュール概念図を図 4.5 に示す．

双方同じようなモジュールで作成することができ，また，モジュールにかかる負荷も同程度となるようにプログラムしている．

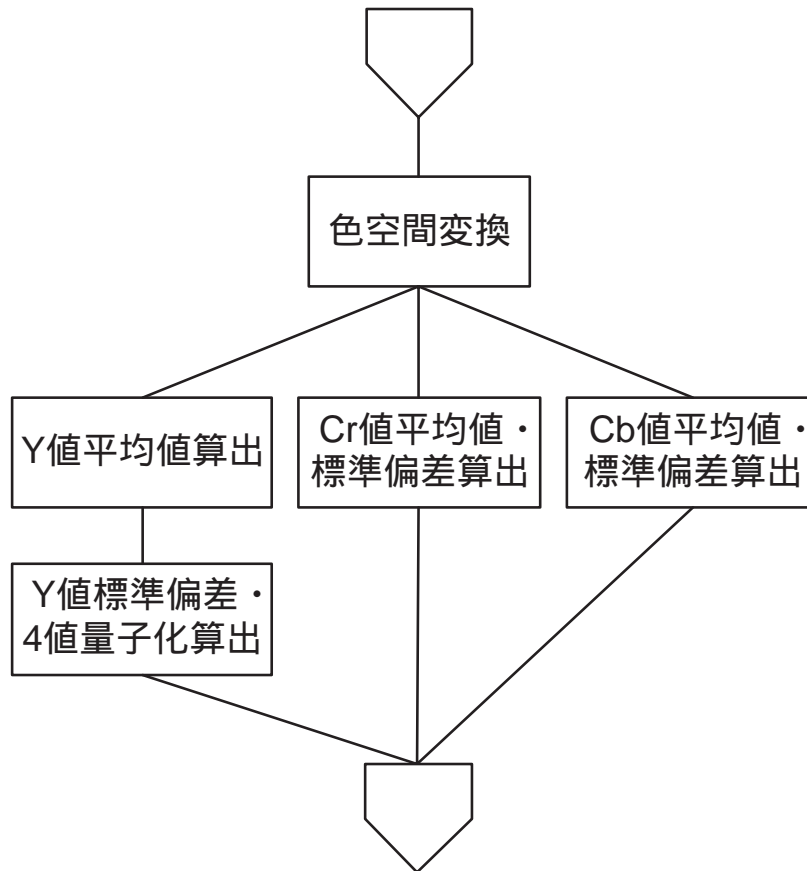


図 4.4 注視領域用フローグラフ概念図

## 4.5 測定手法

投入する画像サイズは VGA サイズ (640 × 480) とし、画像全体における指定領域の割合 (1, 0.50, 0.33, 0.25) による、エンコード時にかかるスループット、並びに圧縮率を算出する。

測定式は以下のものとする。

### 1. スループット

$$\frac{\text{水平 1 ラインの MB 数} \times \text{ブロック内ピクセル数}}{(t_2 - t_1) \times 3.5ns} (\text{PixelRate/sec.})$$

$t_2$  は定常状態の (320, 8) が含まれる MB 中で最初に出力されるデータのステップ時刻

$t_1$  は定常状態の (320, 4) が含まれる MB 中で最初に出力されるデータのステップ時刻

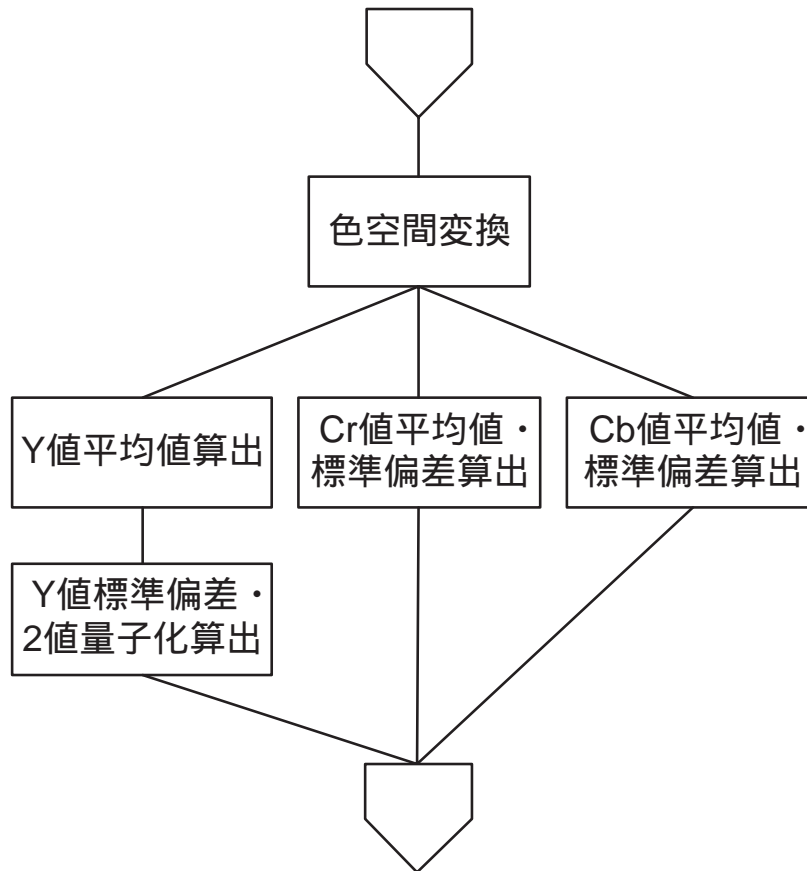


図 4.5 周辺領域用フローグラフ概念図

## 2. 圧縮率

30 フレーム中 1 回の割合で全画面を圧縮すると仮定し、

$$\frac{[30 \times (\text{注視領域の } 1MB \text{ 中に必要なビット数} \times \text{注視領域の } MB \text{ 数}) + \text{周辺領域の } 1MB \text{ に必要なビット数} \times \text{周辺領域の } MB \text{ 数}]}{\text{全フレームビット数}}$$

1. は、コンピュータシステムの処理能力を測る評価指標であり、本稿では、一定時間内にいくつの MB が処理されるかによりスループットを算出する。図 4.6 にスループットを算出するためのイメージを図示する。図に示すように、スループットはパケットが入力されて出力される間隔が、定常状態となったデータを抽出して算出する必要がある。また、画像右隅のブロックのデータが出力されてから、画像左隅の MB のデータが開始されるまで MB の

処理は行われなくなり、フローグラフ中で休止状態のノードが多くなる。平均的な出力間隔を算出する必要があるため、画像の中心位置(図中  $t_1, t_2$ ) をスループット算出に用いている。

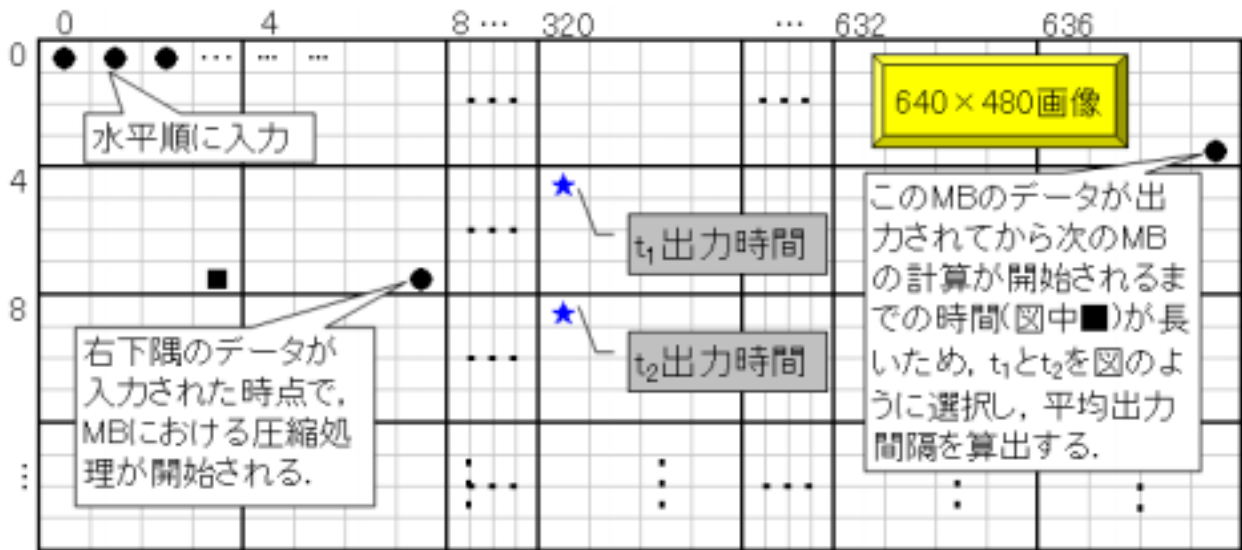


図 4.6 スループット算出手法イメージ図

2. は映像圧縮・伸張処理でしばしば利用される評価指標である。表 3.2 から指定領域の大きさに応じた圧縮率を算出した。

次節でスループット性能、並びに圧縮率を領域の割合ごとに算出した結果を述べる。

## 4.6 性能評価

3 章で提案した二種類の領域用圧縮手法を、アプリケーションにより実現し、その画質・圧縮率を評価したが、処理速度に関しては評価できていない。そこで提案手法の速度、並びに動き領域の割合における圧縮率を評価する。なお、符号化処理のアルゴリズムは復号化処理よりも演算待機時間が長く、復号処理時のスループットより遅いことが、アプリケーション作成時の演算数などを用いた評価予測からわかっている。そこで評価に用いた部分は、本提案システム中で処理負荷の高い符号化部分のみとなっている。

表 4.1 は、本提案の符号化部分を DDMP10A シミュレータで実装・実行し、動き領域の

割合に対する符号化時のスループット性能の結果を表している。

表 4.1 動き領域の割合に対するスループット性能

動き領域の面積比 (%)	0.25	0.33	0.50	1
スループット (frame/sec.)	156.25	117	78.12	39.06

この評価結果から、動き領域の割合が最大全画面 (VGA サイズ:640 × 480) であった場合でも、39fps の速度で符号化可能であり、準動画に対して十分な処理速度が実現できているといえる。また、動き領域の面積が小さいほど高速に処理可能であるという結果を示した。

2章で述べたように、準動画は速度要求は高くないが、符号時にこれだけのスループットが実現可能であれば、蓄積型の配信だけでなく、ストリーミングなどのリアルタイム通信への応用も可能といえる。

表 4.2 は、動き領域の割合に対する圧縮率の結果を、表 3.2 を用いて算出した。

表 4.2 動き領域の割合に対する圧縮率性能

動き領域の面積比 (%)	0.25	0.33	0.50	1
圧縮率 <sup>-1</sup>	18	14	9	13

この結果から、動き領域が 1/3 以下の時に圧縮率が高いことがわかる。しかし、1/2 になると急激に圧縮率が下降し、動き領域が画面全体となると動き領域が 1/3 の時と同程度の圧縮率となる。これは、注視領域用圧縮は周辺領域用圧縮に比べて、 $M = 4$  の MB の持つビット数が約 3 倍となっているため、動き領域が 1/3 以下になると圧縮率が下がる傾向にあると考えられる。

図 4.7 は、これまでの 2 つの表をまとめてグラフ化したもので、VGA サイズの動画を対象として、動き領域の平均占有率に対する圧縮性能および圧縮率を測定した結果である。

比較として、同じフレーム単位の圧縮方式である、Motion-JPEG2000 の圧縮率を同表にプロットしている。

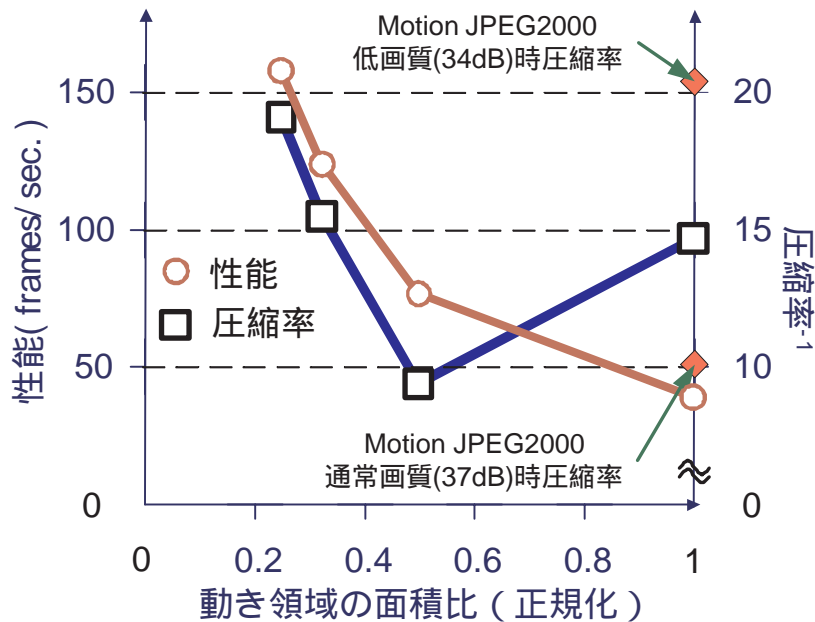


図 4.7 超軽量映像圧縮方式の性能

通常画質 (37dB 時) において圧縮率は低くなるが, (34dB 時) においてはかなり圧縮率が高くなる。また, Motion-JPEG2000 では可逆圧縮が可能であるため, 40 ~ 50dB まで表現可能であるが, 本提案は 36dB 程度が標準画質である。

速度性能は Motion-JPEG2000 の比較資料が無かったために評価できていないが, 通常の MPEG などのフレームレートが 30fps であるのに対して, 本提案において動き領域が画面全体の 1/4 の時, 150fps<sup>\*2</sup> を実現可能であるのは驚異的といえる。

本提案と同画質にした場合, Motion-JPEG2000 より圧縮率が優れており, 速度性能は動き領域の大きさに依存するが, 最低でも 39fps の速度性能を実現した。

\*2 圧縮時に処理可能な速度性能を示している。復号時にこの速度で表示するためには, ディスプレイのリフレッシュレート (垂直走査周波数) が対応していなければならない。70fps(Hz) 以上でちらつきが気にならない周波数である。

## 4.7 結言

2章で準動画を対象に，人間の認知特性を活かした映像圧縮処理の有効性を述べ，3章で領域の抽出法，並びに各領域への映像圧縮・伸張方式の提案を行った．さらに，適応的に領域の圧縮が可能なように，動き領域を伸張して圧縮する手法を提案した．本章では，本提案を組み込みシステム向けで，領域の演算時の負荷を軽減する，データ駆動型プロセッサ DDMP 上で実装し有効性を評価した．

評価した項目は以下の3つである．

- 動き領域の割合に対するスループット
- 動き領域の割合に対する圧縮率
- 既存の標準技術との比較

その結果，画面に対する動き領域の占める割合に応じて，圧縮率，速度が変化し，動き領域が画面全体の  $1/3$  以下の時に性能が良いという結果が得られた．既存の標準技術である Motion-JPEG2000 と画質を同じにし，圧縮性能を比較しても，圧縮率が高いことを実証した．



# 第 5 章

## 結論

高性能 LSI 技術やネットワーク技術の進展に伴い、映像メディアが様々な用途に活用され、多数の用途に要求される性能を満たすべく標準方式も普及している。本研究では、これら多数の用途総てに対応するのではなく、用途に応じた映像圧縮処理について考察し、利用者の視点から最適な映像圧縮・伸張システムを構築するべく、準動画向けに超軽量となる映像処理方式の検討を行った。

準動画は、通常は静止状態の映像で、ある瞬間だけ動画像として表示するような映像で、動画像として動く部分は局所的な部分のみであり、人間の視野と同じく、注視領域と周辺領域とに分離が可能である。準動画を最適な方式で圧縮するために、まず人間の認知特性という性質から、最適な映像処理システムを考案した。認知特性とは、人間の視野を注視領域と周辺領域とに分離すると、注視領域の情報伝達が保証されれば、周辺領域の情報量は少なくともよいというものである。そこで、人間のもつ認知特性を活かして、準動画の注視領域には情報量を多く、周辺領域には情報量を少なくすることで、適応的な画質・圧縮率・速度の実現が可能となる。本研究では遠隔講義や遠隔監視など、動く領域が小さい準動画を対象として、超軽量な映像圧縮・伸張処理システムの構築を目指す。

この方式では注視領域と周辺領域を分離するために領域抽出を行う必要がある。しかし、古くから研究され提案されている領域抽出手法は、高性能なものほど計算が複雑で、準動画向けに軽量化するには向いていない。そこで演算量がボトルネックとならないように、カメラアングルの固定時に、利用者に動きがあると予測される部分に関して、あらかじめマウスなどの外部入力装置を利用して指定する手法を採用した。領域を計算しなければならないのは、最初だけで、後は毎フレーム単位を領域の内部で注視領域用の処理、特定のフレーム間

隔のみを外部で周辺領域用の処理と切り替えるだけでよい。

また、領域ごとに情報量を変化させるために、抽出した領域の要件に応じた映像圧縮を行う必要がある。映像圧縮処理方式には標準技術が存在し、現在までに標準化されている技術は、非常に多用途において利用されている。しかし、標準技術は多用途での利用を想定しており、用途によっては過剰な仕様となる恐れがある。現在実用化されているほとんどの標準技術は、ASIC を利用したものとなっているが、専用プロセッサに固定してしまうと、各種サービスへの柔軟な対応が困難となり、対応するためには、チップを最初から作り直す必要がある<sup>\*1</sup>。

そこで、圧縮・伸張における処理が単純で、並列処理可能なピクセルベースの圧縮手法を採用した。本稿では、このピクセルベース圧縮のアルゴリズムを改良し、注視領域用、周辺領域用の二種類の提案を行った。注視領域用処理は、MB における量子化を 2 値ではなく 4 値とし、MB 内で表現可能な色数を増加させた。周辺領域については、高速・高圧縮が要求されるため、MB の基数を 8 として、MB の持つ情報量を削減し処理の軽量化を図った。

基本の圧縮アルゴリズムは、利用者が最初に動きがあると予測した領域を指定し、通常時はその内部を圧縮対象とし、それ以外を周辺領域として一定間隔で処理対象とする。しかし、このアルゴリズムでは、指定した領域から物体がはみ出した時、物体が切れた映像になってしまう。

本研究ではこれを補間する技術として、適応的に動き領域を伸張して方式を提案した。アルゴリズムを以下にまとめる。

1. 画像を  $N \times N$  のタイルに分割し、毎フレーム圧縮時に動き領域の境界タイルと、画像全体の境界タイルで動きの有無を検査する。
2. 検査は境界タイルに属する境界 MB の左上隅の輝度データを、直前のフレームと比較する。
3. 比較が一致しなければこの MB が属するタイルを、動き領域に追加し、それ以降のフ

---

<sup>\*1</sup> ここではフルカスタム IC を指すが、配線を切り替えるだけで要求に合わせるセミカスタム IC もある。

フレームで圧縮対象タイルとし、次フレームからは差分のみの圧縮とする。

4. 境界タイルが追加タイルで、かつ 3. の比較結果と一致すれば、動き領域から除外する。

これにより適応的に動き領域のタイルを追加・削除可能となり、動き領域の形状を更新しながら圧縮可能となる。この処理は、検査による計算が必要となるため、画質/圧縮率/速度の要求に応じて、利用を切り替えた利用法が望ましい。

以上の提案手法を実装する場合、位置情報の計算が、最低一度は必要であり、フレームごとに動き領域の境界タイルを検査する必要がある。そのため、この処理によるオーバーヘッドが最小となることが望ましい。また、ASIC を用いることを前提としたものではなく、各種多様なサービスが次々と頻出するような今日においても、柔軟に対応可能なように、組み込み型のプロセッサの利用が望ましい。

そこで本研究では、DDMP を用いてシミュレーションし、符号化部分に関して評価を行った。DDMP は組み込みシステム向け汎用プロセッサであり、世代番号とよばれる位置識別子を有し、位置情報による多数の連想処理が並列に実現可能で、領域抽出時の負荷をさらに軽量化する。さらに、ピクセルベースの特徴である、MB 単位の並列性、画素値単位の並列性、フレーム単位の並列性が有効に利用可能なアーキテクチャを持つ。極省電力・超高速という有利な利点も持つため、本システムを戸外で利用する場合でも最適に利用可能である。

シミュレーションによる評価は、MB におけるスループットからフレームレートを算出した。その結果、画面全体に対する動き領域の占める割合に応じて、速度が変化し、動き領域が小さいほどフレームレートが向上することを示した。圧縮率に関しての評価は、動き領域のサイズが画面全体の  $1/3$  以下の時に、最適な圧縮率を示した。既存のフレーム単位圧縮である Motion-JPEG2000 の性能と、画質を同じにして評価した結果、本提案の方が圧縮率が高いという結果を得た。

映像処理の普及に伴い、映像に対する様々な要求が生まれている現在、各種サービスに柔軟に対応するためには、総ての用途に対応できるような標準技術の方式ではなく、特定用途に限定した圧縮処理方式を考慮することが重要である。これにより用途ごとに高い処理性能

を実現でき、人間にとって使いやすいものになる。

最後に今後の課題を述べる。今回提案した映像圧縮・伸張処理方式は、超軽量という名にふさわしいスループット性能を示した。しかし、本評価は符号処理のみであり、システム全体に渡る評価は不完全といえる。例えば復号処理を1チップで同時に行うことを考慮したプログラムの評価など、使用するプロセッサ数を少なくし、小型軽量化を考慮した検討をする必要がある。また、注視領域用、周辺領域用の圧縮アルゴリズムは、映像によってその画質の要求性能は変動するため、数パターンの評価映像に対して評価をする必要がある。今後はDDMP実機を用いて、実用的な各種動画の圧縮性能について、詳細に評価する予定である[10]。

# 謝辞

本研究において、指導教員として懇切なる御指導、御鞭撻を賜った高知工科大学の岩田 誠教授に心より感謝の意を表します。

清水 宏明 教授には、副指導教員として、日頃から温かく見守って頂き深く感謝いたします。

岡田 守 教授にはお忙しい中、本研究で利用した画像処理の基礎を助言して頂き、心より感謝いたします。

情報システム工学科 大森 洋一 助手には、本研究の重要な御指導、御示唆を頂きました。ここに感謝の意を表します。

情報システム工学科の諸先生方には、講義及び研究において御指導、御助言を頂きました。ここに感謝の意を表します。

シャープ (株) の宮田 宗一 氏、浦谷 宗宏 氏、芳田 真一 氏、中野 貴彦 氏には、共同研究として大変お世話になり、本研究考案の糸口となりました。心より感謝いたします。日本テレコム (株) の林 秀樹 氏には、本研究に的確なご指摘、アドバイスを頂きました。ここに感謝の意を表します。大学院修士課程 岩田研究室の森川 大智 氏、別役 宣奉 氏には、学部時から研究のみならず多くの時を共にし、心身共に支えて頂きました。本当に感謝しています。大学院修士課程 岩田研究室の福永 諭 氏には、修士課程からの同僚として御世話になりました。ありがとうございました。大学院修士課程 岩田研究室の中村 勲二 氏には、映像グループとして共に議論でき、また学部生の指導を積極的に手伝って頂きました。本当にありがとうございました。大学院修士課程 岩田研究室の三宮 秀次 氏には、上記共同研究、並びに私生活において大変御世話になりました。至極感謝しています。大学院修士課程 岩田研究室の小倉 道寛 氏、志摩 浩 氏には、ネットワーク管理、PC 管理など御世話になりました。ありがとうございました。最後になりましたが、岩田研究室学部生の皆さん、並びに関係者の方々に心から御礼を申し上げます。

# 参考文献

- [1] H. Terada, S. Miyata, and M. Iwata, “DDMP’s: self-timed super-pipelined data-driven multimedia processors,” *Proc. of IEEE*, 87(2), 282-296 (1999).
- [2] 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,” 安田, 藤原 監訳, 共立出版, pp415 (1999).
- [3] Charilaos Christophoulos, Athanassios Skodras, Touradj Edraimi, “THE JPEG2000 STILL IMAGE CODING SYSTEM : AN OVERVIEW,” *Proc. of IEEE*, 46(4), 1103-1127 (2000).
- [4] 三木弼一, “MPEG-4 のすべて,” 工業調査会 (1999).
- [5] “視覚情報ハンドブック,” 日本視覚学会, 191-243 (2000).
- [6] E.J.Delp and O.R.Mitchell, “Image Compression using Block Truncation Coding,” *IEEE Trans. on Comm.*, 27(9), 1335-1342 (1979).
- [7] 高木 幹雄, 下田 陽久, “画像解析ハンドブック,” 東京大学出版会, (1991).
- [8] 青木 茂樹, 大西 正輝, 小島 篤博, 福永 邦雄, “HMM による行動パターンの認識,” 電子情報通信学会論文誌 (D-II), vol.J85-D-II, no.7, pp.1265-1270, (2002).
- [9] 大久保修司, 桐島俊之, 佐藤宏介, 千原國宏, “三次元動作認識における最適視点位置の選択,” 電気関係学会関西支部連合大会講演論文集, p.310 (1998).
- [10] 橋本 正和, 中村 勲二, 岩田 誠, 大森 洋一, “動き部分が小さい用途に適した超軽量映像処理方式の提案,” 第 65 回情報処理学会全国大会 (2002-3 発表予定).
- [11] 的場 成浩, 青木 透, 池田 宏明, “PSNR によるビデオ客観評価測定装置と測定結果,” 画像電子学会 VMA 研究会, (2001).
- [12] 池田 宏明, Poudel Bed Prasad, 小町 祐史, “AV システムの品質評価 (QA) の必要性和標準化,” 画像電子学会第 5 回 VMA 研究会, (2000).

# 付録 A

## 画質評価手法

### A.1 映像の客観評価

映像の画質を評価する上で、コンピュータなどの計算によって数値を求める客観評価と、人間が実際に画像を見て判断をする主観評価の二種類が存在する。客観評価は数値として画像を定量的に評価できるため、データとして扱いやすいが、映像を実際に見て評価するのは利用する人間であるため、主観評価として多人数の意見を集計する方式が用いられる場合もある。どちらが最良の方式と断言することはできず、主観客観双方の評価が一致するような評価方式の確立が望まれている。

映像を客観評価する手法として様々な方式が提案されている。その中で最もよく用いられている評価手法として、PSNR(Peak Signal to Noise Ratio) 画像品質誤差がある。また、その他の評価手法として、RMS(Root Mean Square) 平均二乗誤差、コントラスト誤差などがある。ここではこれら画質評価手法について解説する。なお、本研究で利用した画質評価手法は、最も広く用いられている PSNR である [11][12]。

#### A.1.1 PSNR

映像における客観映像評価の一つとして PSNR がある。PSNR は平均ノイズ電力 (MSE) を参照画像と劣化画像の輝度信号の差の二乗平均とし、最大信号電力 (peak-signal) を最大輝度信号の大きさの二乗とする。PSNR は平均ノイズ電力 (MSE) および最大信号電力から求められる。

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE} [dB]$$

$MAX =$  画像の最大値

$$MSE = \frac{\sum (X_i - \hat{X}_i)^2}{M}$$

$M =$  総画素数

$X_i :$  元画素値  $\hat{X}_i :$  変換後の画素値

PSNR を用いることで輝度信号に対する映像品質の客観評価指数が得られる。しかし、最近では色の観点も含めた客観評価の機運が高まっており、PSNR に関しても色の観点を含んだ議論がなされつつある。

そこで、以下に RGB に関する PSNR 値の算出方式を示す。

$$MSE = \frac{\sum (\Delta RGB)^2}{M}$$

$$\Delta RGB = \sqrt{(R - \hat{R})^2 + (G - \hat{G})^2 + (B - \hat{B})^2}$$

本研究においては、上記式を用いて PSNR 値を算出した。

また、PSNR を 3 次元空間へ適用する場合、以下の式を用いる。

$$MSE = \frac{1}{(P_2 - P_1 + 1)(M_2 - M_1 + 1)(N_2 - N_1 + 1) \sum_{p=P_1}^{p=P_2} \sum_{m=M_1}^{m=M_2} \sum_{n=N_1}^{n=N_2} (d(p, m, n) - o(p, m, n))^2}$$

なお、上記式は図 A.1 に示した映像の構成に沿ってある。

RGB の 3 次元空間への適用する式は以下である。

$$MSE_{RGB} = \frac{1}{(P_2 - P_1 + 1)(M_2 - M_1 + 1)(N_2 - N_1 + 1) \sum_{p=P_1}^{p=P_2} \sum_{m=M_1}^{m=M_2} \sum_{n=N_1}^{n=N_2} (\Delta RGB)^2}$$

さらに、YCrCb の 3 色について PSNR を算出する手法もある。



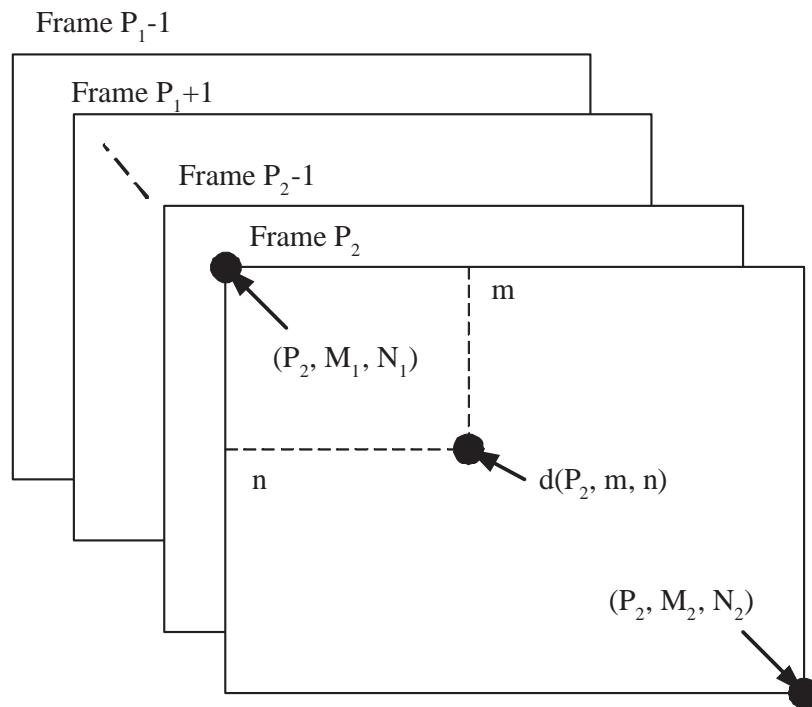


図 A.1 映像の構成

既存の演算手法では、輝度値のみを対象としていたが、伝送路において色彩情報の劣化が激しい帯域を用いた場合、輝度値の変化のみでは評価が困難といわれている。そのような場合、三次元空間での PSNR による評価が重要となってくる。

しかし、人間の色覚は、輝度の変化には敏感で、色の変化には鈍感であるという特性を持つ。この錯覚を利用した映像圧縮処理方式として、JPEG などがあげられ、カラー間引きや DCT、量子化の処理で実現している。そのため、一概に三次元を対象とした PSNR の評価が、有効な評価とはいえない。

### A.1.2 RMS

平均二乗誤差とは誤差の二乗 = 誤差の大きさとして、その平均をとったものである。画質評価においてもこの手法が用いられることがある。

式を以下に示す。

$$RMS = \sqrt{\frac{\sum (X_i - \hat{X}_i)^2}{\sum X_i^2}} \times 100[\%]$$

### A.1.3 コントラスト誤差

コントラストとは画像上の最も明るい部分と暗い部分の差のことであり、その差分の誤差を算出することで画質を評価する手法である。この誤差が小さいほど画質がよく、主に、画像の境界 (エッジ) の保存率を測るための手法となっている。

式を以下に示す。

$$\text{コントラスト誤差} = \frac{\text{ばらつき (圧縮画像)} - \text{ばらつき (元画像)}}{\text{ばらつき}} \times 100[\%]$$

$$\text{ばらつき} = \sqrt{\frac{\sum (X_i - \bar{X})^2}{M}}$$

$$\bar{X} = \frac{1}{M} \sum_{i=1}^M X_i$$

これまでに述べた客観評価手法は、画質評価としてよく用いられるが、最適な評価手法というものは無い。評価したい用途に応じて評価手法を切り替えて利用するのが、現在の主流となっている。