

平成17年度 修士論文

JPEG2000を用いたHDTV低遅延符号化
に関する研究

Low Delay HDTV Coding Using JPEG2000

指導教授 渡辺 裕 教授

2006年2月

早稲田大学大学院 国際情報通信研究科
国際情報通信学専攻

4204A133-0

佐 野 雄 磨

Yuma SANO

目次

| | | |
|------------|-----------------------------|-----------|
| 第1章 | 序論 | 1 |
| 1.1 | 研究の背景 | 1 |
| 1.1.1 | デジタルテレビ放送 | 1 |
| 1.1.2 | HDTVの符号化 | 2 |
| 1.2 | 本研究の目的 | 3 |
| 1.3 | 本論文の構成 | 4 |
| 第2章 | JPEG2000 符号化方式の概要 | 5 |
| 2.1 | はじめに | 5 |
| 2.2 | JPEG2000 の概要 | 5 |
| 2.3 | JPEG2000 の符号化アルゴリズム | 7 |
| 2.3.1 | DC レベル変換 | 7 |
| 2.3.2 | コンポーネント変換 | 8 |
| 2.3.3 | Wavelet 変換 | 8 |
| 2.3.4 | 量子化 | 12 |
| 2.3.5 | EBCOT | 12 |
| 2.3.6 | 符号ストリームの作成 | 14 |
| 2.3.7 | プログレッションの順序 | 14 |
| 2.4 | タイルとプレシネクト | 15 |
| 2.4.1 | タイル構造 | 15 |
| 2.4.2 | タイル分割による不連続性 | 15 |
| 2.4.3 | プレシネクト分割 | 16 |
| 2.5 | Motion JPEG2000 | 17 |
| 2.5.1 | Motion JPEG2000 の特徴 | 17 |
| 2.5.2 | Motion JPEG2000 の符号化方式 | 19 |
| 2.5.3 | Motion JPEG2000 のファイルフォーマット | 21 |
| 2.6 | おわりに | 29 |
| 第3章 | 分割による低遅延符号化方式 | 31 |
| 3.1 | はじめに | 31 |
| 3.2 | 分割符号化の概念 | 31 |
| 3.3 | JPEG2000 タイル分割の符号化遅延低減への応用 | 32 |

| | | |
|------------|----------------------------|-----------|
| 3.3.1 | R-D(Rate-Distortion) 最適化 | 32 |
| 3.3.2 | 想定環境 | 33 |
| 3.4 | 予備実験 | 34 |
| 3.5 | 提案レート制御方式 | 37 |
| 3.5.1 | 量子化誤差分布の制御 | 37 |
| 3.6 | 性能評価 | 39 |
| 3.6.1 | 符号化条件 | 39 |
| 3.6.2 | 符号化性能 | 41 |
| 3.6.3 | コーデック遅延量 | 41 |
| 3.7 | プレシントに基づく低遅延符号化への応用 | 45 |
| 3.7.1 | プレシント分割 | 45 |
| 3.7.2 | 提案レート制御手法の適用 | 47 |
| 3.8 | おわりに | 50 |
| 第4章 | 品質変動低減のための提案方式 | 51 |
| 4.1 | はじめに | 51 |
| 4.2 | JPEG2000 を用いた動画像符号化とその従来手法 | 51 |
| 4.2.1 | JPEG2000 による動画像符号化の問題点 | 51 |
| 4.2.2 | CBR と VBR | 52 |
| 4.2.3 | 従来手法 | 52 |
| 4.3 | 提案方式 | 54 |
| 4.3.1 | 想定環境 | 54 |
| 4.3.2 | フィールド間の品質変動の低減 | 54 |
| 4.3.3 | R-Q 曲線の導出 | 56 |
| 4.3.4 | R-Q 曲線の予測 | 57 |
| 4.3.5 | R-Q 曲線の予測に基づくビット配分 | 58 |
| 4.3.6 | 品質の変動幅の制御 | 58 |
| 4.3.7 | バッファによる符号量制御 | 59 |
| 4.3.8 | バッファサイズと遅延 | 61 |
| 4.4 | 性能評価 | 61 |
| 4.5 | おわりに | 66 |
| 第5章 | 結論 | 67 |
| 5.1 | 総括 | 67 |
| 5.2 | 今後の課題 | 68 |
| | 謝辞 | 69 |
| | 参考文献 | 70 |

| | |
|--------------------|----|
| 付録 A バッファリング遅延量の導出 | 73 |
| 図一覧 | 73 |
| 表一覧 | 76 |
| 研究業績 | 77 |

第1章

序論

1.1 研究の背景

1.1.1 デジタルテレビ放送

近年、カメラやディスプレイの高機能化、符号化技術の進歩、ネットワークの広帯域化など多くの技術の発達を受け、益々多くのマルチメディアコンテンツが流通し身近な存在となっている。その中で、とりわけ大きな変化としてテレビ放送のデジタル化があげられる。テレビジョンは、20世紀前半に登場して以来、人々の娯楽として、また情報源として生活に欠かすことのできないものであった。これまでのテレビ放送はアナログ方式であり、送ることのできる情報量も限られていた。これに対し、映像の高精細化と多チャンネル化を目指し、テレビジョンのデジタル化が研究されてきた。

その結果、2000年12月にBSデジタル放送が開始され、さらに2003年12月には関東圏・中京圏及び近畿圏で地上デジタル放送が開始された。今後、全国でデジタル放送が開始される予定である⁽¹⁾。デジタル放送によって、HDTV(High Definition Television)による高画質放送、データ放送などの付加サービス、移動体や携帯端末向けサービスなどの多チャンネル放送が可能になる⁽²⁾。最近では、HDTV対応のプラズマテレビや薄型液晶が増えている。

従来のテレビ方式として、日本や北米ではNTSC方式が、欧州ではPAL方式が採用されてきた。NTSC方式が走査線525本であるのに対して、HDTVでは1125本から1250本に増加した。さらに、アスペクト比を従来の4:3から、人の視界に近い16:9へ変更した。HDTVは、走査線が倍増したことと、アスペクト比の広がったことによって、高解像度と高臨場感が実現される。参考までに、主な映像フォーマットの種類を表1-1に示す。

これら背景を受けて、近年標準TVのフォーマットがHDTVに移行しつつあり、TV局ではHDTV解像度の映像素材を扱うようになりつつある⁽³⁾。TV局で作成された映像や、現場で撮影された映像は、それぞれ各中継所へ伝送される。ここで、HDTVを伝送、蓄積する際に重要な技術が符号化技術である。HDTVの情報量は現行テレビフォーマットの約6倍の1Gbit/sとなる。したがって、効率的に伝送、蓄積を行う場合ためには、圧縮符号化が不可欠である。例えば、放送局外から映像素材を中継所へ伝送するためのFPU(Field Pick-up Unit)やワイヤレスカメラに代表される無線素材伝送設備では、伝送レートの制約から一般

表 1-1: 主な映像フォーマット

| Format | Resolution | Aspect Ratio |
|-----------|------------|--------------|
| HDTV 1080 | 1920x1080 | 16:9 |
| HDTV 720 | 1280x720 | 16:9 |
| SDTV NTSC | 720x480 | 4:3 |
| SDTV PAL | 720x576 | 4:3 |
| SIF NTSC | 352x240 | 4:3 |
| SIF PAL | 352x288 | 4:3 |
| CIF | 352x288 | 4:3 |
| QCIF | 176x144 | 4:3 |



図 1-1: HDTV の無線伝送

に圧縮伝送を伴う。特に、リアルタイム性を必要とするライブによる現場中継では運用上の問題から、かけ合いや素材切替（スイッチング）などにおける運用性を改善するために、伝送遅延の低減が強く求められている。そこで、HDTV を低遅延で符号化し伝送できるような符号化方式が必要となる。

1.1.2 HDTV の符号化

HDTV を圧縮する際に、対象とするアプリケーションによって、圧縮するビットレートは異なる。HDTV は非圧縮では 1Gbps 必要だが、素材伝送では 100Mbps、TV における二次分配では 15 ~ 20Mbps のビットレートが必要であり、ネットワークを介して配信する際は 10Mbps で十分である⁽⁴⁾。

現在、映像の符号化方式として最も普及しているのが MPEG-2 である⁽⁵⁾。MPEG-2 は ISO/IEC JTC1/SC29/WG11(Moving Picture Experts Group) で標準化され、DVD の蓄積やデジタル放送など、さまざまな分野で標準圧縮方式として用いられている。MPEG-2 は、DCT 変換と動き補償を用いたハイブリッド符号化であり、符号化効率の点で優れた符号化方式である。また、最新の圧縮符号化技術として、MPEG4 Part10 H.264/AVC があげ

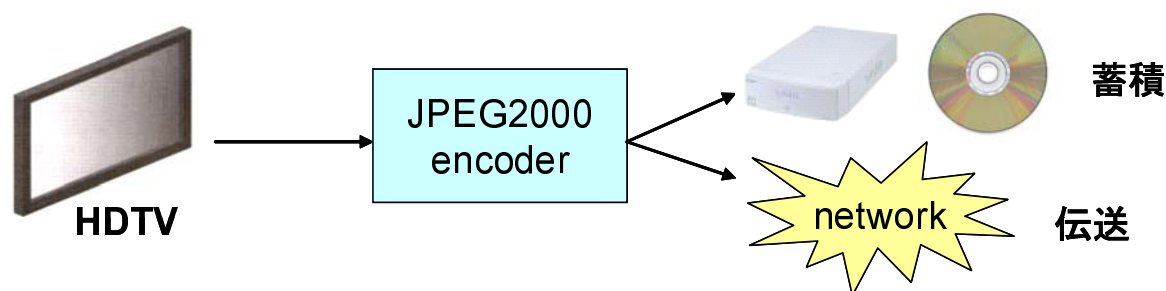


図 1-2: JPEG2000 による HDTV 符号化

られる⁽⁶⁾。H.264/AVC は、MPEG-2 と比較しての二倍以上の圧縮率を得る符号化方式として現在最も注目されている符号化方式である。一方で、本研究で対象としている HDTV レベルの高解像度の映像に対しては、DCT を用いた MPEG-2 や H.264/AVC よりも Wavelet 変換を用いた符号化方式である JPEG2000⁽⁷⁾ が有効であることが報告されている⁽⁸⁾。また、MPEG-2 は双方向の動き補償を用いているために、符号化の際にフレームの並び替えやバッファリングによる大きな符号化遅延が生じる。これに対し、JPEG2000 は、動き補償を用いないフレーム内符号化方式であり、フレームの並び替えやバッファリングによる遅延が生じない。さらに、JPEG2000 は品質スケーラビリティ、解像度スケーラビリティ、空間スケーラビリティを持ち、ワンソースマルチユースで受信側の環境に応じた映像の配信が可能である。したがって、動画像の圧縮符号化において、JPEG2000 によるフレーム内符号化技術を採用することにより、高い符号化効率とスケーラビリティを維持しつつ、動き補償と DCT によるハイブリッド符号化と比較してはるかに少ないコーデック遅延の達成が期待できる。そのほかにも、JPEG2000 はフレーム内符号化であるために、伝送の際のエラー耐性が強いことや、映像の編集に適しているという長所を持つ。これらの特徴を受け、近年 HDTV の符号化方式として JPEG2000 が注目されており、既にデジタルシネマの符号化フォーマットとして採用例がある⁽⁹⁾。

以上の背景より、JPEG2000 エンコーダは高画質および低遅延という特徴を活かし、現場中継を典型例とするリアルタイム映像伝送アプリケーションの運用性を大きく改善するものと期待される。

JPEG2000 を代表とするイントラ符号化を動画像符号化に用いることで、符号化遅延が低減できる一方で、問題点も存在する。JPEG2000 符号化を単純に動画像へ適用すると、主にビット配分の弊害に起因した品質の劣化の発生が懸念される。イントラ符号化を動画像符号化に用いる際には、この品質劣化の問題を解決する必要がある。

1.2 本研究の目的

本論文では、標準 TV のフォーマットが HDTV に移行しつつあることを受け、典型的な映像アプリケーションとして 1080i HDTV を想定し、高画質・低遅延伝送を実現可能な最

適符号化方式の提案を行う．品質を維持したままで，符号化の遅延を低減することを目的とする．

伝送遅延は，限りなく小さいことが理想的であるが，10 フレーム (300msec) 以下に収まれば，人間の対話上ストレスはないと言われている．さらに，放送局用途では，伝送遅延が1フレームに収まれば運用上の支障は少ないと言われている．既に提供されている無線素材伝送設備において，圧縮符号化方式として MPEG-2 を採用しているものが大半を占めており，符号化アルゴリズムの原理上，300msec を上回るコーデック遅延が強いられている状況にある．そこで，フレーム内符号化である JPEG2000 を用いて符号化することによって，フレームの並び替えなどによる遅延を解消する．さらに，符号化遅延の大きな要因として入力画像の読み込みがあげられる．これに対して，本方式では入力画像を分割することによって，入力待ち時間とバッファリング時間，ひいてはコーデック遅延を低減する．その際に，入力画像の分割に起因して発生する境界の不連続性や，不適切なビット配分による品質の劣化を防ぐような符号量制御方式を提案する．さらに，JPEG2000 のようなフレーム内符号化を用いる符号化方式を動画像符号化に適用する際に生じる問題点も合わせて解決する．

1.3 本論文の構成

以下に本章以降の構成を示す．

第2章 本研究で用いる符号化方式である JPEG2000 について説明する．JPEG2000 の概要と符号化処理のアルゴリズムとその技術について述べる．特に，分割符号化における分割単位であるタイルとプレシントについて説明する．さらに，JPEG2000 の動画像フォーマットである Motion JPEG2000 の概要を述べる．

第3章 JPEG2000 のタイルそしてプレシントを単位として，分割符号化することによって符号化遅延を低減することができることを示す．その際，各分割領域に対するビット配分が品質に大きく影響を与えることを示す．そして動画像として品質を維持するために，Explicit 量子化とポスト量子化を組み合わせる符号量制御方式を提案する．さらに，実験によって提案方式の性能評価を行う．

第4章 まず動画像符号化の際に各フレームに対して適切なビット配分が行われないと，動画像としての品質が劣化することを説明する．さらに，各フレームの品質が変動しないようなビット配分方法を提案する．後続予測によって，フレームを複数バッファリングすることなく低遅延で，当該フレームに割り当てるビット量を決定する．さらに，バッファサイズの制限の下，隣接フレームの量子化ステップの変動に制限をかけることによって，フレーム間の品質の変動を低減する．実験によって，提案方式の性能評価を行う．

第5章 本研究の総括と今後の課題について述べる．

第2章

JPEG2000 符号化方式の概要

2.1 はじめに

本章では、HDTV の符号化方式として JPEG2000 をとりあげる。まず JPEG2000 符号化方式の概要とその符号化アルゴリズムとその技術について説明する。次に、本研究で用いる分割領域であるタイルとプレシントについて説明する。さらに、JPEG2000 の動画画像フォーマットである Motion JPEG2000 について説明を行う。

2.2 JPEG2000 の概要

JPEG2000 は、2000 年 12 月に ISO/IEC JTC1/SC29/WG1 によって国際標準化された静止画像符号化の国際標準である。静止画符号化に関しては、すでに JPEG が広く普及しているが、JPEG2000 は JPEG と比較して圧縮効率や、様々な機能の面で優れている。

- 動き補償を用いないイントラ符号化である。
そのため、伝送の際のエラー耐性が強い。
さらに、映像の編集に適している。
- MPEG-2、MPEG-4 で用いられている DCT 変換ではなく DWT 変換を用いている。
そのため、高解像度の画像に対して高品質を保つことができる。
- ポスト量子化によって、精密な符号量制御が可能である。
- DWT によるサブバンド分割による空間スケーラビリティ、EBCOT 処理による品質スケーラビリティ、Precinct による位置スケーラビリティなど、高いスケーラビリティを持ち、ワンソースマルチユースを実現する。
- ロスレス符号化が可能。
- 階調表現で RGB がそれぞれ 38bit まで対応可能。

JPEG2000 の各パートとその役割 JPEG2000 の要求仕様は多くのものがある。これらすべてを基本方式に含めると、基本方式がとても複雑になり、使われない標準方式となるおそ

れがある．そこで，基本部分は JPEG2000 の特長を生かしつつ，かつ比較的簡単な構成で実現できる技術のみをすることになった．基本方式は JPEG2000 Part-1 と称され，大きな特徴としては以下の 2 点があげられる．

- JPEG2000 準拠のデコーダが必ず満たす条件を規定した
- 搭載される技術は特許的にフリー宣言されているもの

前者は JPEG ベースライン方式と同様に，JPEG2000 準拠とされるデコーダは基本方式で規定されているすべての技術をサポートする必要がある点を示している．なお，エンコーダについては基本方式のすべてをサポートする必要はない．また，後者は JPEG2000 を実装する上で非常に大きな意味を持っている．

基本方式でサポートされない技術（例えば特許的に有償である技術）は拡張方式として JPEG2000 Part-2 と称する別勧告となった．

このほかに JPEG2000 では，動画を静止画の系列として扱う Motion JPEG2000(Part-3)，相互接続試験用のテスト方法を記述した勧告 (Part-4)，参照用のソフトウェア (Part-5)，絵文字混在画像を対象とした符号フォーマット (Part-6)，セキュリティ関連の JPSEC(Part-8)，API やプロトコルを扱う JPIP(Part-9)，三次元データを扱う JP3D(Part-10)，ワイヤレス通信における JPEG2000 の適用を扱う (Part-11)，ファイルフォーマットを扱う ISO Base Media File Format(Part-12)，特許料フリーを目指したエントリーレベルの JPEG2000 エンコーダーの定義 (Part-13) がある．なお，Part-7 が欠番となっているが，もともと Part-7 は「機器設計のためのガイドライン」ということで，JPEG2000 機器が最低限持つべき機能を定義したものであるとして存在していた．ISO の標準には規定 (normative) と参考 (informative) という部分があり，前者は標準に準拠するために守らなければならない条件であるのに対して，後者は標準に準拠するための参考情報であって必ずしも守る必要はない．Part-7 は参考情報という役割として審議されていたが，審議を進めていくうちに，参考ではなく規定とすべきであることが認められ，基本方式への追随という形で基本方式に含まれることになった．欠番となったのは既に Part-8 以降が予約されていたからである．

JPEG2000 の各パートについて表 2-1 に示す⁽⁷⁾．

表 2-1: JPEG2000 の各パート

| | |
|---------|--|
| Part-1 | Core coding system (intended as royalty and license-fee free - NB NOT patent-free) |
| Part-2 | Extensions (adds more features and sophistication to the core) |
| Part-3 | Motion JPEG 2000 |
| Part-4 | Conformance |
| Part-5 | Reference software (Java and C implementations are available) |
| Part-6 | Compound image file format (document imaging, for pre-press and fax-like applications, etc.) |
| Part-7 | has been abandoned |
| Part-8 | JPSEC (security aspects) |
| Part-9 | JPIP (interactive protocols and API) |
| Part-10 | JP3D (volumetric imaging) |
| Part-11 | JPWL (wireless applications) |
| Part-12 | ISO Base Media File Format (common with MPEG-4) |
| Part-13 | Entry Level JPEG 2000 Encoder |

2.3 JPEG2000 の符号化アルゴリズム

JPEG2000 の符号化アルゴリズムを図 2-1 に示す．それぞれの処理について説明する．

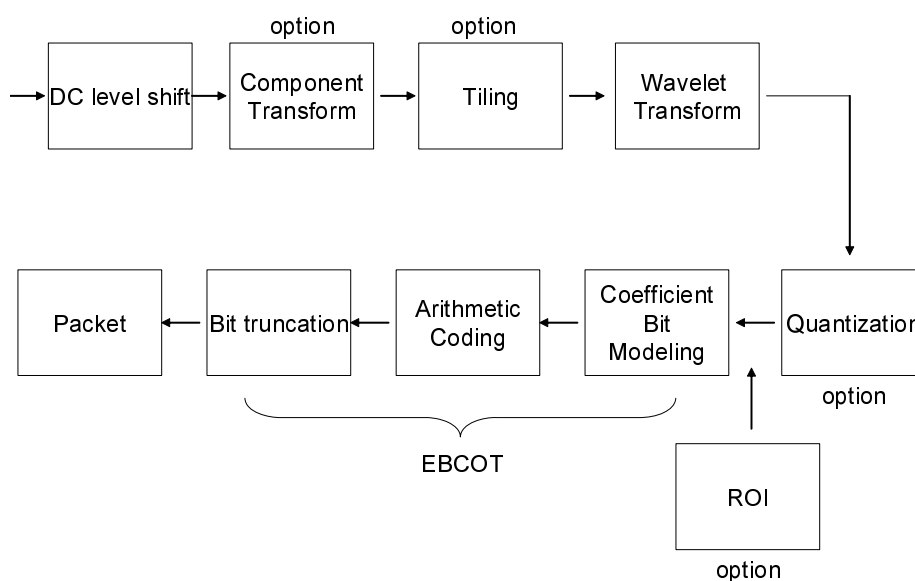


図 2-1: JPEG2000 の符号化アルゴリズム

2.3.1 DC レベル変換

入力信号が正の値である場合には，各信号値から信号のダイナミックレンジの半分を減算するレベルシフトを行う．逆に復号時には各信号値に信号のダイナミックレンジの半分を加算するレベルシフトを行う．なお，この DC レベル変換はオプションである．

2.3.2 コンポーネント変換

入力画像が RGB の 3 つのコンポーネントから構成される場合に、線形変換によって輝度と色差を表す YCbCr の 3 成分に変換する。コンポーネント変換には可逆と非可逆がある。可逆変換は RCT (Reversible multiple component transformation) と呼ばれ、非可逆変換は ICT (Irreversible multiple component transformation) と呼ばれる。以下に、RCT と ICT の順変換と逆変換の式をそれぞれ示す。

RCT

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} \frac{R+2G+B}{4} \\ R - G \\ B - G \end{bmatrix}$$

$$\begin{aligned} G &= Y - \left(\frac{Cb - Cr}{4} \right) \\ R &= Cb + G \\ B &= Cr + G \end{aligned}$$

ICT

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.229 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.0 & 0 & 1.402 \\ 1.0 & -0.34413 & -0.71414 \\ 0.5 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix}$$

2.3.3 Wavelet 変換

Wavelet 変換の概要

JPEG2000 では、入力信号を周波数領域に変換する方法として離散 Wavelet 変換 (Discrete Wavelet Transform : DWT) を用いている。Wavelet 変換は、JPEG で用いられている DCT 変換同様、周波数成分の偏りを利用して冗長度を削減する方法である。Wavelet 変換では周波数軸を等しい帯域幅で均等分割するサブバンド符号化に基づいている。図 2-2 に 2 分割フィルタによる信号の分析・合成システムを示す。

画像信号は直流成分を含む低域周波数に主要な信号成分が集中しているため、低周波数成分に対してのみ分割フィルタを適用していく。この分割方法は、オクターブ分解、ツリー構造分解または Mallat 分解と呼ばれ、この方法に基づくサブバンド分割の回数を表すパラメータは、分解レベルと呼ばれる。変換は、2 次元変換であるが、これは 1 次元フィルタを

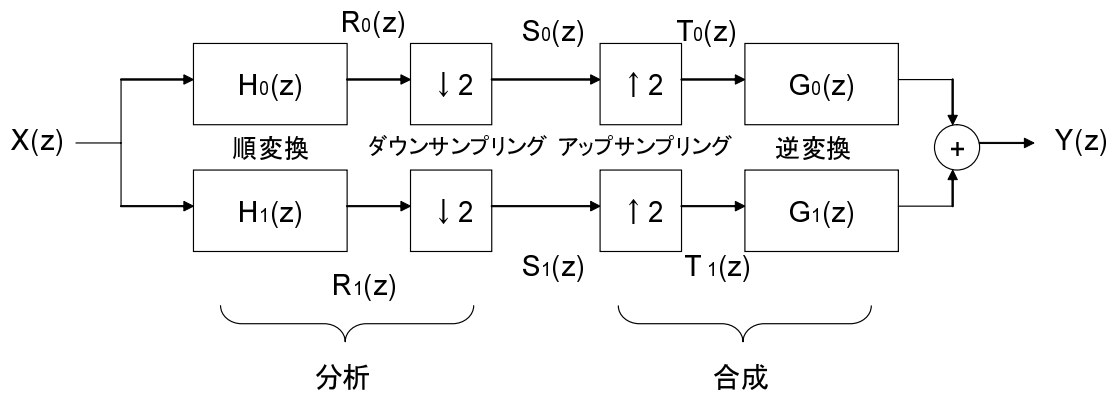


図 2-2: 2 分割フィルタによる信号の分析・合成システム

順変換では垂直方向，水平方向の順に適用することで2次元変換係数を算出する．図 2-3 に分解レベルが 2 の場合の 2 次元空間に対するサブバンド分解の例を示す．

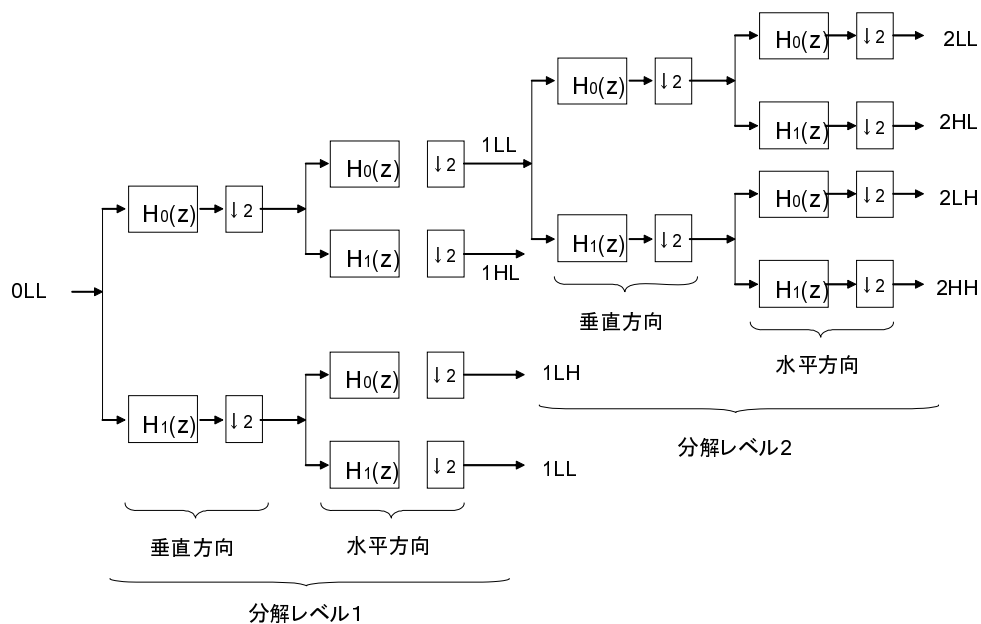


図 2-3: 分解レベルが 2 の場合の 2 次元空間に対するサブバンド分解

サブバンド分割によって，信号は次の 4 つのサブバンドに分けられる．

- サブバンド LL : 垂直方向と水平方向ともにローパスフィルタリングを行い，2:1 ダウンサンプリングした信号．
- サブバンド HL : 垂直方向にローパスフィルタリング，水平方向にハイパスフィルタリングを行い，2:1 ダウンサンプリングした信号．

- サブバンド LH : 垂直方向にハイパスフィルタリング, 水平方向にローパスフィルタリングを行い, 2:1 ダウンサンプリングした信号 .
- サブバンド HH : 垂直方向と水平方向ともにハイパスフィルタリングを行い, 2:1 ダウンサンプリングした信号 .

図 2-4 に各サブバンドの配置を示す .

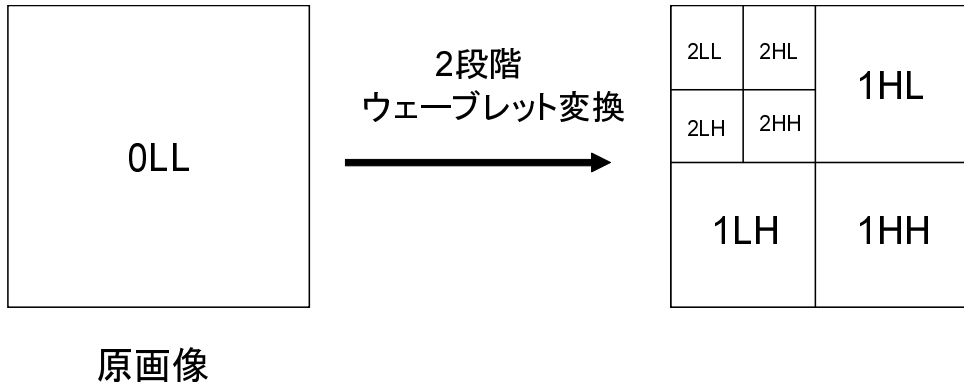


図 2-4: 各サブバンドの配置

さらに例として, Lena 画像を 3 段階で Wavelet 変換したものを図 2-5 に示す .

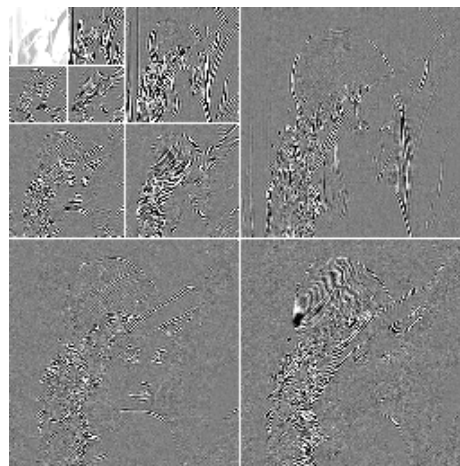


図 2-5: Lena 画像の 3 段階 Wavelet 変換

可逆 Wavelet 変換と非可逆 Wavelet 変換

Wavelet 変換は変換係数が整数で構成される整数型 DWT と実数で構成される実数型 DWT の二つに分類することができる . 整数型 DWT は可逆変換が可能で, 回路構成も小さくできる . 一方, 実数型 DWT は整数型 DWT と比較して同圧縮率において画質がよいが, 可逆変

換はできない。JPEG2000Part1では整数型フィルタとして 5×3 フィルタ, 実数型フィルタとして 9×7 フィルタが定義されている。表2-2~2-5に 5×3 可逆フィルタと 9×7 非可逆フィルタのフィルタ係数を示す。

表 2-2: 5×3 フィルタの係数 (順変換)

| i | LPF $h_0(i)$ | HPF $h_1(i)$ |
|---------|--------------|--------------|
| 0 | 1 | 6/8 |
| ± 1 | -1/2 | 2/8 |
| ± 2 | | -1/8 |

表 2-3: 5×3 フィルタの係数 (逆変換)

| i | LPF $h_0(i)$ | HPF $h_1(i)$ |
|---------|--------------|--------------|
| 0 | 1 | 1/2 |
| ± 1 | -2/8 | 1/2 |
| ± 2 | -1/8 | |

表 2-4: 9×7 フィルタのフィルタ係数 (順変換)

| i | LPF $h_0(i)$ | HPF $h_1(i)$ |
|---------|----------------------|----------------------|
| 0 | 0.6029490182363579 | 1.115087052456994 |
| ± 1 | 0.2668641184428723 | -0.5912717631142470 |
| ± 2 | -0.07822326652898785 | 0.05754352622849957 |
| ± 3 | 0.09127176311424948 | -0.01686411844287495 |
| ± 4 | | 0.02674875741080976 |

表 2-5: 9×7 フィルタのフィルタ係数 (逆変換)

| i | LPF $h_0(i)$ | HPF $h_1(i)$ |
|---------|----------------------|----------------------|
| 0 | 1.115087052456994 | 0.6029490182363579 |
| ± 1 | 0.5912717631142470 | -0.2668641184428723 |
| ± 2 | -0.05754352622849957 | -0.07822326652898785 |
| ± 3 | -0.09127176311424948 | 0.01686411844287495 |
| ± 4 | 0.02674875741080976 | |

具体的に， 5×3 可逆フィルタの分析側の伝達関数は，

$$H_0(z) = (-z^{-2} + 2z^{-1} + 6 + 2z^1 - z^2)/8$$

$$H_1(z) = (-z^{-1} + 2 - z^1)/2$$

また合成側の伝達関数は，

$$F_0(z) = -H_1(-z)$$

$$F_1(z) = H_0(-z)$$

と表すことができる．また 9×7 非可逆フィルタの分析側の伝達関数は，表 2-4 のフィルタ係数 $h_0(i)$ ， $h_1(i)$ を用いて以下のように表すことができる．

$$H_0(z) = \sum_{i=-4}^4 h_0(i) z^i$$

$$H_1(z) = \sum_{i=-3}^3 h_1(i) z^i$$

合成側の伝達関数に関しては， 5×3 可逆フィルタと同様である．

2.3.4 量子化

係数のダイナミックレンジを削減する．JPEG2000Part1 の基本方式では，スカラー量子化とポスト量子化の2つの量子化方法が定義されている．スカラー量子化には，各解像度レベルにおける全てのサブバンドの量子化を規定する方式と，最下位の解像度レベルにおけるLLサブバンドのみを規定して，残りのサブバンドはあらかじめ定められている式を用いて規定する方式の2種類がある．前者を明示的な量子化 (explicit quantization)，後者を暗黙的な量子化 (implicit quantization) と呼ぶ．なお，可逆変換時には量子化は行わない．

一方，ポスト量子化では，完成した符号列の下位ビットプレーンを切り捨てることによって量子化を行う．ポスト量子化のメリットは，符号量の制御を1パスで実現可能であることだ．しかし，ポスト量子化の量子化ステップ幅は，2のべき乗となるため，量子化ステップ幅が大きくなった場合の再生画像の画質が細かく制御できないという問題点がある．

2.3.5 EBCOT

JPEG2000 では，エントロピー符号化としてEBCOT(Embedded Block Coding with Optimized Truncation) といわれるブロックベースのビットプレーン符号化が用いられる．EBCOT 処理の段階を以下に示す⁽¹⁰⁾⁽¹¹⁾．

- コードブロック分割
- 係数ビットモデリング
- 算術符号化

それぞれの過程について説明する。

コードブロック分解

まず、入力画像の各サブバンドを同一サイズのコードブロックと呼ばれるブロック単位に分割する。EBCOT ではコードブロックを単位として、以降の処理を独立に行う。コードブロックのサイズは面積が 16 よりも大きく、4096 を超えない範囲で自由に決めることができる。

係数ビットモデリング

分割されたコードブロックの符号化対象となる Wavelet 係数は、ビットプレーンに変換される。さらにそれぞれのビットプレーン内の各ビットに対しては、significance propagation pass(有意な係数が周囲にある有意でない係数の符号化パス)、magnitude refinement pass(有意な係数の符号化パス)、cleanup pass(残りの係数情報の符号化パス) の 3 つの符号パスに分解される。ビットプレーン分解と符号化パスへの分解の様子を図 2-6 に示す。

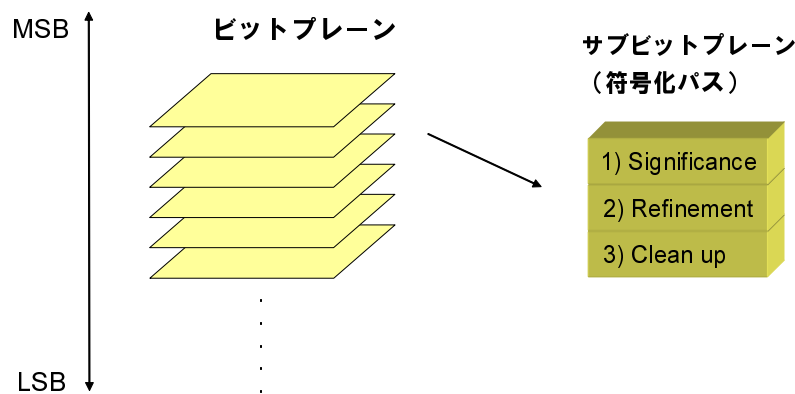


図 2-6: ビットプレーン分解と符号化パスへの分解

算術符号化

係数ビットモデリングで得られたビットプレーン分解された各サブバンドの係数はエントロピー符号化によってさらに圧縮されビットストリームが生成される。このエントロピー符号化には、MQ-Coder と呼ばれる符号化方式を用いる。係数モデリングで作られた符号パスを最小単位として算術符号化が行われる。

2.3.6 符号ストリームの作成

算術符号化によってコードブロック単位で圧縮されたデータは、1個以上のレイヤに分配される。レイヤは圧縮データを画像品質への寄与度に応じて階層化したものである。レイヤ分割された情報を順番に復号していけば、段階的に画像品質が改善される。各コードブロックの対応するレイヤ情報は、さらにパケットと呼ばれるバイト単位のセグメント形式で符号ストリーム内に配置される。図2-7にコードブロックのデータをレイヤへ分割する一例を示す。

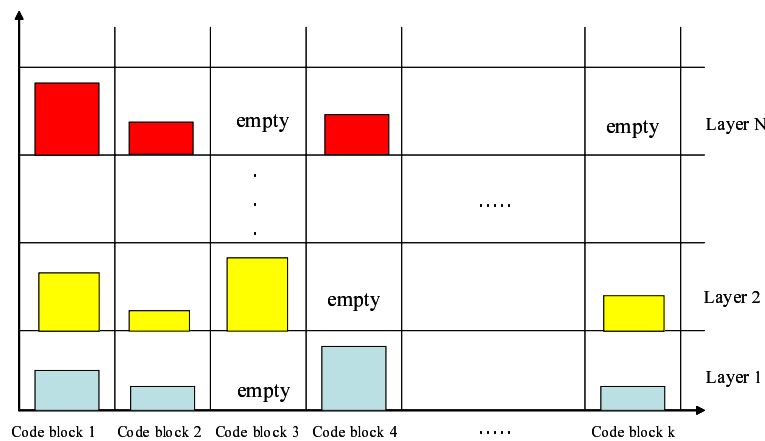


図 2-7: コードブロックデータのレイヤへの分割の例

2.3.7 プログレッションの順序

上までで作成されたパケットは、特定のレイヤ、コンポーネント、解像度レベル、プレジント(位置)からの画像再生への寄与情報を担っている。画像再生は、符号ストリームの最初から順番にパケットを復号して行われるため、パケットを符号ストリーム内で順番に並び替えることによって、品質、解像度、位置、コンポーネントを任意に制御できる。このプログレッションの順序としては、以下の5つのパターンの順序が定められている。

1. LRCP(レイヤ(L), 解像度レベル(R), コンポーネント(C), 位置(P))
2. RLCP(解像度レベル(R), レイヤ(L), コンポーネント(C), 位置(P))
3. RPCL(解像度レベル(R), 位置(P), コンポーネント(C), レイヤ(L))
4. PCRL(位置(P), コンポーネント(C), 解像度レベル(R), レイヤ(L))
5. CPRL(コンポーネント(C), 位置(P), 解像度レベル(R), レイヤ(L))

2.4 タイルとプレシネクト

2.4.1 タイル構造

JPEG2000 では、入力画像を矩形のタイルに分割可能である⁽⁷⁾。分割された各タイルは、タイル境界を越えての画素参照は行わず独立に符号化、復号される。サイズの大きい画像を符号化するには、タイルを用いて並列的に符号化、復号することによって処理を軽減し、遅延を減少することが可能である。しかし一方で、低ビットレート時には、タイルの境界付近は画素値の不連続性が生じるという欠点がある⁽¹²⁾⁽¹³⁾。

2.4.2 タイル分割による不連続性

入力画像から分割されたタイルは、それぞれ独立に符号化される。Wavelet 変換の際に、タイルの境界の係数については、係数を折り返すことによってタイル外部への係数値を得て Wavelet 変換を行う。一方で、タイル分割しない場合は、タイル境界にあたる位置の係数は折り返し処理を行わず、フィルタ長の係数を参照して Wavelet 変換を行う。タイル分割を行うと、接する2つのタイルで折り返し処理によって Wavelet 変換が行われるために、その境界に不連続性が生じ、画質の劣化の原因となる。特に低ビットレート時にその傾向は顕著である。図 2-8 に 0.3bpp 時のタイル境界の不連続性の様子を示す。画像を中央に水平な歪が確認できる。

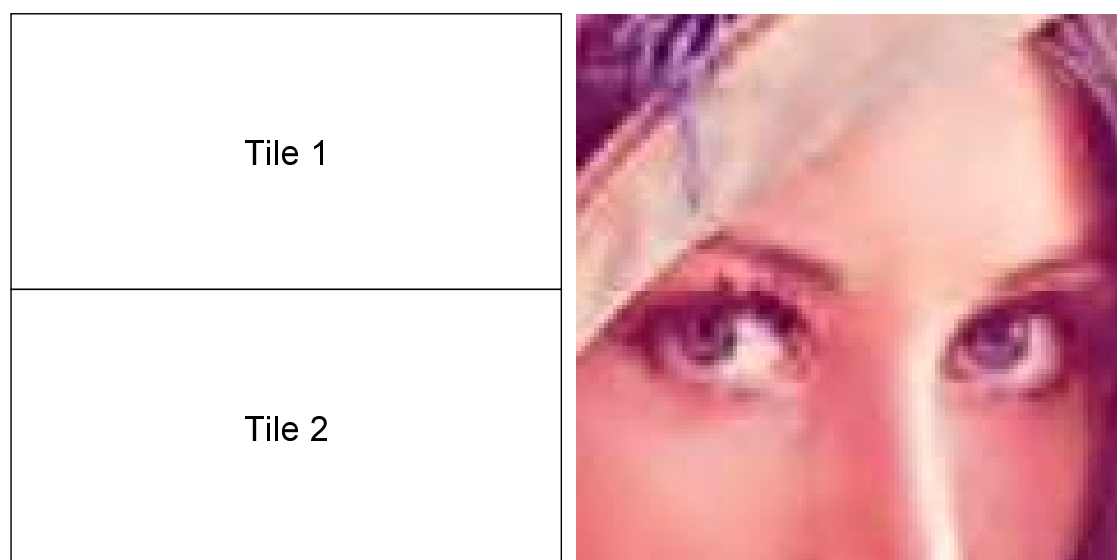


図 2-8: タイルの分割法 (左) とタイル境界の不連続性 (右)

2.4.3 プレシント分割

JPEG2000 の符号化手順の中で、コードブロックは集められてプレシントという領域を形成することが可能である。プレシントは単なるコードブロックの集合であり、プレシントサイズはコードブロックサイズを考慮して設定する必要がある⁽⁷⁾。タイルとは異なり、プレシントを設定することによって符号化や画質には影響を与えることはない⁽¹⁴⁾。また、プレシントの大きさは、2のべき乗である必要があるが、そのサイズは各解像度レベルで異なる値をとることが可能である。また、JPEG2000 では、タイルパートごとにパケットの順序を決めることが可能である。プレシントは、パケット順序を決める単位の一つであり、プレシントごとにパケットを連続的に配置可能である。特定のプレシント情報を得たり、空間的なランダムアクセスなどを行う際に有用である⁽¹⁵⁾⁽¹⁶⁾。以下に、タイルによる分割とプレシントによるの違いを示す。

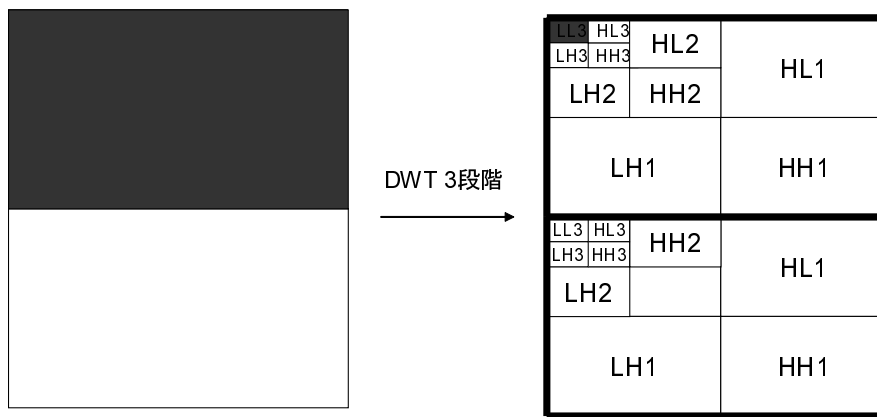


図 2-9: タイル構造のサブバンド分割

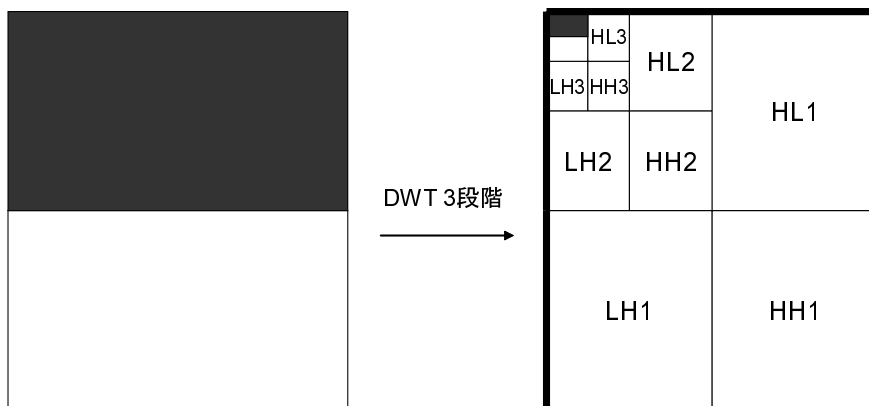


図 2-10: プレシント構造のサブバンド分割

入力画像の上下二等分をタイルとして設定した場合のサブバンド分割の様子を図 2-9 に、

プレシントとして設定した場合のサブバンド分割の様子を図 2-10 に示す。サブバンド分割を 3 段階した後の最低域成分におけるタイルもしくは Precint が図の右側の黒色部分にあたる。タイルはそれぞれのタイル内でサブバンド分割が行われ、プレシントは画像全体でサブバンド分割された後、プレシントとして分割される。

2.5 Motion JPEG2000

JPEG2000 では、動画像フォーマットとして Part3 に Motion JPEG2000 が定められている⁽¹⁷⁾。ここでは、Motion JPEG2000 についての概要を述べる。

2.5.1 Motion JPEG2000 の特徴

ここでは、JPEG2000 の特徴をそのまま継承しながら、さらに動画圧縮のための新機能を備えた Motion JPEG2000 の特徴について述べる⁽¹⁸⁾、⁽¹⁹⁾、⁽²⁰⁾。

編集の容易性

Motion JPEG2000 では、連続する動画像の 1 枚 1 枚の画像を独立に JPEG2000 で符号化する手順を取っている（フレーム内符号化）。これを図 2-11 に示す。

すべての符号化対象が I ピクチャである Motion JPEG2000 (DV も同様) では、ピクチャ単位の符号化ができるので、ピクチャの順番を入れ替えたり、切り出したり、あるいはスロー順方向再生、スロー逆方向再生、シャトルといったビデオレコーダの機能にも好都合である。一方、MPEG-2 では図 2-12 のように、動き補償を用いるため圧縮効率は向上するものの、編集が容易ではないという欠点がある。

低遅延

CODEC の遅延もピクチャ構造に依存し、図 2-12 の構造は余計に数枚のピクチャの遅延が生じる。TV 電話のような双方向通信などの低遅延を要求されるアプリケーションでは、低遅延の I ピクチャの CODEC は非常に都合がよい。

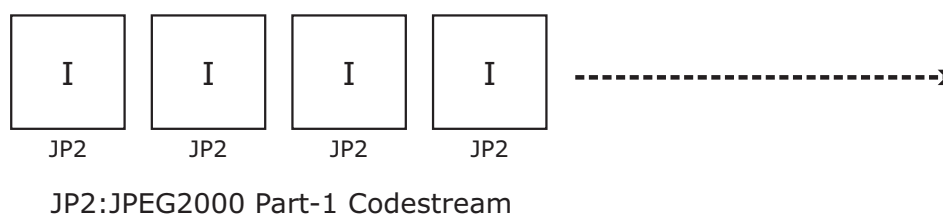


図 2-11: Motion JPEG2000 のピクチャ構造

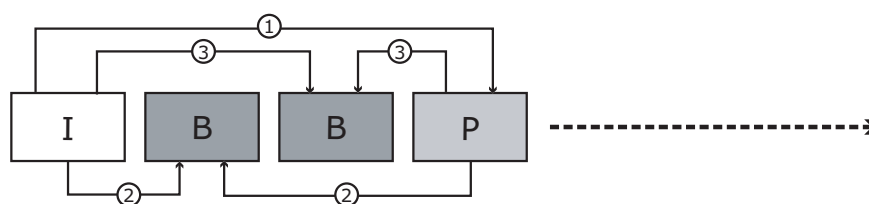


図 2-12: MPEG-2 のピクチャ構造

可逆・非可逆圧縮

JPEG2000 Part-1 準拠である Motion JPEG2000(Part-3) では、Part-1 と同じように可逆 Wavelet フィルタを用いることで可逆圧縮を実現できる。また、可逆圧縮して得られたコードストリームを非可逆のコードストリームにトランスコードすることもできる。これは MPEG などの従来の動画国際標準方式にはないメリットである。

スケーラビリティ

JPEG2000 の持つスケーラビリティ機能をそのまま使えるので、空間解像度レベルのプログレッション、画質レベルのプログレッションのほか、コンポーネントのプログレッションなど、豊富な種類がある。これらの機能は、WWW での高速かつ高能率的なブラウジングや、ビデオ編集機でも利用価値がある。

解像度・ビット精度・フレームレートの自由度

JPEG2000 では解像度の自由度が非常に大きく、規格上では 2^{31} まで、水平・垂直サイズが対応可能になっている。また、ビット精度については、符号1ビット+絶対値38ビットの計39ビット深度まで対応可能である。Motion JPEG2000 の場合には、本来が静止画の連続形式であることから、フレームレートは任意に設定可能である。このように、従来の動画符号化規格には存在しなかった自由度がある。

動画と音声の同期

既存の Motion JPEG の技術では、動画と音声の同期法は規格化されていなかった。一方、DV ではフレーム単位に同期技術が確立されている。Motion JPEG2000 では、“MJ2 file format”によって動画と音声の同期が実現される。MJ2 file format は、Apple 社が開発した QuickTime フォーマットをベースにして開発されたものであり、拡張性が高いというメリットがある。

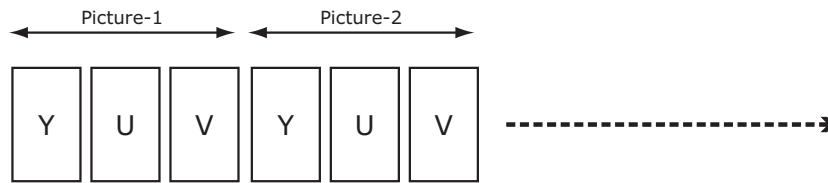


図 2-13: 3 コンポーネントの場合のピクチャ構造

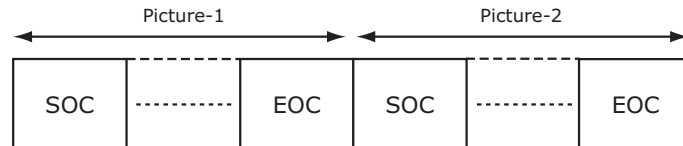


図 2-14: 符号化コードストリーム構造

2.5.2 Motion JPEG2000 の符号化方式

マルチコンポーネント符号化

カラー画像を符号化する際、複数のコンポーネントを符号化対象にしなければならない。カラー画像の基本フォーマットは、色の3原色である RGB であるが、RGB の冗長性を除去したフォーマットとして YUV、または YCbCr が存在する。色変換式は以下の式で与えられる。

$$\begin{aligned}
 Y &= 0.299 \times R + 0.587 \times G + 0.144 \times B \\
 U &= -0.16875 \times R - 0.33126 \times G + 0.5 \times B \\
 V &= 0.5 \times R - 0.41869 \times G + -0.8131 \times B
 \end{aligned}
 \tag{2-1}$$

MPEG や JPEG の場合は、YUV 各コンポーネントのブロックをまとめて符号化処理単位としている。JPEG2000 の場合は、別々に符号化する。したがって、YUV の3コンポーネントの場合には、Y,U,V の順に JPEG2000 の符号化を行い、1つのコードストリームが生成されることになる。

図 2-13 は YUV3 コンポーネントのピクチャ構造を示している。3つのコンポーネントをまとめた符号化コードストリームから、各ピクチャが構成されている。YUV の順番はとくに規定がないが、通常は1つの符号化コードストリームの中に、Y,U,V の符号化データを並べる。

なお、各ピクチャの符号化コードストリームは、いずれも JPEG2000 Part-1 の規格に準拠している。したがって、図 2-14 で図示したように、符号化コードストリームの開始を示す SOC(Start of Codestream) から始まり、EOC(End of Codestream) で終了する。

フレームベース符号化・フィールドベース符号化(インタレース)

現在、動画の表示については、テレビに代表されるインタレース画像と、PC モニタで代表されるプログレッシブ画像が混在している。これら両方の走査型に対応可能な符号化

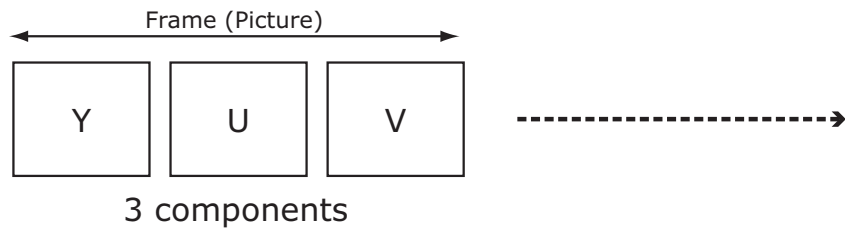


図 2-15: フレームベースの YUV コンポーネントのピクチャ構造

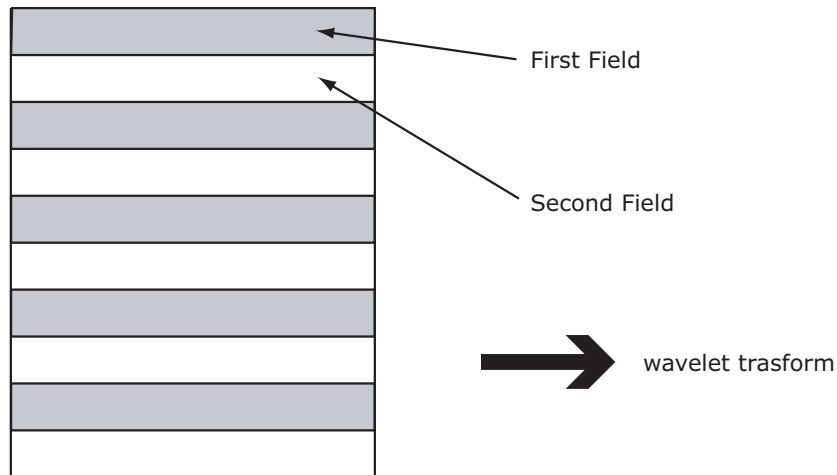


図 2-16: フレームベース符号化時の Wavelet 変換

モードを Motion JPEG2000 の符号化方式は備えている。

図 2-15 はフレームベース符号化モード時の、Y,U,V の 3 コンポーネントを符号化する際のピクチャ構造を図示したものである。1 フレームが 1 ピクチャとなっていることがわかる。一方、図 2-15 はフィールドベース符号化モード時の、Y,U,V の 3 コンポーネントを符号化する際のピクチャ構造を図示したものである。1 フレームが 2 ピクチャから構成されている。

図 2-16 はフレームベース符号化モード時の、JPEG2000 の前段部処理である Wavelet 変換手段について図示したものである。インタレース画像の先頭フィールド、後続フィールドを分離せず、フレーム構造を保ったまま Wavelet 変換処理を施す。

一方、図 2-17, 2-18 は、フィールドベース符号化モード時の Wavelet 変換法について図示したものである。両者の相違点は図 2-17, 2-18 が、preceding_field_first の値が 1 か 0 かの違いであり、これは図 2-17 に示すように、時間的に前に位置するフィールドが先頭フィールドである場合には preceding_field_first=1、逆に時間的にあとに位置するフィールドが先頭フィールドである場合には preceding_field_first=0 と定めている。これは、インタレース画像に両タイプが存在するからである。したがって図 2-18 は、時間的に前に位置するフィールドが先頭フィールドである場合であり、同様にフィールド同士をまとめて Wavelet 変換処理を施す。

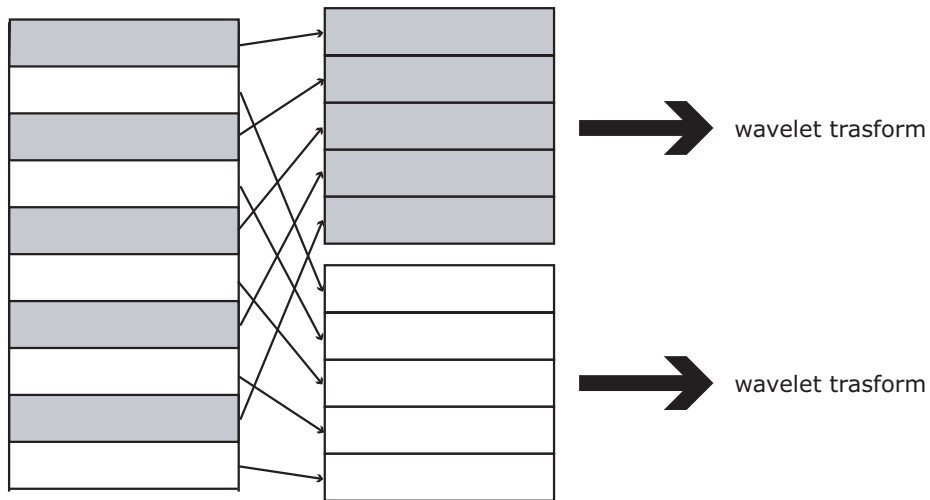


図 2-17: フィールドベース符号化時の Wavelet 変換 (preceding_field_first=1)

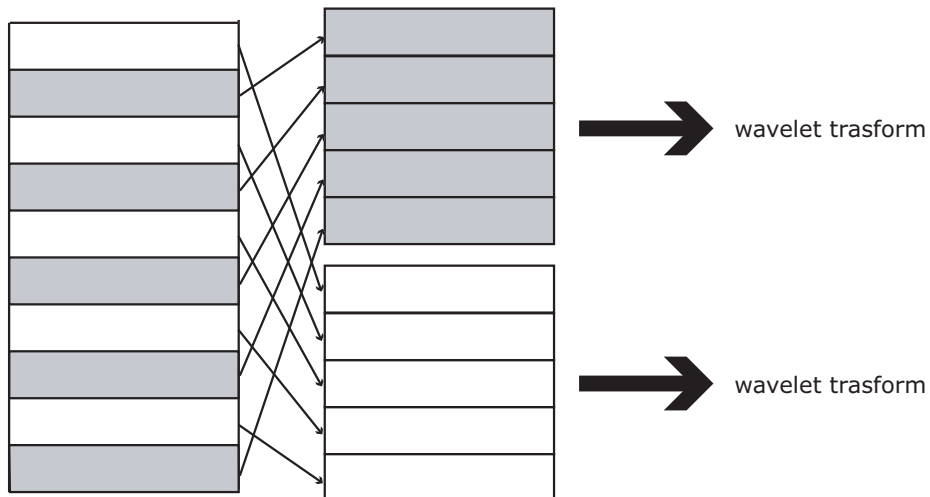


図 2-18: フィールドベース符号化時の Wavelet 変換 (preceding_field_first=0)

2.5.3 Motion JPEG2000 のファイルフォーマット

JPEG2000 のファイルフォーマット

JPEG2000 のファイルフォーマットには、Part-1 がサポートする最小のファイルフォーマットとしての JP2、Part-2 の拡張方式まで包含する JPX、さらに Part-6 の Compound Image 用の JPM が存在している。

ここでは、最小のセットである JP2 について述べる。JP2 ファイルフォーマットは、純粋な符号化コードストリームに特定のアプリケーション用に特化したメタデータを関連付け、これらを保存するデータ形式である。概念的には、JP2 ファイルフォーマットは、符号化コードストリームに関する他の情報と一緒に符号化コードストリームをカプセル化する役割を持つ。

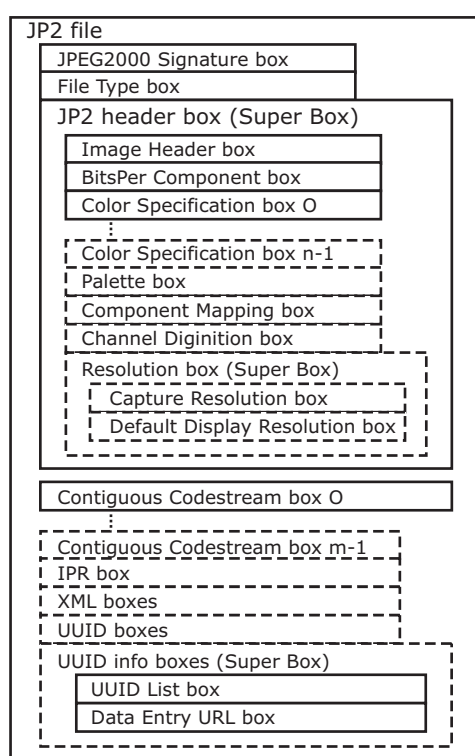


図 2-19: JP2 のファイル構造

JP2 のファイル識別子

JP2 のファイルの識別子は、コンピュータのファイルシステム中に保存される場合には “.jp2” というファイル拡張子になる。Macintosh のファイルシステムでは、JP2 ファイルは “jp2¥040” というタイプコードを与えられる。

JP2 のファイル構造

JP2 ファイルフォーマットの構成要素は “box” と呼ばれる。独立な box もあれば、その中に別の box を包含する box もある。JP2 ファイルフォーマット中に包含されるすべての情報は box にカプセル化される。

JP2 のファイル構造を図 2-19 に示す。また、図中の点線はオプションで、実線は必須の box であることを意味している。図中の box の順序については基本的に自由であると規定されている。しかし、JPEG2000 Signature box だけは必ず先頭に位置していなければならない。さらに File Type Box が JPEG2000 Signature Box の直後に位置し、JP2 Header Box が Contiguous Codestream Box の前に位置することが義務付けられている。

色空間の指定

JP2 ファイルフォーマットでは、画像の色空間を定義するのに 2 つの方法が存在している。1 つ目は色空間の識別番号を示す方法であり、これによって sRGB およびグレースケール画像を定義できる。もう 1 つの方法は、Restricted ICC profile の色空間を用いて、色空間の変換を行うものである。JP2 ファイルフォーマットの場合は、ICC profile は符号化コードストリームを復号して得られたサンプルを、どのようにして標準の色空間 (PCS: Profile

Connection Space) にコンバートするかについて定義している。

ICC Profile Format には2つの異なる ICC profile のクラスが定義されていて、これらは Monochrome Input Profile と Three Component Matrix-Based Input Profile である。これらの profile は、原色空間から PCSXYZ の色空間への変換を、非線形曲線変換および 3x3 マトリクス変換に限定している。

Metadata の指定

JP2 ファイルフォーマットの特徴の1つとして、JP2 file に metadata を追加できるという機能がある。すべての情報は box 中にカプセル化されており、すべての box には type があるため、reader が、自分が理解できない情報を持つ box は無視する一方、関連情報は抽出できるような簡単な構造が、フォーマットに備わっている。

QuickTime のファイルフォーマット

Part-1 の JP2 ファイルフォーマット以外では、Part-2 のファイルフォーマットとして JPX が存在している。この JPX は Part-1 の JP2 を大幅に拡張した形式のフォーマットである。また、Part-3 用のファイルフォーマットとして新たに MJ2 ファイルフォーマットが作られた。Part-3 の符号化コードストリームは、ピクチャ単位では完全に Part-1 準拠の符号化コードストリームである。したがって、Part-3 では符号化シンタックスよりもファイルフォーマットの重要度が高いといえる。

MJ2 ファイルフォーマットは、Apple 社の開発した QuickTime フォーマットをベースに構築された。したがって、QuickTime の atom という box 構造を継承しているほか、数多くの QuickTime の特徴を持っている。MJ2 ファイルフォーマットを理解するためには、QuickTime ファイルフォーマットを理解することは重要である。QuickTime ファイルフォーマットの特徴を以下に示す。

- 動画、静止画、音声などの時間軸を持つすべてのメディアを対象に時間管理を行うことができる。
- 時間軸を持つあらゆるメディアの統合を実現し、かつタイムベースで管理可能
- 豊富なファイル形式に対応可能
- 豊富な AV コーデックを Plug-in 可能
- 豊富なプラットフォームに対応
- Video Streaming にも対応可能

Movie の再生法

図 2-20 は Movie を再生する際の各要素の動きを図示したものである。この中で重要な役割を持つのが、表 2-6 の Function である。図 2-20 を用いて、Movie を再生する際の仕組みについて述べる。まず、Data が Media の中に記録されている。この Data は、QuickTime

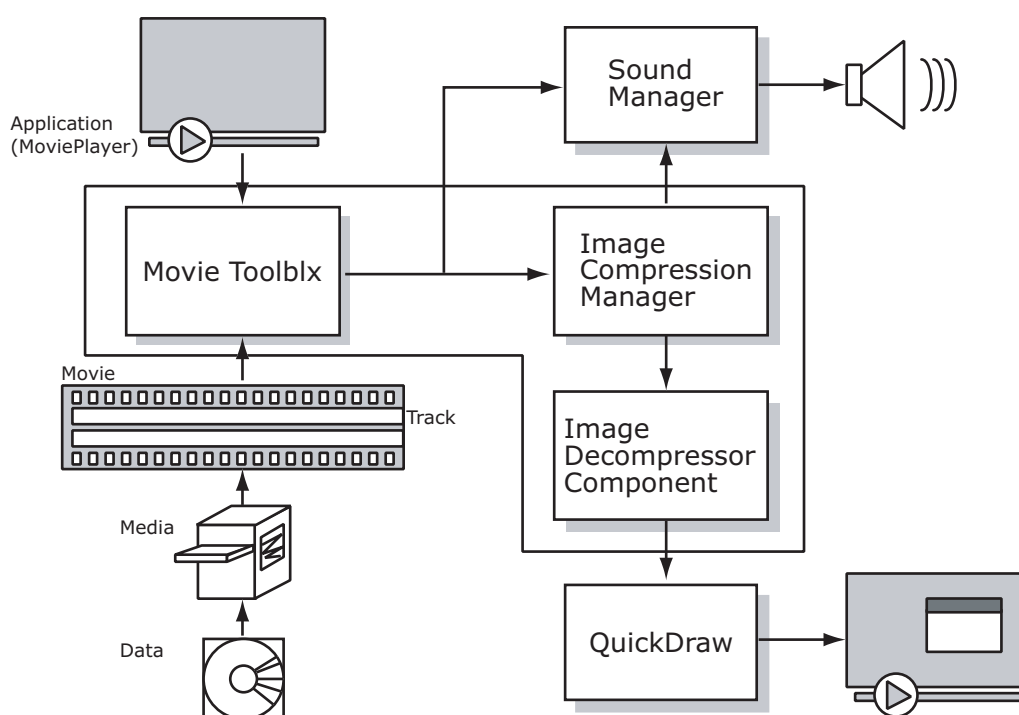


図 2-20: Movie を再生する際の QuickTime のしくみ

ファイルフォーマットに準拠する形で、Movie 中の track 内に記録されている必要がある。一方、Application 側の MoviePlayer を PC などで起動すると、Movie Toolbox が呼び出されて、Track 内に記録されているじっさいの Data のロードが行われる。ロードされた Data は Image Compression Manager に渡されて、圧縮された Data の場合にはそれに適合したデコーダの Component が呼び出される。同時に、その Image Compression Manager において圧縮データの慎重が行われ、伸張された画像は、最終的に QuickDraw を用いて描画される。以上が画像系の仕組みである。

Movie の時間軸と Track

Movie には、1 個以上の Track が含まれる。各 Track は Movie の時間軸システム中で解釈される Media Data を参照する。また、各 Track は Movie の先頭で開始されるが、終了時刻に関する制限はない。さらに Track 内の実データは Movie の先頭時刻から、あるオフセット時刻を経た時刻に記録されている。Movie の先頭時刻に開始されない Track Data は、Empty Space を内部に持つことになる。

図 2-21 は、5 つの Track を持った Movie の例を示している。各 Track を持った Movie の例を示している。各 Track 中の横線の影がついた部分は Movie の先頭時刻と Track Data の開始時刻とのオフセット時間を示している（この部分は各 Track の先頭時刻での Empty Space と見ることできる）。時刻 6 のときに、Video-1、Audio-1、Video-2 の全部で 3 個の Enable 状態の Track が存在している。Other-1 Track は時刻 8 になるまで Enable にはならず、Audio-2 は時刻 10 で Enable になっていることがわかる。

また Movie は、複数個の Layer をその中に持つことができる。各 Layer は相互に関連付

表 2-6: QuickTime で movie を再生するときの重要な機能

| | |
|---------------------------|---|
| Movie Toolbox | Movie の再生や作成および修正を行うための関数の集まりとして構成され，この中の関数を呼び出すことによって QuickTime の機能にアクセスできる |
| Image Compression Manager | 静止画およびビデオシーケンスの圧縮・伸張を行う関数の集まりとして構成され，デバイスやアルゴリズムに関係なく圧縮・伸張できるようにする共通のインターフェイスを提供する |
| Component | “Component Manager” が管理するソフトウェアの実体で，必要なときに一連の定義された機能を呼び出すことで，コンポーネント化された Movie 作成機能を提供する．標準で約 175 のソフトウェアコンポーネントと約 20 のカテゴリから成り，拡張性を念頭に置いたコンポーネントベースとなっている． |

けられた複数個の Track を内包することができる．前述の Movie Toolbox は Layer ごとに Visual Presentation を構築することができる．すなわち Movie Toolbox を用いて Movien layer に個々の Track を割り当てることができるのである．

QuickTime の基本 Box 構造

ここまで，主に時間軸を中心とした QuickTime の Movie のメカニズムについて述べた．ここではさらに QuickTime の基本的な Box 構造について述べる．図 2-22 のように，QuickTime は基本的に Moov と Mdat から構成されている．Moov は Movie Resource の役目を持ち，参照情報（時間管理，データ参照）を記憶・保持していて，QuickTime のファイル内に必ず 1 個存在しなければならない．他方，Mdat には実際のデータ（画像の符号化コードストリームや音声データなど）が入っている．

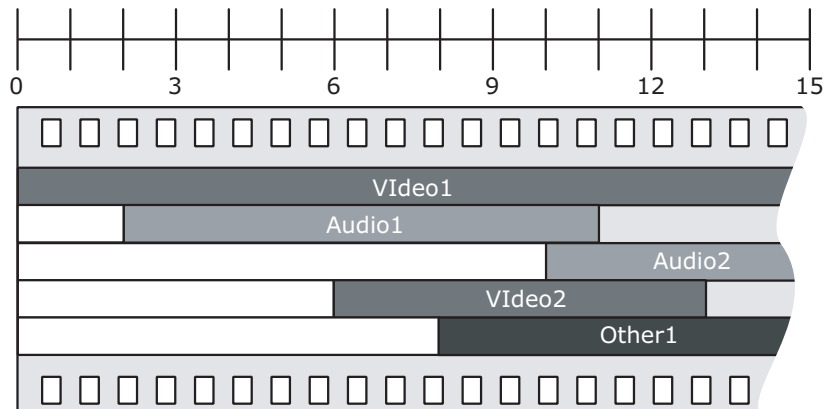


図 2-21: 複数の track を持つ movie の例

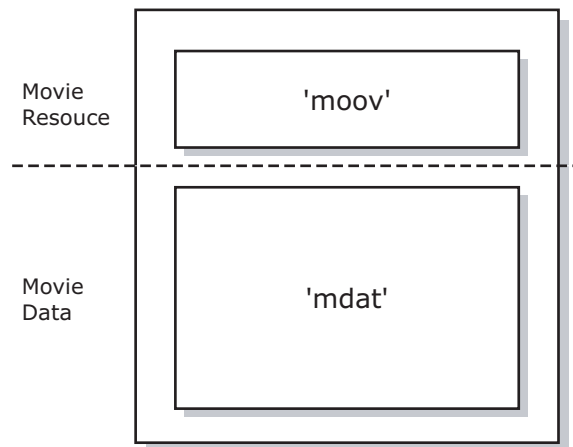


図 2-22: QuickTime の基本的な Box 構造

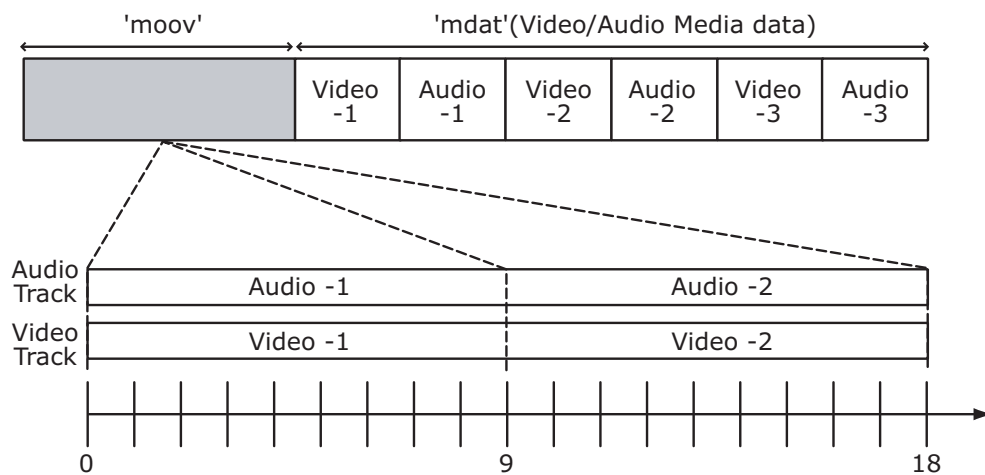


図 2-23: moov による時間管理

図 2-23 は Moov と Mdat の動きを具体的に図示したものである。Moov には時間管理情報、データ参照情報が記録されており、Mdat には Video-1、Video-2、Video-3、Audio-1、Audio-2、Audio-3 の計 6 つのデータが記録されている。図 2-23 の例では、Video Track は時刻 0 から時刻 9 までは Video-1 と Audio-1 が同期してプレゼンテーションできるように Moov 内部で時間管理およびデータ参照管理が行われ、時刻 9 から時刻 18 までは Video-2 と Audio-2 が、時間管理およびデータ参照管理されている。

Motion JPEG2000 ファイルフォーマット

JPEG2000 の Part-1 のファイルフォーマットである JP2 ファイルフォーマットについては図 2-19 を用いて説明した。ここでは、Motion JPEG2000 のファイルフォーマットである MJ2 ファイルフォーマットと JP2 ファイルフォーマットの相違点について述べる。図 2-24 は両者を簡単に示した略図である。実線は必須、破線はオプションであることを示す。

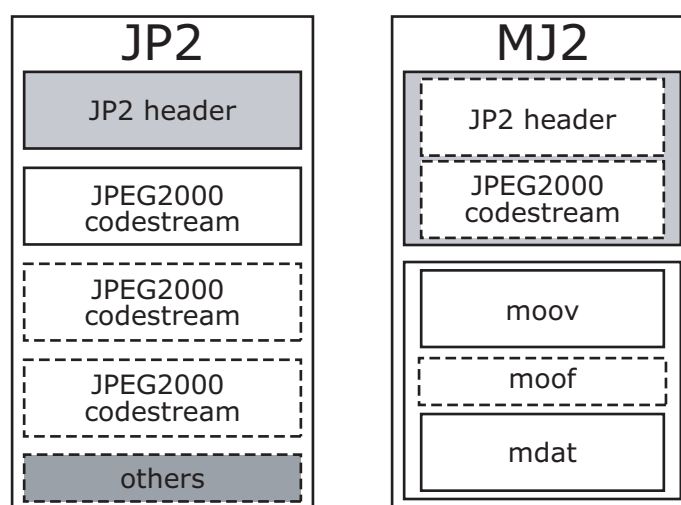


図 2-24: jp2 と mj2 の構造

以下は MJ2 フォーマットの特徴である。

1. JP2 ファイルフォーマットと同じ JP2 Header と 1 つの JPEG2000 Codestream をオプションとして持つことができる。
2. 1. の場合, JP2 ファイルフォーマットのように複数個の JPEG2000 Codestream を持つことは許されない。
3. Moov, Mdat のほかにオプションとして moof を持つことができる。

MJ2 ファイルフォーマットの Box 構造

MJ2 ファイルフォーマットでは, 数多くの Box が階層構造を形成して存在しており, すべての Box に各々の働きがある。表 2.5.3 はこれを表にしたものである。Moov と Mdat が上位階層に位置しているのがわかる。

ファイルフォーマットのファミリメンバ

Motion JPEG2000 のファイルフォーマットはすでに述べたとおり MJ2 だが, 共通のフォーマットを持ったその他のファイルフォーマットがファミリとして存在する。その他のファミリメンバとしては, 以下のファイルフォーマットが存在する

- JPEG2000 用のファイルフォーマット JP2
- MPEG-4 用のファイルフォーマットである MP4
- QuickTime 用のファイルフォーマット (MP4 と MJ2 のベース)

これらのファイルフォーマットの仕様は, File Structure の定義として共通のものを用いている。また Size や Type といった Object の一般的な構造の定義も共通性を持たせている。

これらの定義の中で判別不可能な Object については, これを Reader が無視できるようになっている。また実際に Conflict が存在しないはずなのに Conflict が発生した際には, ここで定義された仕様を最優先することを取り決めている。

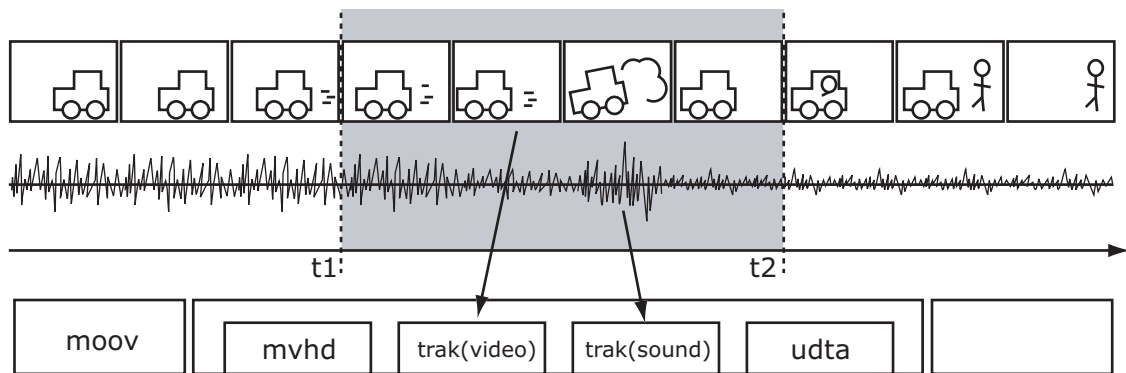


図 2-25: MJ2 ファイルフォーマットでの画像と音声の同期 (コントロールフロー)

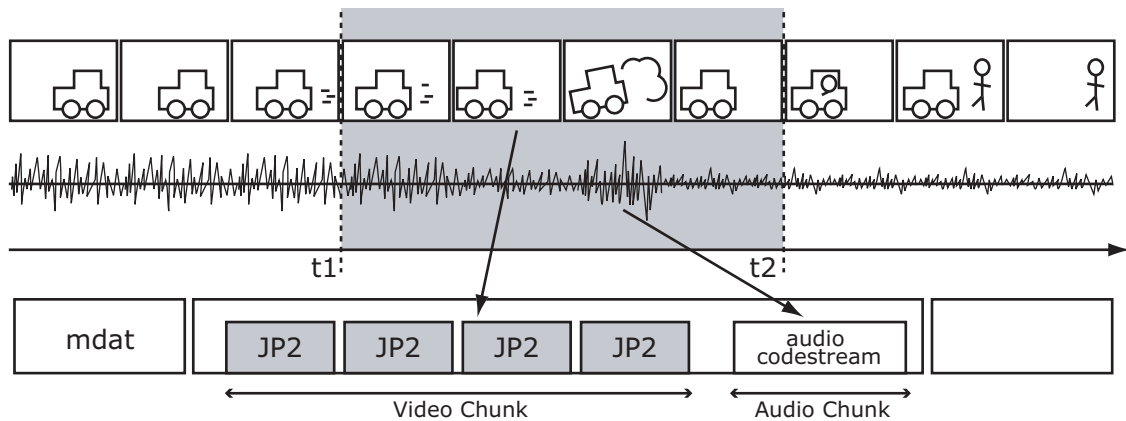


図 2-26: MJ2 ファイルフォーマットでの画像と音声の同期 (データフロー)

MJ2 ファイルフォーマットによる制御方法

MJ2 ファイルフォーマットを実際に用いた制御方法について、図 2-25, 2-26 を用いて述べる。図 2-25 は Moov の管理情報について図示したものである。時刻 t_1 から t_2 までの時間のビデオ情報が Trak(Video) に、音声情報が Trak(Sound) に記録されている。一方、Mvhd は表 2.5.3 にもあるように Movie のヘッダ情報を保持している。また Udfa はユーザ情報や Copyright 情報を記録することができる。一方、図 2-26 は Mdat 内部のデータ情報を示している。時刻 t_1 から t_2 までの時間のビデオの実データは 4 つのピクチャ分存在しているので、4 個の JP2 Codestream があり、これらをまとめて Video Chunk と呼ぶ。

一方、音声は時刻 t_1 から t_2 までの音声の実データが Audio Codestream として mdat 内に記録されている。以上のように Moov 内部の Trak 情報と Mdat 内部の実データの情報の両方を利用することで、ビデオと音声の同期を取りながらプレゼンテーションを実現することができる。

2.6 おわりに

本章では、本研究で用いる符号化方式である JPEG2000 について説明した。JPEG2000 の概要と符号化処理のアルゴリズムとその技術について述べた。特に、分割符号化における分割単位であるタイルとプレシントについて説明を行った。さらに、JPEG2000 の動画像フォーマットである Motion JPEG2000 の概要を述べた。

表 2-7: Box のタイプと階層構造

| | | | | | | |
|------|------|------|------|------|------|---|
| jP | | | | | | the JP2 family signature |
| ftyp | | | | | | file type and compatibility |
| moov | | | | | | container for all the meta-data |
| | mvhd | | | | | movie header, overall declarations |
| | trak | | | | | container for an individual track or stream |
| | | tkhd | | | | track header, overall information about the track |
| | | tref | | | | track reference container |
| | | edts | | | | edit list container |
| | | | elst | | | an edit list |
| | | mdia | | | | container for the media information in a track |
| | | | mdhd | | | media header, overall information about the media |
| | | | hdlr | | | handler, declares the media(handler) type |
| | | | minf | | | media information container |
| | | | | vmhd | | video media header, overall information (video track only) |
| | | | | smhd | | sound media header, overall information (sound track only) |
| | | | | hmhd | | hint media header, overall information (hint track only) |
| | | | | dinf | | data information box, container |
| | | | | | dref | data reference box, declares source(s) of media data in track |
| | | | | stbl | | sample table box, container for the time/space map |
| | | | | | stsd | sample descriptions (codec types, initialization etc.) |
| | | | | | stts | (decoding) time-to-sample |
| | | | | | stsc | sample-to-chunk, partial data-offset information |
| | | | | | stsz | sample sizes (framing) |
| | | | | | stco | chunk offset, partial data-offset information |
| | mvex | | | | | movie extends box |
| | | trex | | | | track extends defaults |
| moof | | | | | | movie fragment |
| | mfhd | | | | | movie fragment header |
| | traf | | | | | track fragment |
| | | tfhd | | | | track fragment header |
| | | trun | | | | track fragment run |
| mdat | | | | | | media data container |
| free | | | | | | free space |
| skip | | | | | | free space |
| | udta | | | | | user-data, copyright etc. |

第3章

分割による低遅延符号化方式

3.1 はじめに

本章では、まず入力画像を分割することによって符号化遅延を低減する概念を説明する。分割の単位を JPEG2000 のタイルとプレシントとし、符号化遅延を低減する符号化方式を提案する。さらに、分割符号化を行うにあたって、動画像としての品質を維持するために、Explicit 量子化とポスト量子化を組み合わせた符号量制御方式を提案する。最後に、実験によって提案方式の性能評価を行う。

3.2 分割符号化の概念

本章では、符号化遅延低減のための分割符号化方式について述べる。第2章で述べたように、フレーム内符号化である JPEG2000 を用いることによって、フレームの並び替えなど、動き補償に起因する遅延を大きく低減できる。一方で、符号化遅延のもう一つの大きな要因として、入力画像の読み込み時間があげられる。JPEG2000 の符号化処理の中では、入力画像の読み込みが完了してから、符号化処理が行われる。したがって、符号化遅延を低減するためには、入力画像の読み込み時間を削減する必要がある。

そこで、本章では入力画像の読み込み時間を低減するために、入力画像を分割して符号化する方式をとる。入力画像を分割し、読み込んだ分割領域から順次符号化を行う。これによって、画像全体を読み込まずに、分割した領域ごとに符号化を行うことが可能である。分割符号化の概念を図 3-1 と図 3-2 に示す。

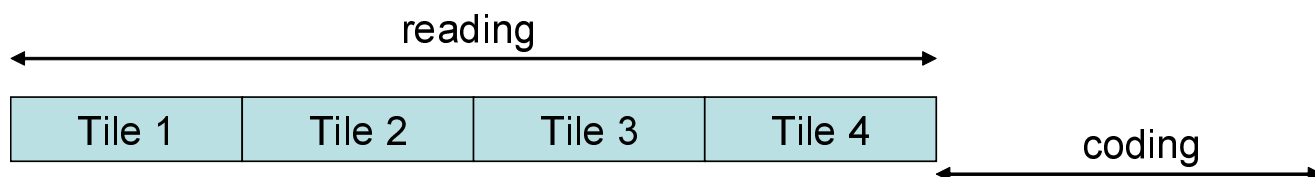


図 3-1: 入力画像非分割時の読み込みと符号化

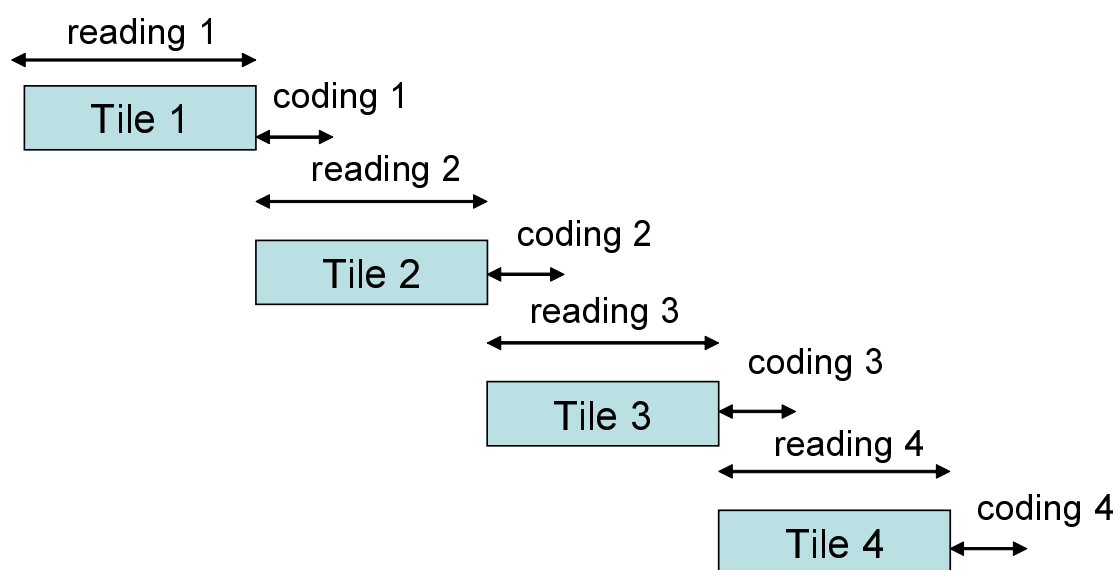


図 3-2: 入力画像分割時の読み込みと符号化

3.3 JPEG2000 タイル分割の符号化遅延低減への応用

JPEG2000 では、第 2 章で述べたように分割の単位としてタイルがある。分割されたタイルは、それぞれ独立に符号化される。そのため、入力画像の読み込み時間を短縮し、符号化遅延の低減に利用できる。タイル分割を用いて符号化遅延を低減するにあたって、各タイルに対して適切なビット配分が行われないと、それぞれ符号化されたタイルを復号し、合わせて再生した際に、画質の劣化が生じるという問題点がある。そこで、本章では、各タイルの画像特性を考慮した符号量制御の方式を提案する。

3.3.1 R-D(Rate-Distortion) 最適化

JPEG2000 の符号量制御⁽⁷⁾ では、各フレームの R-D 特性が用いられる。Wavelet 変換を行った後、Wavelet 係数は EBCOT(Embedded Block Coding with Optimized Truncation) 処理によってビットプレーンさらには符号化パスに分解される。その後、歪が最小となるように、R-D 最適化に基づいて、ビット量の調整が行われる。

JPEG2000 では、画像を符号化した後に、目標とするビットレート、歪 (画質) に適合するように、R-D 最適化を用いて、各コードブロックのビットを調整する。

まず、画像全体の歪を D とすると、 D は以下のように表される。

$$D = \sum_i D_i^{n_i} \quad (3.1)$$

式 (3.1) において、 i はコードブロック、 n_i はコードブロック i におけるビット切捨て点を表し、 $D_i^{n_i}$ はコードブロック i の歪である。歪の計算には、ウェーブレット変換が直交変換であることより、MSE または重み付け MSE が用いられる。また画像全体の符号量を R とす

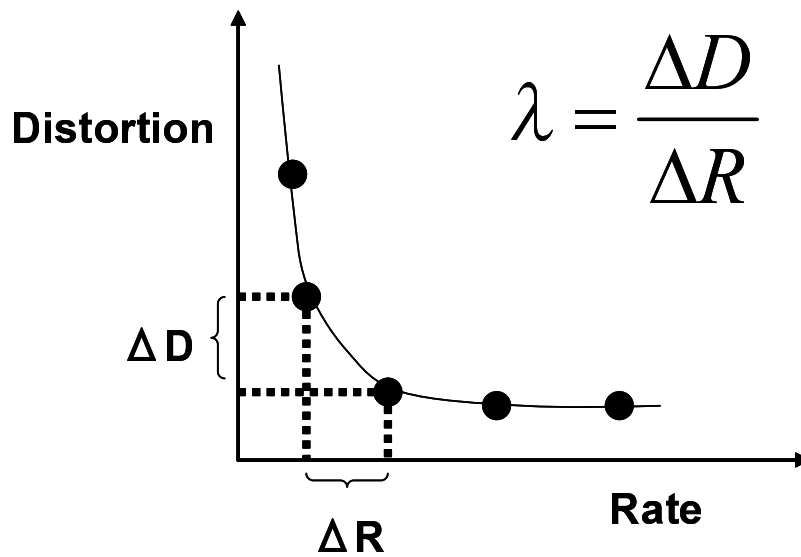


図 3-3: Rate,Distortion, の関係

ると, R は以下のように表される.

$$R = \sum_i R_i^{n_i} \tag{3.2}$$

式 (3.2) において, $R_i^{n_i}$ はコードブロック i の符号量である. ここで, R を最小にするためには, D が最小となる n_i を決定することが必要である. これを解決するための方法として, Lagrange 未定係数法を利用でき, (3.3) で示される値を最小にすることによって最終的に最適歪となる符号列を得られる^{(7),(21)}.

$$\sum R_i^{n_i} - \lambda D_i^{n_i} \tag{3.3}$$

そこで, 各符号化パスで, Δ_R と Δ_D を用いて, を計算で求め, ビット切捨て点を決定する. Rate と Distortion, さらに の関係を図 3-3 に示す.

$$= \Delta_D / \Delta_R \tag{3.4}$$

式 (3.3) で示される値を最小にするということは, すなわち符号量制御において, の値が小さい符号化パスから切り捨てて目標符号量に合わせることと同等である. 以上の R-D 最適化を行うことによって, JPEG2000 符号化では目標ビットレートに対して最小の歪を達成することが可能である.

3.3.2 想定環境

本章で想定する符号化器のブロック図を図 3-4 に示す. 映像入力として, 1080i HDTV 信号を想定し, JPEG2000 符号化がビデオフィールドまたは後述するタイル単位に連続的に適用される. 符号化, および復号に伴う処理遅延を短縮する目的から, JPEG2000 符号化

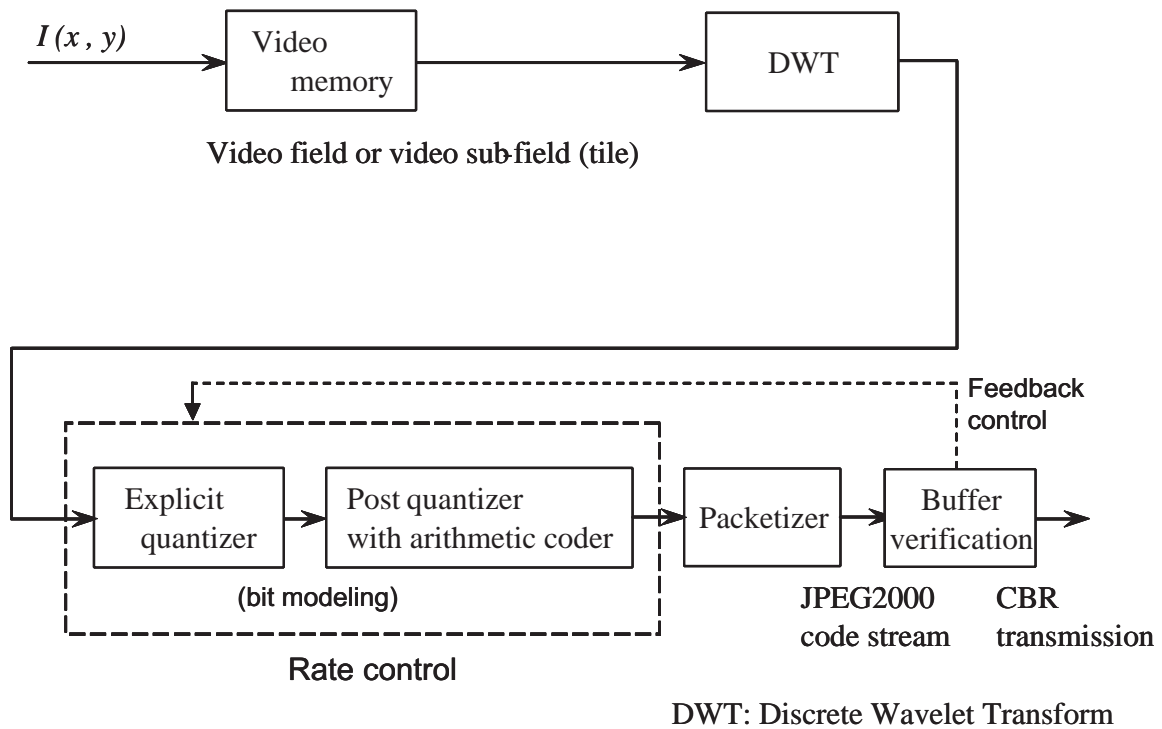


図 3-4: 想定する符号化器

においてタイル構造を採用することとした。JPEG2000 のタイル構造において、符号化対象となるビデオフィールドはタイルと呼ばれる複数の矩形エリアに分割され、タイルごとに JPEG2000 符号化処理が個別に適用される。MPEG-2 や MPEG-4 といった動画用の標準符号化方式では一般に、エンコーダ出力が CBR (Constant Bit-Rate) 符号化に適合するかどうかを検証するためのバッファモデルが規定されている⁽⁵⁾。同検証用バッファのことを以下では単に検証バッファと呼ぶ。バッファモデルの導入に伴う弊害として、バッファサイズに比例したバッファリング遅延の増大が懸念される。バッファリング遅延を最小化する目的から、本章で扱う符号化システムでは、JPEG2000 の符号化シンタックスは維持しつつ、検証バッファにて、各タイルに相当する部分的な符号化ストリームをビデオアクセスユニットとして扱うこととした。本章では、ピクチャという用語は、ビデオフィールドに相当する符号化ストリームを指し、インデックス i, j をタイルの識別に用いる。本インデックスの定義を図 3-5 に示す。あるタイルに対して、 i は時間的な順序を示し、 j は当該ビデオフィールド内での空間的な順序を 0 から $(tile_cnt - 1)$ の連番で示す。tile_cnt はビデオフィールドを構成するタイルの総数である。

3.4 予備実験

JPEG2000 において、分割符号化の単位としてタイルがあげられる。JPEG2000 の参照ソフトである文献⁽²²⁾は、タイル分割符号化に対応している。しかし、実際に符号化する際に

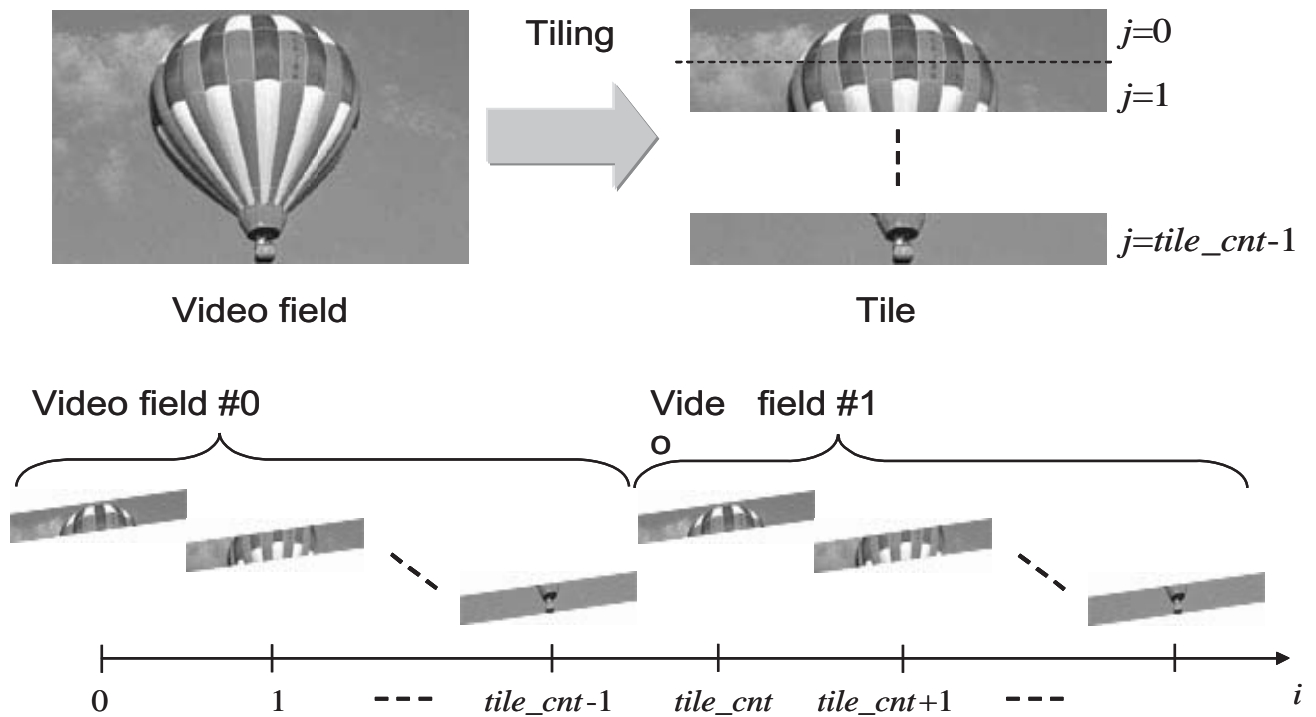


図 3-5: ビデオフィールドに対するタイル分割処理

は、画像全体について R-D 最適化を行い各タイルに割り当てるビット量を決定している。すなわち、先頭タイルを符号化するにあたって、先頭タイルだけでなくピクチャ全体を読みこみ、さらに R-D 最適化を行い、その後ようやく先頭タイルが符号化されることになる。これは R-D 最適化を行っているため、品質的にはピクチャとして最適になるが、ピクチャ全体を読み込むまで、先頭タイルの符号化が始められず、分割することによる符号化遅延を低減することができなかった。

分割することによって、符号化遅延を低減するためには、ピクチャ全体を読み込む前に、先頭タイルから読み込んだ順に各タイルにビット量を割り当てなくてはならない。しかし、先頭タイルを読み込んだ時点では、ピクチャ内の後続タイルの情報がわからないために、適切にビット量を割り当てることは困難である。

ここで、まず予備実験として単純計算で各タイルに割り当てるビット量を計算する方法を考える。

$$R(t) = \frac{(L_W + t) \times B_W / \text{tile_cnt} - \sum_{i=0}^{t-1} S(i)}{L_W} \tag{3.5}$$

- L_W : 符号量制御のウィンドウサイズ
- $S(i)$: i 番目のタイルにて発生した符号化ビット数

当該タイル (再生順序を t 番目とする) の符号化処理に先立ち、目標符号量を算出する。ここでは、文献⁽²⁵⁾と同様のスライディングウィンドウをタイル単位の符号量割当に適用する

表 3-1: 符号化条件

| | |
|------------------|------------------------------------|
| 入力画像 | 1920×1080 4:2:2 コンポーネント (8 ビット) |
| 符号化レート | 0.8bit/pixel |
| DWT フィルタ | (9,7) 非可逆フィルタ |
| サブバンド分割レベル | 3 レベル |
| コードブロックサイズ | 128 × 32 |
| Visual weighting | 非適用 |

こととした。具体的には、実際に割り当てたビット量と理想値との誤差を吸収する目的から、 t 番目のタイルに割当可能な符号量 $R(t)$ を式 (3.5) により算出する。 B_W はビデオフィールドに割り当てられる平均符号化ビット数である。

以上の計算によるビット配分方式を方式 (3.5) と呼ぶことにする。タイル分割符号化において、各タイルに対するビット配分が品質にどれほどの影響を与えるかを確認するために以下のような予備実験を行う。方式 (3.5) によりタイル分割符号化した場合と、文献⁽²²⁾のビット配分法によりタイル分割符号化した場合について、再生画像の品質を比較する。符号化条件は、表 3-1 に示すとおりである。Streetcar と Whaleshow の各 1 フィールドについて、品質の比較の結果を 3-2 に示す。また、符号化における入力画像の読み込み時間について比較した結果を表 3-3 に示す。

表 3-2: 分割符号化の品質に与える影響

| Label | PSNR [dB] | |
|------------------------------|-----------|-----------|
| | Streetcar | Whaleshow |
| 文献 ⁽²²⁾ でタイル 4 分割 | 35.189 | 32.080 |
| 文献 ⁽²²⁾ でタイル 6 分割 | 35.144 | 31.978 |
| 方式 (3.5) でタイル 4 分割 | 33.444 | 30.961 |
| 方式 (3.5) でタイル 6 分割 | 31.888 | 29.798 |

表 3-3: 分割符号化の遅延に与える影響

| Label | Reading Delay [msec] |
|------------------------------|----------------------|
| 文献 ⁽²²⁾ でタイル 4 分割 | 16.68 |
| 文献 ⁽²²⁾ でタイル 6 分割 | 16.68 |
| 方式 (3.5) でタイル 4 分割 | 4.17 |
| 方式 (3.5) でタイル 6 分割 | 2.66 |

表 3-2 より、品質については文献⁽²²⁾による R-D 最適化時は、方式 (3-5) から求めたビット配分時と比較して、0.8bpp で 1~3dB 品質が良好であることがわかる。しかし、遅延時間については、表 3-3 より文献⁽²²⁾による R-D 最適化時はピクチャ全体を読み込まなくてはならないため、読み込み時間が 16.68msec かかり遅延を抑えられていないことがわかる。一方で、方式 (3-5) によるビット配分法では読み込み時間は、タイル 4 分割と 6 分割でそれぞれ 4.17msec、2.66msec に抑えられることがわかる。

この結果から、文献⁽²²⁾によるタイル分割方式では品質では理想的だが、タイルに分割することによって符号化遅延を低減できない。一方で、方式 (3-5) のビット配分に基づく符号化では、各タイルに独立に符号量制御を行っているために、読み込み時間は短縮可能である。しかし、不適切なビット配分によって、R-D 最適化状態と比較して品質が低下している。この結果を踏まえて、文献⁽²²⁾の R-D 最適化時の品質と同等でかつ、方式 (3-5) によるビット配分時と同等に符号化遅延を低減する必要がある。

3.5 提案レート制御方式

3.5.1 量子化誤差分布の制御

JPEG2000 符号化処理では符号量制御において R-D 最適化が行われる。同手法により MSE は最小となるが、サブバンド内でコードブロックごとに符号化された符号化パス数に開きが生じ、これが 2 のべき乗のオーダで量子化誤差に反映される結果、歪み量は空間的に偏った分布となり得る。このため動画像符号化への応用時には、同分布が時間的に変化することで、主観画質を低下させる可能性がある⁽²³⁾。

文献⁽²³⁾は、ポスト量子化と Explicit 量子化を併用するという点において、提案方式と一致するが、符号量の割当に関して、文献⁽²³⁾では、フィールドあるいはフレームといった画面単位での処理を基本としている。このため、タイルやプレシントといった分割画面単位の符号化を適用することが低遅延化には寄与しない方式となっていた。同問題を解決するために、提案方式では符号量割当の単位、およびビデオアクセスユニットの単位をともに分割画面 (タイルまたはプレシント) として細かく制御することにより、コーデック遅延の大幅な削減を図る。以上の要件を満足する目的から、以下の前提に基づきレート制御手法の検討を行った。

- 図 3-4 に示すとおり、量子化の精度を高く維持し、かつ符号化パス数を制御するために、Explicit 量子化とポスト量子化の併用を行うことを基本とする。タイルの符号化に先立ち、目標符号量が割り当てられ、一連の量子化制御から成るレート制御によって目標値が達成される。
- 量子化誤差の空間的な分布が偏るのを防ぐため、ポスト量子化において廃棄する符号化パス数 n_p (n_p は整数) は、タイル内の全コードブロックで同一とする。 n_p に対する最適値の導出手法については後述する。
- Explicit 量子化において、量子化テーブルは式 (3-6) により行列 δ として定義され

る．行列 $delta$ の要素はサブバンドごとに存在し， $band$ はサブバンドの識別に用いられる． dqt は量子化テーブル $delta$ の初期値として参照される値を指し， $cq(i)$ は i 番目のタイルに対し dqt の各要素に共通に適用される重み係数である． $cq(i)$ は当該ビデオフィールドの先頭タイルの符号化に先立ち，直前ピクチャの結果をもとに更新される．

$$delta[band] = dqt[band] \times cq(i) \quad (3.6)$$

- 検証バッファのサイズは，タイルごとの発生ビット数の変動の大きさに比例して増加する．それに伴い，コーデックのバッファリング遅延も増加する．よって，バッファリング遅延の低減のため検証バッファサイズを最小化する目的から，タイルごとの発生ビット数の変動は低く抑えるものとする．
- ビデオフィールド内で j 番目のタイルに対して，量子化パラメータと符号化ビット数の関係は，式 (3.7) に示す非線形関数 f_j により近似可能であるとする．同式にて R は符号化ビット数を示し， Q は LL サブバンドにおける $delta[band]$ の値を表す． (Q, R) の実データサンプルは，当該タイルの実際の符号化結果をもとに式 (3.8) を用いて得られる．式 (3.8) にて npl_i は切り捨てられたビットプレーンの数を示し， $r(i, npl)$ は i 番目のタイルにて npl に相当する符号化パス数を切り捨てた際の発生符号量を示す．関数 f_j の更新は， j 番目に位置するタイルの符号化が完了する都度，式 (3.9) に従い行われる．同式において， α_j, β_j, C_j は関数を決定するパラメータである． C_j は j 番目のタイルの発生符号量のうち，ヘッダパートに相当する符号量を示す．

$$R = f_j(Q) \quad (3.7)$$

$$Q = cq(i) \times 2^{npl_i}, R = r(i, npl_i \times 3) \quad (3.8)$$

$$f_j(Q) = \alpha_j \times Q^{\beta_j} + C_j \quad (3.9)$$

提案するレート制御手法について，具体的な処理手順を以下に説明する．

目標符号量の算出 当該タイル (再生順序を t 番目とする) の符号化処理に先立ち，目標符号量の算出が必要となる．提案方式では，文献⁽²⁵⁾と同様のスライディングウィンドウをタイル単位の符号量割当に適用することとした．具体的には，検証バッファの占有量に関して理想値に対する誤差を吸収する目的から， t 番目のタイルに割当可能な符号量 $R(t)$ を式 (3.10) により算出する． B_W はビデオフィールドに割り当てられる平均符号化ビット数である．

$$R(t) = \frac{v_0 + (L_W + t) \times B_W / tile_cnt - (v_T + \sum_{i=0}^{t-1} S(i))}{L_W} \quad (3.10)$$

| | |
|--------|---------------------------------|
| L_W | : 符号量制御のウィンドウサイズ |
| v_0 | : 検証バッファにて先頭タイルが引き抜かれる直前のバッファ残量 |
| v_T | : タイルの引抜き直後における検証バッファ残量の目標値 |
| $S(i)$ | : i 番目のタイルにて発生した符号化ビット数 |

Q-table の更新 当該ピクチャの符号化に備え $cq(t)$ の算出を行う。当該ピクチャの先頭タイルを、再生順序が t 番目のタイルとする。提案方式では np を 3 に保持するという規範のもと、 $cq(t)$ を決定する。符号化処理の中で各ビットプレーンは、画質への寄与度を考慮して 3 つの符号化パスに分解される。つまり意図した制御が安定的に働いている状況下では、ポスト量子化処理によって 1 ビットプレーンに相当する 3 つの符号化パスが切り捨てられることを意味する。 t 番目のタイルに対する量子化パラメータの目標値として、式 (3-11) を満足する $Q(t)$ を $d(t)$ として選択した後、式 (3-12) により $cq(t)$ を決定する。JPEG2000 の符号化シンタックス上、Q-table の記述はピクチャごとに付与されるメインヘッダーにおいて必須であるが、タイルパートヘッダーではオプション扱いである点を考慮し、以上の処理はビデオフィールドにおける先頭タイル ($j = 0$) に限り適用する。

$$R(t) = \sum_{j=0}^{tile_cnt-1} f_j(Q(t))/tile_cnt \quad (3-11)$$

$$d(t) = cq(t) \times 2 \quad (3-12)$$

ポスト量子化 ポスト量子化処理において、当該タイルに対する目標符号量が $f_j(d(t))$ により与えられ、これを満足する np の値が以下の手順で決定される。まず、検証バッファに対してオーバーフローおよびアンダーフローを発生させないという要件を設け、選択可能な候補を閉区間 $[np_{min}, np_{max}]$ に限定する。同候補より $|r(t, np) - f_j(d(t))|$ を最小とする np を全探索により特定する。ここで $r(t, 0)$ が目標値 $f_j(d(t))$ を下回る場合には、すべての符号化パスを通過させた場合においても発生ビット数が不足してしまう。同問題に対処するため、 $cq(t)$ の算出方法は、上述の通りポスト量子化部に入力される符号量が目標値を 1 ビットプレーンに相当する分だけ上回るよう設計したものである。1 ビットプレーン上回るということは、2 進数で 1 ビットシフトすることに相当するため、式 (3-12) において $cq(t)$ に 2 を乗じている。

3.6 性能評価

3.6.1 符号化条件

提案方式による符号化性能を評価する目的から、符号化実験を行った。本章で使用する符号化条件は表 3-4 に示した通りである。提案要素技術に対応した符号化シミュレーション

表 3-4: 符号化条件 2

| | |
|------------------|---|
| 入力画像 | 1920×1080 / 60Hz (インターレース) 4:2:2 コンポーネント (8 ビット) |
| 符号化レート | 0.5bit/pixel, 0.8bit/pixel |
| DWT フィルタ | (9,7) 非可逆フィルタ |
| サブバンド分割レベル | 3 レベル |
| コードブロックサイズ | 128 × 32 |
| Visual weighting | 非適用 |

ツールのソフトウェア実装は、JPEG2000 のコーデックソフトウェアである Kakadu⁽²⁶⁾ をベースに行った。入力には ITE HDTV 標準テスト画像より以下の 3 種類を使用した。評価対象とするタイル構造は、表 3-5 に示す 2 パターンとした。性能改善の評価目的から比較対象とする従来方式として、文献⁽²²⁾、文献⁽²³⁾ による方式を取り上げることとした。ここで文献⁽²²⁾ はポスト量子化のみにより画面単位の符号量制御を行っており、画面ごとに与える符号量は既知であることが前提であるため、各画面に一律の符号量を与えることとする。またポスト量子化処理が画面単位となっているため、タイルやプレシントといった分割画面単位の符号化を適用した場合においても、分割画面単位の符号化とはならず、文献⁽²³⁾ と同様にこれらの符号化ツールが低遅延化に寄与しない。一方で文献⁽²¹⁾ は文献⁽²²⁾ に対する高速化実装のアプローチであり、性能的には文献⁽²²⁾ と同等レベルを標榜するものである点から、比較対象としては両方式を文献⁽²²⁾ で代表できるといえる。

- European market

画面中央を汽車が移動する。カメラ操作はなし。
 汽車の色調が鮮明であるが精細部は少ない。
 第 300 フレームを開始点として 120 フィールド

- Whale show

シャチの動きを追う素材。水平方向に高速なパニングを伴う区間あり。観客席に精細部を含む。
 第 240 フレームを開始点として 120 フィールド

- Street car

画面中央を路面電車が横切るシーン。カメラ操作はない。電車やビルの広告ロゴなど、部分的に精細部を含む。
 第 330 フレームを開始点として 120 フィールド

表 3-5: 対象とするタイル構造

| Label | Horizontal size | Vertical size | <i>tile_cnt</i> |
|-------|-----------------|---------------|-----------------|
| A) | 1920 | 135 | 4 |
| B) | 1920 | 90 | 6 |

3.6.2 符号化性能

提案手法の符号化性能を評価するため、Y 信号について求めた PSNR のシーケンス平均を表 3-6 にまとめる。以下、本文中で PSNR の表記は、Y 信号の PSNR を意味する。参考までにタイル分割を行わずに提案レート制御を適用した場合の結果についても併せて示した(表中 No tiling)。比較のため、従来方式として文献⁽²²⁾、文献⁽²³⁾による PSNR のシーケンス平均を表 3-7 にまとめた。ただし、文献⁽²³⁾でタイル分割を行わない場合の結果は、原理的に提案方式と同一であるため省略した。まず各タイル構造において、提案方式の従来方式に対する PSNR の低下は高々 0.3dB 程度である。文献⁽²²⁾、文献⁽²³⁾との差はレート制御をタイル単位としたことに起因し、特に文献⁽²²⁾に限ってはレート制御を MSE 最小化規範で行っていることによる差も積まれており、トータルの品質差に対する寄与度としては後者が支配的である。次に、タイル構造と PSNR の関係について考察する。

結果より、符号化レートと入力画像に関するいずれの組合せにおいても、タイル分割の導入に伴う PSNR の低下は高々 0.2dB であることが分かる。主観評価結果より、同効率低下は主観画質上、無視できるレベルであることを確認している。

分割数が増えるにつれて効率が低下するのは主に以下 2 点が原因である。

- 各タイル境界部分の DWT 係数の取得に際し、折り返し処理を施している。
- タイル単位にタイルパートヘッダーを伴い、サイド情報に費やす符号化ビット数が増加する。提案方式では、1 タイルパートヘッダは 112 ビットとなる。したがって、4 分割した場合は 448 ビット、6 分割した場合は 672 ビットのタイルパートヘッダが発生する。

各タイルに適用される np の値を観測した結果、 $R(t)$ として初期値が適用される先頭ビデオフィールドを除いて、常に 3 に保持されていることを確認した。これより、符号化パス数はコードブロック間で均等に保持され、R-D 最適化手法の導入時に懸念された符号化パス数の開きに起因する劣化要因は解消されているといえる。さらに実際の量子化パラメータは、ビデオフィールドを構成するタイル間で同一の値に保持されており、例えば隣接タイル間の著しい品質差など、タイルベースのレート制御に特有の劣化要因は発生していないといえる。

3.6.3 コーデック遅延量

提案したレート制御の安定性を評価する目的から、検証バッファにて各ビデオアクセスユニットが引き抜かれた直後の占有量を観測した。提案方式で必要なバッファサイズを確認す

表 3-6: 提案方式による PSNR の結果

| Test sequence | Bit-rate (bit/pixel) | PSNR [dB] | | |
|-----------------|----------------------|-----------|-------|-----------|
| | | A) | B) | No tiling |
| European market | 0.5 | 35.83 | 35.75 | 35.94 |
| | 0.8 | 38.00 | 37.95 | 38.07 |
| Whale show | 0.5 | 31.00 | 30.91 | 31.11 |
| | 0.8 | 34.20 | 34.13 | 34.29 |
| Street car | 0.5 | 33.37 | 33.31 | 33.47 |
| | 0.8 | 35.65 | 35.60 | 35.72 |

表 3-7: 従来方式による PSNR の結果

| Test sequence | Bit-rate (bit/pixel) | PSNR [dB] | | | | |
|-----------------|----------------------|--------------------|-------|--------------------|-------|-----------|
| | | 文献 ⁽²³⁾ | | 文献 ⁽²²⁾ | | |
| | | A) | B) | A) | B) | No tiling |
| European market | 0.5 | 35.85 | 35.81 | 35.98 | 35.92 | 36.07 |
| | 0.8 | 38.03 | 37.99 | 38.17 | 38.13 | 38.22 |
| Whale show | 0.5 | 31.05 | 30.98 | 31.25 | 31.19 | 31.33 |
| | 0.8 | 34.20 | 34.14 | 34.40 | 34.33 | 34.48 |
| Street car | 0.5 | 33.39 | 33.36 | 33.58 | 33.54 | 33.64 |
| | 0.8 | 35.66 | 35.61 | 35.96 | 35.93 | 36.03 |

るために、仮の検証バッファサイズ L_T を最小のケースとして 1 タイルあたりの平均符号量に相当する ($B_W/tile_cnt$) に設定し、式 (3-5) の v_0, v_T はそれぞれ、式 (3-13)、式 (3-14) に従うこととした。仮の検証バッファサイズに対する不足を考慮して、提案方式で必要なバッファサイズを決定する。一例として素材 European market に対しタイル構造 A), B) のもとで観測した検証バッファ占有量の時間遷移をそれぞれ、図 3-6、図 3-7 に示す。図 3-6、図 3-7 において、上の図はビットレートが 0.5bpp の時、下の図は 0.8bpp の時の結果を表す。縦軸はバッファ占有量を L_T により正規化した値で示す。

$$v_0 = L_T \quad (3-13)$$

$$v_T = 0 \quad (3-14)$$

図 3-6、図 3-7 において縦軸の変動幅は、バッファ占有量を直接表すのではなく、バッファサイズ L_T について、検証バッファをオーバーフローまたはアンダーフローさせないために必要となるバッファサイズに対する不足分とみなせる。したがって、縦軸がマイナスになってもそれはアンダーフローを意味するものではなく、タイルごとの発生符号量が平均的

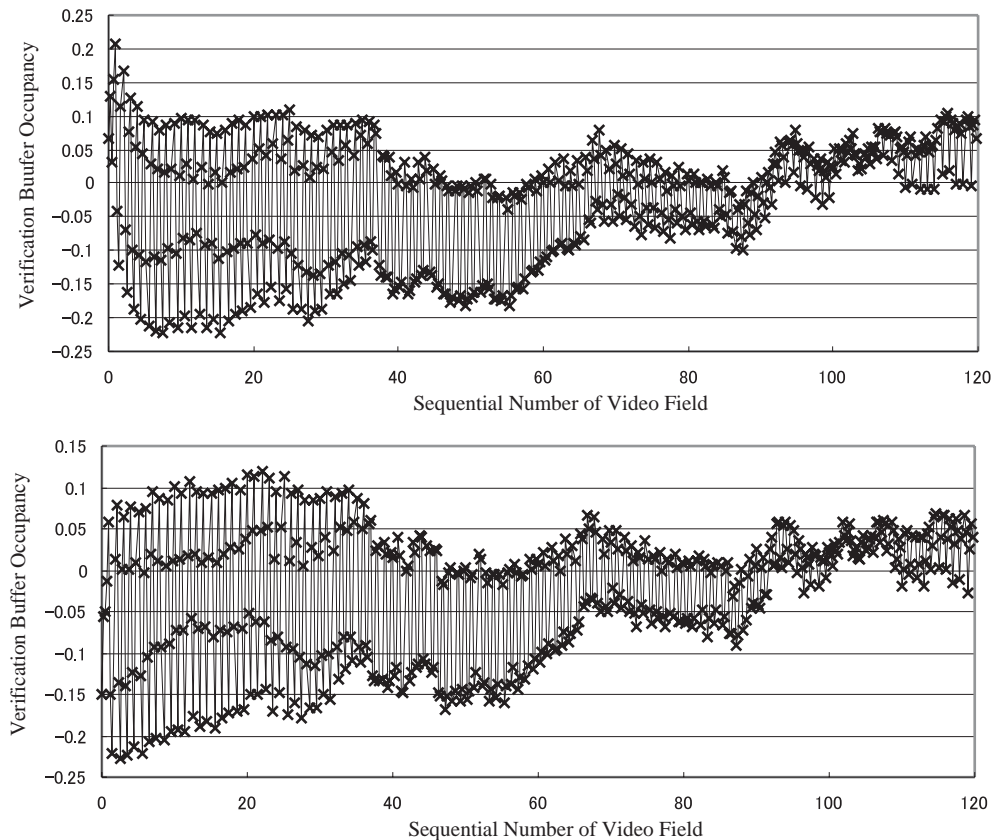


図 3-6: タイル構造 A) における検証バッファ占有量の時間遷移

な符号量に対して変動していることを表す．これを踏まえ，所要のバッファサイズ L_V は式 (3.15) により算出できる． D_B はバッファサイズの不足分を L_T で正規化した値である．つまり D_B は，図 3-6，図 3-7 における縦軸の最大値と最小値の差を表す．入力画像と符号化レートの組合せごとに， L_V を計算した結果を表 3-8 にまとめる．参考までにタイル分割を行わずに提案レート制御を適用した場合の結果を合せて示した（表中 No tiling）．表中 L_V は B_W による正規化値により示した．本章で扱う 3 種類のテスト画像とは別の素材に対する実験結果も踏まえ，同バッファサイズの正規化値としては，タイル構造 A)，B)，No tiling に対してそれぞれ，0.5，0.4，1.2 で十分であることを確認した．エンコーダおよびデコーダにて消費されるバッファリング時間の合計は，所要バッファサイズに比例して増加し，具体的には式 (3.16) により算出可能である． T_W はバッファリング時間の合計（単位は秒）であり， bit_rate は符号化レート（単位は bit/sec）を示す．同計算式が成立する根拠を付録に示す．

$$L_V = (1.0 + D_B) \times L_T \quad (3.15)$$

$$T_W = L_V / bit_rate \quad (3.16)$$

上記解析を踏まえ，その他の主要な遅延要因を考慮に入れ，トータルのコーデック遅延量

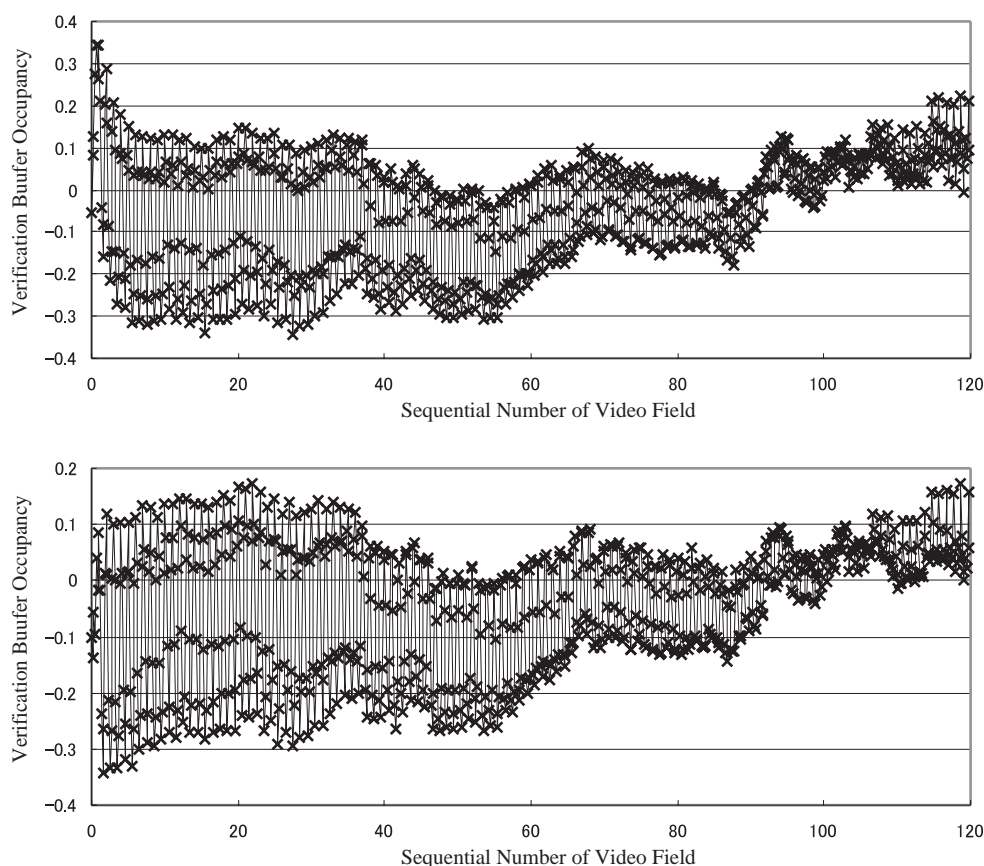


図 3-7: タイル構造 B) における検証バッファ占有量の時間遷移

を推定する．提案方式では，ハードウェア上で符号化処理を行うことを想定しており，符号化処理にかかる時間は瞬時であると仮定している．したがって，コーデック遅延としては，符号化処理時間は省略する．よって，コーデック遅延の主要因としては，バッファリング時間に加え，エンコーダにおける符号化対象画像の入力待ち時間が挙げられる．具体的には，符号化対象画像の先頭画素が入力されてから，最終画素の取得が完了するまでにかかる時間を指す．この考えに基づき，トータルのコーデック遅延量を積算した結果を表 3-9 に示す． W_E はエンコーダにおける符号化対象画像の入力待ち時間を示す．表中，遅延量の算出に際してビデオフィールドの周期は 1080i の HDTV 規格相当である 16.68msec とし，遅延量の単位は msec とした．参考までにタイル分割を行わずに提案レート制御を適用した場合の結果，および文献⁽²²⁾，文献⁽²³⁾によるレート制御を適用した場合の結果を合せて示した．上述のとおり，文献⁽²²⁾，文献⁽²³⁾のコーデック遅延量とともに，画面分割数に依存しないため，結果は各 1 パターンのみ記載した．また従来方式では，符号化処理の単位が画面であることから， W_E は 1 フィールド間隔に相当する．結果よりタイルベースの低遅延符号化構造の導入により，タイル分割の非適用時，および従来方式と比較しコーデック遅延が著しく削減されていることが分かる．また提案レート制御手法は，分割数に依存することなく機能す

表 3-8: 所要バッファサイズの推定値

| Test sequence | Bit-rate (bit/pixel) | Required buffer size | | |
|-----------------|-------------------------|----------------------|------|-----------|
| | | A) | B) | No tiling |
| European market | 0.5 | 0.35 | 0.28 | 1.03 |
| | 0.8 | 0.33 | 0.25 | 1.10 |
| Whale show | 0.5 | 0.45 | 0.36 | 1.08 |
| | 0.8 | 0.43 | 0.34 | 1.10 |
| Street car | 0.5 | 0.44 | 0.34 | 1.15 |
| | 0.8 | 0.46 | 0.34 | 1.08 |

表 3-9: コーデック遅延量の推定値

| Label | W_E [msec] | Buffering delay [msec] | Total delay [msec] |
|--------------------|-----------------|---------------------------|-----------------------|
| A) | 4.17 | 8.34 | 12.51 |
| B) | 2.66 | 6.67 | 9.33 |
| No tiling | 16.68 | 20.01 | 36.69 |
| 文献 ⁽²³⁾ | 16.68 | 20.01 | 36.69 |
| 文献 ⁽²²⁾ | 16.68 | 16.68 | 33.36 |

るため、タイル構造の選択肢に幅を与えるが、より小さなタイルサイズの適用が、さらなる遅延量の削減に寄与することが読み取れる。

3.7 プレシントに基づく低遅延符号化への応用

3.7.1 プレシント分割

タイル分割は、タイル境界の DWT 係数の取得に際する折り返し処理によって、不連続性が生じる可能性がある。これは、ビットレートが低くなれば顕著となり、品質の劣化の原因となる⁽¹³⁾。そこで、不連続性が生じない分割方法として、プレシントの利用を検討する。JPEG2000 において、分割画面単位に符号化データを扱うためのツールとしてプレシントが提供されている。プレシントは、符号化ストリーム上でコードブロックの集合体として分割画面（プレシントと呼ぶ）を形成するものである。このため、タイルのように符号化処理単位を細分化することなく、画面全体に対して符号化を適用した上で、符号化ストリームを分割画面ごとに分けて扱えるという特徴を有する。したがって、タイルとは異なりプレシントを設定することで画質に影響を与えることはない。ここでプレシントの本来の目的とは外れるものの、入力画像の取得順序に沿ってプレシントを割り当てることにより、各プレシントを生成するのに必要な画像エリアが得られた時点で符号化を開始す

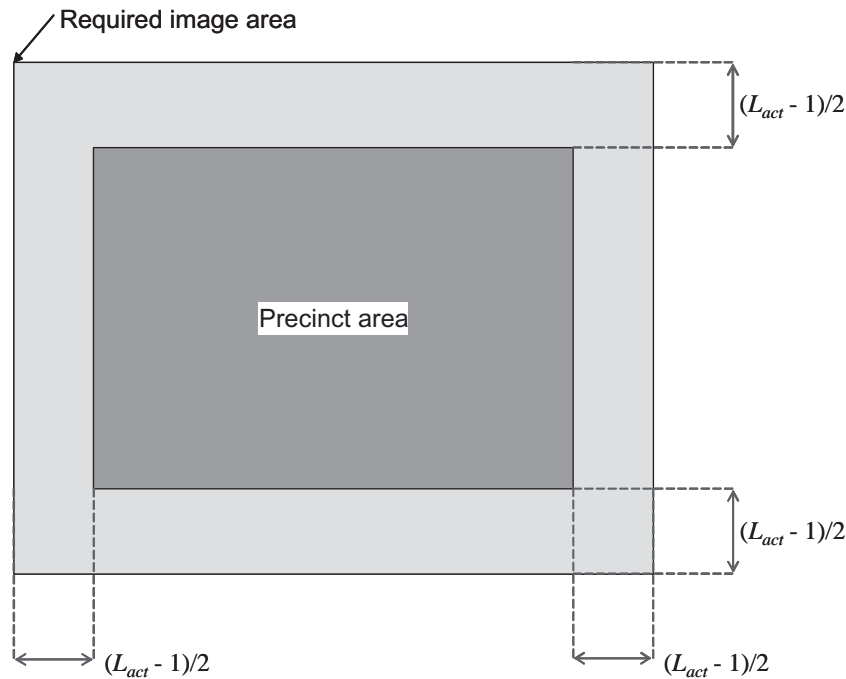


図 3-8: プレシントの生成に必要な画像エリア

ることが可能となり，タイルと同様に符号化処理遅延の削減に有効である⁽²⁷⁾．プレシントを生成するのに必要な画像エリアは，図 3-8 に示すとおりプレシントのエリアと DWT (Discrete Wavelet Transform) の有効タップ長 L_{act} により一意に定まる．有効タップ長とは，サブバンド分割後の最低域成分に属する 1 係数を符号化するために参照すべき画素数を指す．DWT フィルタのタップ数，DWT のレベル数をそれぞれ， m_{tap} ， n_L とおくと，有効タップ長 L_{act} は式 (3-17) により算出可能である．DWT フィルタのタップ数が 9 で，DWT のレベル数が 3 である場合を例にとると，プレシントのエリアに加えて，上端と下端にそれぞれ 28 ライン分の画像データを取得できれば符号化を開始できることになる．上記有効タップ長の制約は，デコーダ処理における DWT 逆変換に対しても同様に影響を与え，あるプレシントの復号結果を得るためには，同一ビデオフィールド内で隣接するプレシントの符号化ストリームが必要となる．

$$L_{act} = (2^{n_L} - 1) \times (m_{tap} - 1) + 1 \quad (3-17)$$

以上より JPEG2000 を用いた低遅延符号化のためのアプローチとして，タイルとプレシントの 2 通りが考えられるが，タイルを利用する場合，特に低ビットレートにおいてはタイル化歪みと呼ばれる画面分割処理に特有の品質劣化を伴う可能性がある⁽¹³⁾．プレシントはこの種の歪みを伴わないものの，プレシントサイズが水平，垂直ともに 2 のべき乗で表現可能な値に限定されている．ここでプレシントサイズはサブバンド分割レベルごとに異なる値を設定可能であるが，画像空間との対応をとる目的から，サブバンド分割レベルが

表 3-10: 対象とするプレシメント構造

| Label | Horizontal size | Vertical size | (x_0, y_0) | <i>prec_cnt</i> |
|-------|-----------------|---------------|--------------|-----------------|
| C) | 2048 | 128 | (0,100) | 5 |
| D) | 2048 | 64 | (0,100) | 9 |

低周波の方向に1つ下がるごとに、元のサイズを2で除した値を設定するのが一般的である。これに対し、あるサブバンド分割レベルで選択可能なコードブロックサイズは当該レベルのプレシメントサイズを超えないよう制限されるため、コードブロック数の増加により効率低下を招く可能性がある。

3.7.2 提案レート制御手法の適用

提案レート制御手法は、3.5. で導入した内容において、プレシメントをタイルとして扱うことにより、プレシメントベースの低遅延符号化においても同様に実装可能である。プレシメントベースの低遅延符号化における、提案方式の有効性を評価する目的から符号化実験を行った。符号化条件は3.6.1に従うこととし、評価対象とするプレシメント構造は、表3-10に示す2パターンとした。 (x_0, y_0) はリファレンスグリッドにおける画像領域の左上端の座標を指し、符号化開始待ち時間の低減を図る目的から、ビデオフィールド内で先頭プレシメントの符号化対象エリアを小さくするために、いずれの構造においても(0,100)に設定した。*prec_cnt* はビデオフィールドを構成するプレシメントの総数を示す。各サブバンド分割レベルにおけるプレシメントサイズは、画像空間におけるプレシメントエリアとの対応がとれるよう設定した。一例として、プレシメント構造C)における有効画像領域に対するプレシメントのマッピングを図3-9に示す。

まずプレシメントベースの低遅延符号化構造に提案レート制御手法を適用した際のPSNRのシーケンス平均を表3-11に示す。比較のため、従来方式として文献⁽²²⁾、文献⁽²³⁾によるPSNRのシーケンス平均を表3-12にまとめた。従来方式との比較については、表3-6、表3-7で確認した結果とほぼ同様であるといえる。

表3-6、表3-7との比較より、分割処理を非適用とした場合に比べて、高々0.1dB程度の効率低下が確認できる。非分割に対しての効率低下要因として、メインヘッダーにてプレシメント分割条件を記述するためのCOC (Coding style component) マーカを伴い、サイド情報が増加した点が挙げられる。また分割数が増えるにつれて効率が低下する原因は、前述のコードブロックサイズの制約によるものであり、画像空間でのプレシメントサイズが小さいほど、ピクチャを構成するコードブロックの総数が増える結果、効率低下を招いたものである。各プレシメントに適用される np の値を観測した結果、タイルベースの場合と同様に先頭ビデオフィールドを除いて、常に3に保持されており、期待する効果が得られていることを確認した。次に表3-8と同様に、入力画像と符号化レートの組合せごとに、所要のバッファサイズ L_V を算出した結果を表3-13に示す。

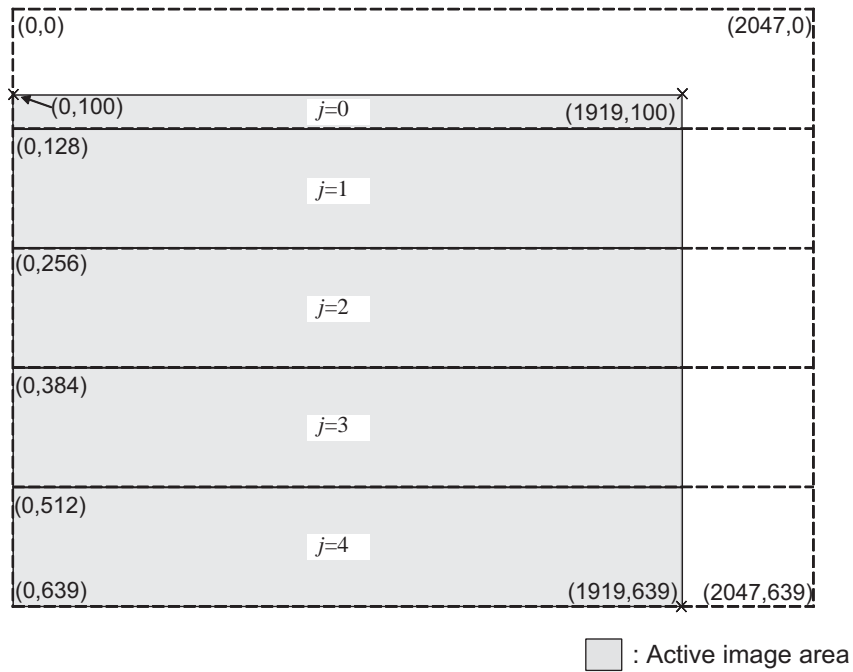


図 3-9: プレシント構造 C) のレイアウト

表中 L_V は B_W による正規化値により示した．表 3-13 より，他のテスト画像を用いた実験結果も踏まえ，同バッファサイズの正規化値としては，プレシント構造 C)，D) に対してそれぞれ，0.5，0.4 で十分であることを確認した．同結果を受けて表 3-9 と同様に，トータルのコーデック遅延量を算出した結果を表 3-14 に示す．エンコーダにおける符号化対象画像の入力待ち時間 W_E は，ビデオフィールド内で先頭のプレシントを生成するのに必要な有効画像エリアが 56 ラインであることから，同ライン数の取得に要する時間として算出した．またデコーダにおいて，上述の有効タップ長の制約から，プレシントの復号処理を開始するためには，当該プレシントの取得に加えて，ビデオフィールド内で後続するプレシントの符号化ストリームが取得されている必要があり，これに伴う待ち時間を表中 W_D

表 3-11: 提案方式による PSNR の結果

| Test sequence | Bit-rate (bit/pixel) | PSNR [dB] | |
|-----------------|----------------------|-----------|-------|
| | | C) | D) |
| European market | 0.5 | 35.90 | 35.82 |
| | 0.8 | 38.05 | 38.00 |
| Whale show | 0.5 | 31.09 | 31.01 |
| | 0.8 | 34.27 | 34.21 |
| Street car | 0.5 | 33.44 | 33.38 |
| | 0.8 | 35.71 | 35.66 |

表 3-12: 従来方式による PSNR の結果

| Test sequence | Bit-rate (bit/pixel) | PSNR [dB] | | | |
|-----------------|----------------------|--------------------|-------|--------------------|-------|
| | | 文献 ⁽²³⁾ | | 文献 ⁽²²⁾ | |
| | | C) | D) | C) | D) |
| European market | 0.5 | 35.91 | 35.86 | 36.04 | 35.98 |
| | 0.8 | 38.06 | 38.01 | 38.21 | 38.17 |
| Whale show | 0.5 | 31.11 | 31.03 | 31.31 | 31.27 |
| | 0.8 | 34.29 | 34.22 | 34.47 | 34.42 |
| Street car | 0.5 | 33.47 | 33.44 | 33.63 | 33.60 |
| | 0.8 | 35.71 | 35.68 | 36.01 | 35.98 |

表 3-13: 所要バッファサイズの推定値

| Test sequence | Bit-rate (bit/pixel) | Required buffer size | |
|-----------------|----------------------|----------------------|------|
| | | C) | D) |
| European market | 0.5 | 0.42 | 0.26 |
| | 0.8 | 0.43 | 0.23 |
| Whale show | 0.5 | 0.46 | 0.33 |
| | 0.8 | 0.46 | 0.32 |
| Street car | 0.5 | 0.45 | 0.34 |
| | 0.8 | 0.48 | 0.37 |

表 3-14: コーデック遅延量の推定値

| Label | W_E [msec] | Buffering delay [msec] | W_D [msec] | Total delay [msec] |
|--------------------|--------------|------------------------|--------------|--------------------|
| C) | 1.72 | 8.34 | 3.33 | 13.39 |
| D) | 1.72 | 6.67 | 1.85 | 10.24 |
| 文献 ⁽²³⁾ | 16.68 | 20.01 | 0 | 36.69 |
| 文献 ⁽²²⁾ | 16.68 | 16.68 | 0 | 33.36 |

として計上した。比較のため文献⁽²²⁾、文献⁽²³⁾によるレート制御を適用した場合の結果を合せて示した。従来方式では、デコーダ処理の単位を画面で扱っているため、 W_D の計上は不要である。結果より、コーデック遅延量を低減する上で、プレシントベースの低遅延符号化構造の導入は、タイルベースと同様に有効であることを確認した。加えて、プレシント構造の選択について、タイル構造の選択と全く同様に、より小さな分割単位の選択が遅延量の削減に効果的であることを確認した。

結論として、低遅延符号化構造と提案レート制御手法の併用は、JPEG2000 符号化の高い符号化効率を維持しつつ、コーデック遅延量の削減を行う上で極めて有効であることを確認した。さらに JPEG2000 符号化では分割画面単位で符号化ストリームを形成するアプローチとして、タイルとプレシントが提供されているが、提案レート制御手法は、同アプローチの選択に依存せず、安定的に機能することを確認した。

3.8 おわりに

本章では、JPEG2000 のタイルそしてプレシントを単位として、分割符号化することによって符号化遅延を低減することができることを示した。その際、各分割領域に対するビット配分が品質に大きく影響を与えることを示した。そして動画像として品質を維持するために、Explicit 量子化とポスト量子化を組み合わせる符号量制御方式を提案した。さらに、実験によって提案方式の性能評価を行った。提案方式は、入力画像を非分割で符号化する場合と比較して品質を大きく劣化させることなく、遅延を低減できることを示した。

第4章

品質変動低減のための提案方式

4.1 はじめに

本章では、まず動画像符号化の際に各フレームに対して適切なビット配分が行われないと、動画像としての品質が劣化することを説明する。さらに、第3章で提案した方式を用いる際に、各フレームの品質が変動しないようなビット配分方法を提案する。具体的には、符号化済みの過去フレームの情報をを用いて、未来の後続フレームの情報を予測する。後続予測によって、フレームを複数バッファリングすることなく低遅延で、当該フレームに割り当てるビット量を決定する。さらに、バッファサイズの制限の下、隣接フレームの量子化ステップの変動に制限をかけることによって、フレーム間の品質の変動を低減する。最後に、提案方式の有効性を実験によって確認する。

4.2 JPEG2000 を用いた動画像符号化とその従来手法

4.2.1 JPEG2000 による動画像符号化の問題点

JPEG2000⁽⁷⁾ は、高い符号化性能と符号化ストリームのスケーラブル性を兼ね備えており、特に高解像度の映像圧縮に関して有効である⁽⁸⁾。

第2章で述べたように、JPEG2000 はイントラ符号化であり、各フレームが独立に符号化されるため、動画像符号化に応用すると以下のような長所がある。

1. 符号化、復号遅延が少ない。
2. ランダムアクセスが可能。
3. 編集上の利用性が高い。
4. 無線伝送上でのエラー耐性が強い。

これらの特徴を受けて、近年 JPEG2000 は静止画にとどまらず、動画像の符号化にも応用されてきている。例えば、デジタルシネマの符号化方式として JPEG2000 が利用されることが決まっている⁽⁹⁾。また、第3章で述べた放送局間の伝送素材や医用画像の符号化に用いられたり、遠隔監視に用いられたりしている。

一方で、JPEG2000 を単純に動画像へ適用すると主にビット配分の弊害に起因した品質劣

化の発生が懸念される⁽²⁴⁾,⁽²³⁾。まず, R-D 最適化によるビット割当てに起因する量子化歪みの分布の偏りが挙げられる。それによって, 動画像の主観品質が劣化するおそれがある。この問題については, 第3章で検討したため本章では省略する。

また, 動画像ではシーケンスを構成する各フレームの特性が変化するため, 各フレームで複雑さが異なる。そのため, 各フレームを独立に符号化すると, 品質にばらつきが生じ, 動画像として品質が変動する。これは, シーンごとの変化が激しいシーケンスで特に顕著に現れる。映像の品質の尺度として, シーケンス全体の平均 PSNR が挙げられるが, PSNR と同時にフレーム間の品質の変動についても考慮する必要がある。人間の目は, 各フレームの品質が激しく変動するよりも, 一定の品質をとる映像を好む。したがって, 複雑なシーンにはビットを多めに割り当て, 簡単なシーンにはビットを少なめに割り当てるといったように, フレーム間の品質が安定するような符号量制御が必要である。各フレームに等ビットを与えて符号化した際の, フレーム間の品質の差の例を図 4-1, 図 4-2, 図 4-3, 図 4-4 に示す。図 4-1 と図 4-2 は二つのフレームの原画像であり, 図 4-3 と図 4-4 はそれぞれ同条件で 0.5bpp で符号化したものである。各フレームについて原画像と比較すると, 同じ条件で符号化しているにも関わらず, 図 4-3 よりも図 4-4 の方がブロック塀の部分を中心に品質の劣化が激しいことがわかる。これは, 図 4-1 よりも図 4-2 の方が複雑な画像だったにも関わらず, 同じビットを割り当てて符号化したためである。

JPEG2000 を動画像符号化に応用するためには以上のような問題点を改善する必要がある。

4.2.2 CBR と VBR

MPEG に代表される動画像圧縮では, 符号量制御方式として, CBR(Constant Bit Rate) 方式と VBR(Variable Bit Rate) 方式が定められている。CBR 方式は, 単位時間あたりに伝送されるビット量が一定である。各フレームの複雑が異なるので, エンコーダの出力ビットレートは変動する。CBR で伝送するためには, バッファでビットレートを一定にする必要がある。VBR 方式は, 単位時間あたりに伝送されるビット量を可変にすることができる。CBR 方式では, 各フレームの品質が変動するおそれがあるが, VBR 方式は各フレームが一定の品質をとることが可能である。本研究では, ピクチャ単位としてフレーム/フィールドを取り上げ, 各ピクチャに等ビットを与える方式を CBR 方式, 各ピクチャに可変ビットを割り当てる方式を VBR 方式とする。

4.2.3 従来手法

符号量制御方式として VBR 方式をとり, シーケンスを構成する各フレームにそれぞれ異なるビットを割り当てることによって, シーケンスの品質を安定させることが可能である。品質変動低減のための, MPEG-2 を用いた可変ビットレートでの符号化については先行研究が多く報告されている⁽²⁸⁾,⁽²⁹⁾,⁽³⁰⁾。これらの多くが, Rate と量子化ステップ Q の関係, Distortion と Q の関係を求め, Q を基準として品質が大きく変動しないような方式を取っ



図 4-1: foreman1 原画像



図 4-2: foreman2 原画像



図 4-3: foreman1 0.5bpp で符号化



図 4-4: foreman2 0.5bpp で符号化

ている．品質変動を厳密に低減するためには，一度目に全フレームを符号化し，シーケンス全体の Rate と Distortion の関係を確認し，二回目の符号化で各フレームに適切なビット量を割り当てる 2-pass 符号化が有効である．しかし 2-pass 符号化は，一度全シーケンスを符号化しなければならないため，符号化遅延が大きくなるという欠点がある．そのため，リアルタイムアプリケーションには不適切である．一度の符号化で，各フレームに適切にビットを割り当てる 1-pass 符号化の研究も行われている．これらは，バッファの占有量や当該フレームの複雑さなどを基準として，割り当てビットを決定している．

一方で，JPEG2000 の符号化処理に特化した可変ビット配分についてはいまだ十分な研究が成されていない．

JPEG2000 を用いた可変ビットレート符号化について既に報告されているものとして，文献⁽³¹⁾が複数フレームを LeakyBucket に蓄積し，蓄積されたフレームのデータを踏まえて品

品質変動が小さくなるような符号化方法を取っている。フレームを大量に蓄積することによって、シーケンスの品質変動は低減されるが、伝送遅延の低減を考える上では、この方法は有効であるとはいえない。

以上の過去研究を踏まえて本章では、低遅延でフレーム間の品質の変動を低減し、なおかつその際の各フレームの量子化歪の分布が偏らないような符号量制御を行う。具体的には符号化遅延を低減するために、符号化前に複数フレームを蓄積することなく、各フレームの R-D 特性を利用して後続フレームの画像の特徴を予測する。その上で、連続するシーケンスの品質の変化が滑らかになりかつ、量子化歪の分散が偏らないような符号量制御方式を提案する。

4.3 提案方式

4.3.1 想定環境

本章では、第3章と同様に映像入力として、1080i HDTV 信号を想定し、JPEG2000 符号化がビデオフィールドまたはタイル単位に連続的に適用される。インデックス i, j をタイルの識別に用いる。あるタイルに対して、 i は時間的な順序を示し、 j は当該ビデオフィールド内での空間的な順序を 0 から $(tile_cnt - 1)$ の連番で示す。 $tile_cnt$ はビデオフィールドを構成するタイルの総数である。また本章では、 n, k をビデオフィールドのインデックスとして用いることとする。以上の想定環境を考慮して、本章ではこれ以降、動画像シーケンスの構成要素をフレームではなくフィールドと表現する。

4.3.2 フィールド間の品質変動の低減

第3章では、量子化歪みの分布の偏りを制御するために、フィールド間の切捨てビット数を制御していたが、各フィールドに等ビットを与えることを前提としていた。第3章の方式によって量子化歪みの偏りを抑えられるが、シーケンスの複雑度が激しく変化する場合は、各フィールドに等ビットを割り当てると各フィールドの品質にばらつきが生じる。そこで、第3章で提案した方式を応用してフィールド間の品質が緩やかに変化し、かつ量子化歪の分布を制御する方式を提案する。本方式では、 R と量子化ステップ Q の関係を基準として、量子化歪の偏りを制御しつつ、品質変動を低減するような方式を用いる。

厳密に一定の品質で符号化するのであれば D 一定で全シーケンスを符号化を行うのが理想的である。また、シーケンス全体で平均 MSE を最低にするためには、目標ビット量を満たす一定 R を用いて全シーケンスを符号化すればよい。すなわち R-D 最適化に基づいてビット切捨て点を決定することによって全体としての品質は最高となる。

これに対して、第3章で述べたように量子化歪みの分布の偏りを解消するために、Explicit 量子化とポスト量子化を併用してビットを切り捨てる必要がある。 Q を一定で Explicit 量子化を行う場合、各ビットプレーンでビット切捨て点が同一となる。したがって、目標符号量が等しくても厳密には、Rate と Distortion の関係が最適とはならない。しかし、 Q は DWT

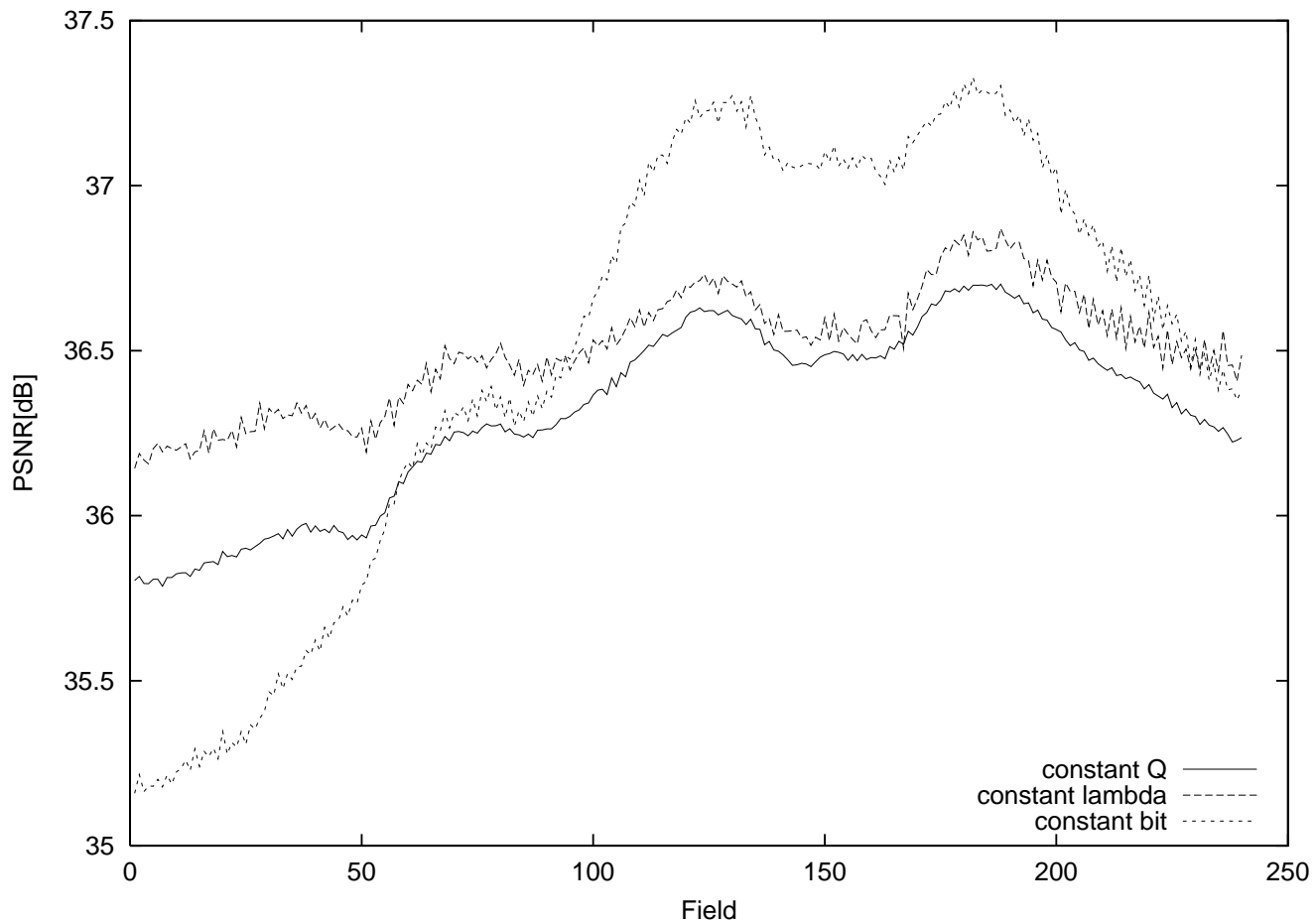


図 4-5: Q 一定時と 一定時の品質変動 (Streetcar)

係数を量子化する粗さであるため、歪 D と密接に関係している。したがって Q を一定に符号化することによって、品質変動を低減するための貢献となりうる。 Q 一定で Explicit 量子化を行った場合、一定時に準ずる安定した品質変動が得られる。

同じビット量という条件の下、 Q 一定で Explicit 量子化した場合と、一定で R-D 最適化に基づくポスト量子化を行った場合の各フィールドの品質の変化を Streetcar, Whaleshow のそれぞれについて、図 4-5, 図 4-6 に示す。図 4-5, 図 4-6 から、 Q 一定で Explicit 量子化を行った場合に、一定の R-D 最適化の場合と比較して、品質がやや低下するが、品質変動が抑えられている様子が見られる。そこで提案方式では、フィールド間の量子化歪の偏りを抑制するために、ではなく Q をパラメータとして、各フィールドを符号化する方式をとる。

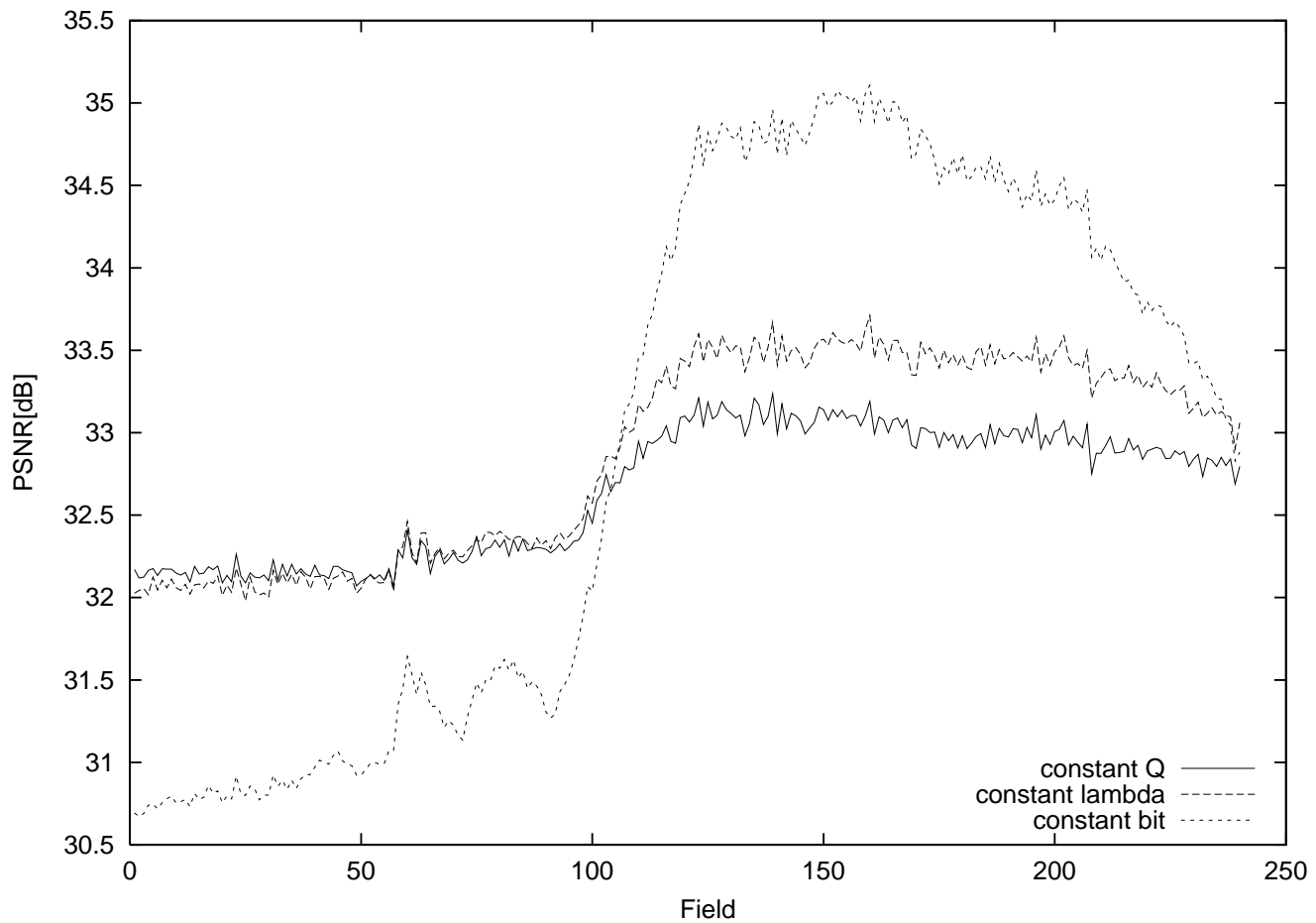


図 4-6: Q 一定時と 一定時の品質変動 (Whaleshow)

4.3.3 R-Q 曲線の導出

まず、量子化歪の偏りを防ぐため、ポスト量子化によって廃棄する符号化パス数 n_p は、全コードブロックで同一とする。そのため、第3章と同様に R と Q の関係を以下のように表す。さらに、JPEG2000 の符号化処理の中で、複数組の R と量子化ステップ Q の組み合わせを得て、それから R と Q の関係を式 (4.4) のように近似する。

$$\mathit{delta}[\mathit{band}] = \mathit{dqt}[\mathit{band}] \times \mathit{cq}(i) \quad (4.1)$$

$$R = f_j(Q) \quad (4.2)$$

$$Q = cq(i) \times 2^{npl_i}, R = r(i, npl_i \times 3) \quad (4.3)$$

$$f_j(Q) = \alpha_j \times Q^{\beta_j} + C_j \quad (4.4)$$

i : タイルの時間的な順序

j : 当該フィールド内での空間的なタイルの順序

第3章では, delta[band] は当該フィールドの先頭タイルの符号化に先立ち, 直前ピクチャの結果をもとに更新している. つまり, Q は各フィールドの先頭タイルの符号化に先立ち更新される. すなわち第3章の方式では, 後続のフィールドを予測するという概念は無い. これに対して, 本章では符号化済みフィールドをもとに後続フィールドを予測し, その結果に基づいて Q を更新する. 後続予測のために, 各フィールドの R と Q の関係を表した R-Q 曲線を各フィールドの特性として複雑さを判断する.

4.3.4 R-Q 曲線の予測

- 式 (4.4) で近似した R-Q 曲線の品質変動低減への適用を考える. 直前フィールドが符号化終了した時点で, 直前フィールドの全てのタイルについての R と Q の関係が得られる. 直前フィールドの R と Q の関係を式 (4.5) のように求める.

$$R = \sum_{j=0}^{tile_{ent}-1} f_j(Q) \quad (4.5)$$

- 量子化ステップ Q は, フィールド内の全てのタイルで共通なので, 式 (4.5) から, 直前フィールド全体の R と Q の関係が求められる. 第 n フィールド全体の R と Q の関係を表す関数 $f_n(Q)$ を式 (4.6) に示す.

$$f_n(Q) = \binom{n}{k} \times Q^{\beta_k} + C \quad (4.6)$$

- 式 (4.6) では, $\binom{n}{k}$, β_k を時間方向の相関を考慮してフィールド数 n の関数として考える. シーケンスを符号化していく中で, 符号化済みフィールドの R-Q 曲線の係数 β_k , C を用いて, 後続フィールドの R-Q 特性を予測する. (C は定数)
- 複数の過去フィールドについて式 (4.6) のように R と Q の関係を求める. 過去フィールドの R と Q の関係から, 未来の後続フィールドについて, R と Q の関係を予測する. 第 n フィールドまで符号化した時点で, 予測から求めた第 k フィールド ($k > n$) の R-Q 曲線における β_k , C をそれぞれ $\beta_n(k)$, $C_n(k)$ と表すこととする. すると, 第 k

フィールドの R と Q の関係 $f_{n(k)}(Q)$ は、 $r_n(k)$ 、 $q_n(k)$ 、 C を用いて式 (4.7) のように求められる。

$$f_{n(k)}(Q) = r_n(k) \times Q^{q_n(k)} + C \quad (4.7)$$

4.3.5 R-Q 曲線の予測に基づくビット配分

次に、各フィールドに割り当てる符号量を計算する。提案方式では、当該フィールドから後続 N フィールド分に割り当てる符号量をビット過不足を考慮して計算する。具体的には、当該フィールド n から N 枚のフィールドについて割り当てる平均符号量 $\overline{R(n)}$ を式 (4.8) により算出する。 B_W は 1 フィールドに割り当てられる平均符号量である。

$$\overline{R(n)} = \frac{(N+n) \times B_W - \sum_{i=0}^{n-1} S(i)}{N} \quad (4.8)$$

$S(i)$: i 番目のフィールドにて発生した符号化ビット数

R と Q の関係は、式 (4.7) によって表されるので、第 n フィールドから N フィールド分のビット量 R は Q を用いて $\sum_{i=n}^{n+N-1} \{ r_n(i) \times Q^{q_n(i)} + C \}$ と表すことができる。そこで、第 n フィールドから N フィールド分について、

$$\sum_{i=n}^{n+N-1} \{ r_n(i) \times Q^{q_n(i)} + C \} = \overline{R(n)} \times N \quad (4.9)$$

を満たすような Q を $Q(n)$ として、第 n フィールドの符号化に用いる。

4.3.6 品質の変動幅の制御

前節で符号化済みフィールドの情報から、後続フィールドの特性を予測したが、これはあくまでも後続の R-Q 曲線の変化の傾向を把握するためのものであって、正確な予測は極めて困難である。したがって、後続予測をすることによって、逆に品質の変動が大きくなる可能性も考えられる。そこで、過去フィールドの符号化に用いた Q の値を利用して、当該フィールドに与えられる Q の値を調整する。まず、式 (4.9) から求めた第 n フィールドの符号化に用いる Q を $Q(n)$ とおく。次に、当該フィールドを含まない直前 $M-1$ フィールドで用いた Q 値の分散を求め、この値を $V1$ とする。さらに、 $Q(n)$ を含めて、当該フィールドから直前 M フィールドの Q 値の分散 $V2$ を求める。

$$V1 = \frac{1}{M-1} \sum_{i=n-M}^{n-1} \{ Q(i) - \overline{Q_{M-1}(n-1)} \}^2 \quad (4.10)$$

$$V2 = \frac{1}{M} \sum_{i=n-M}^n \{ Q(i) - \overline{Q_M(n)} \}^2 \quad (4.11)$$

V1 : 直前 M-1 フィールドの符号化で用いた Q の分散

V2 : 当該フィールドを含む M フィールド分の Q の分散

$\overline{Q_{M-1}(n-1)}$: 第 n-1 フィールドの直前 M-1 フィールド
の Q の平均値

$\overline{Q_M(n)}$: 第 n フィールドの直前 M フィールド
の Q の平均値

そして、V1 と V2 を比較する。式 (4.9) で計算した $Q(n)$ を用いて符号化することによって、Q の分散が急激に増加するようであれば、分散を考慮して $Q(n)$ を再計算する。具体的には、閾値 A を設定し、式 (4.12) を満たすような $Q'(n)$ を求める。

・ $V2 \leq A \times V1$ の場合

$$Q'(n) = Q(n)$$

・ $V2 > A \times V1$ の場合

$$\frac{1}{M} \left\{ \sum_{i=n-M}^{n-1} \{Q(i) - \overline{Q_{M-1}(n-1)}\}^2 + \{Q'(n) - \overline{Q_{M-1}(n-1)}\}^2 \right\} = A \times V1 \quad (4.12)$$

4.3.7 バッファによる符号量制御

提案方式では、後続フィールドの R-D 曲線の予測による画像複雑度の把握と符号化済みフィールドにおけるビット過不足、さらに隣接フィールドの品質差を考慮して、各フィールドを符号化を行っている。しかし、この場合各フィールドに割り当てられる符号量の変動は避けられないため、符号量の変動をバッファで吸収する必要がある。そこで、指定されたバッファサイズの下、符号量を割り当てることを検討する。当該第 n フィールドのバッファ占有量から、今後 N フィールドに関してバッファがアンダーフローもしくはオーバーフローしないような、当該フィールドに割り当て可能な最大ビット量 $R_{max}(n)$ と最小ビット量 $R_{min}(n)$ を求める。 $R_{max}(n)$ 、 $R_{min}(n)$ はフィールドごとに更新される。具体的には、まずバッファサイズを V 、初期遅延量を V_o とする。当該第 n フィールドのバッファ占有量を $V(n)$ とすると、第 n フィールドに割り当てられるビット量の最大値 $R_{max}(n)$ と最小値 $R_{min}(n)$ は式 (4.13)、式 (4.14) のように表される。

$$R_{max}(n) = \frac{V_o + (N + n) \times R - \sum_{i=0}^{n-1} S(i)}{N} \quad (4.13)$$

$$R_{min}(n) = \frac{V_o + (N + n) \times R - \sum_{i=0}^{n-1} S(i) - V}{N} \quad (4.14)$$

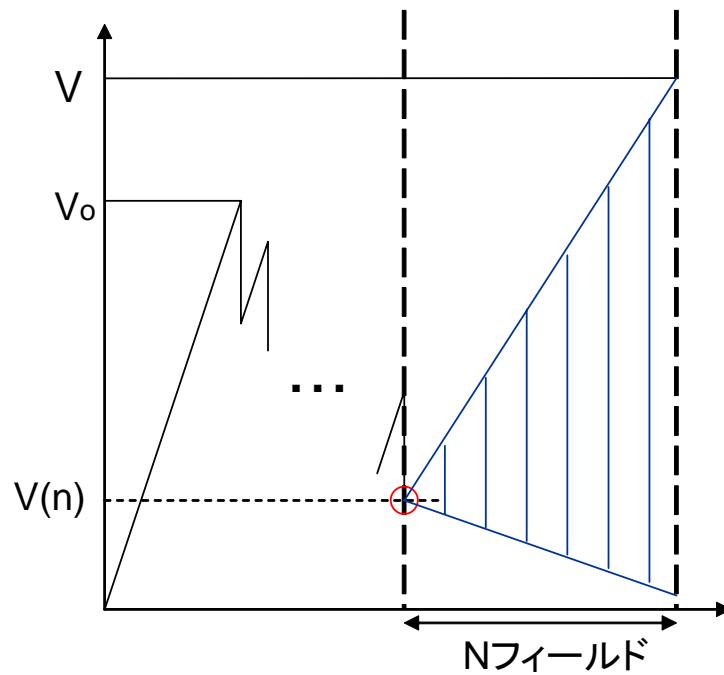


図 4-7: バッファ占有量の遷移の制限

N : 符号量制御に用いるウィンドウサイズ

この定められた範囲の中で、当該フィールドにビットが割り当てられる。式(4.12)で求めた $Q'(n)$ に相当するビット量を $R(n)$ とすると、当該フィールドに割り当てるビット量 $R'(n)$ は、 $R_{max}(n)$ 、 $R_{min}(n)$ を用いて以下のように決定される。

- $R(n) > R_{max}(n)$ の場合
 $R'(n) = R_{max}(n)$
- $R(n) < R_{min}(n)$ の場合
 $R'(n) = R_{min}(n)$
- $R_{min}(n) \leq R(n) \leq R_{max}(n)$ の場合
 $R'(n) = R(n)$

これらの処理を、バッファの占有量のグラフで表したものを、図 4-7 に示す。

当該フィールドから、ウィンドウ分の N フィールドについて、バッファの占有量の変化が図 4-7 の斜線領域内で変化すれば良い。このように、当該フィールドについてのみ、割当てビット量によってバッファがアンダーフローもしくはオーバーフローしないのが確認するの

表 4-1: 符号化条件

| | |
|---------------|---|
| 入力画像 | 1920×1080 / 60Hz (インターレース) 4:2:2 コンポーネント (8 ビット) |
| 符号化レート | 0.8bit/pixel |
| DWT フィルタ | (9,7) 非可逆フィルタ |
| サブバンド分割レベル | 3 レベル |
| フィールド数 | 480 フィールド |
| バッファサイズ | 10Mbit |
| ウィンドウサイズ N | 60 フィールド |
| Q 分散制御ウィンドウ M | 10 フィールド |

ではなく、当該フィールドから N フィールド分の占有量の変化の領域を定めることで、割当ビットの急激な変化さらには品質の急激な変化を防ぐことができる。

4.3.8 バッファサイズと遅延

次に、バッファサイズについて検討する。各フィールドに可変な符号量を与えるために、MPEG-2 や MPEG-4 では VBV (Video Buffering Verifier) が、H.264 では HRD (Hypothetical Reference Decoder) が定められている^{(6), (32)}。

MPEG-2 のメインプロファイルでは、メインレベルで最大ビットレートが 15Mbps、最大 VBV バッファサイズが 1.835Mbit と定められている。これは 122msec の遅延に相当する。また、メインプロファイルの HDTV レベルでは、最大ビットレートが 50Mbps、最大 VBV バッファサイズが 10Mbit である。これは、200msec の遅延に相当する。

H.264 では、メインプロファイルのレベル 4 (HDTV レベルに相当) で、最大 HRD バッファサイズは 20Mbit とされている。

本研究では、HDTV を主なターゲットと設定しているため、MPEG-2 の HDTV レベルの最大 VBV バッファサイズに合わせて、バッファサイズを 10Mbit と設定する。

4.4 性能評価

提案方式による符号化性能を評価する目的から、符号化実験を行った。今回行った符号化条件を表 4-1 示す。

テスト画像は第 3 章でも使用した ITE 標準画像の Streetcar と EuropeanMarket を用いた。

- Streetcar

第 1 フレームを開始点とした 240 フィールドに対して、
第 240 フィールドが中心となるように同 240 フィールドを繋げた
480 フィールド。

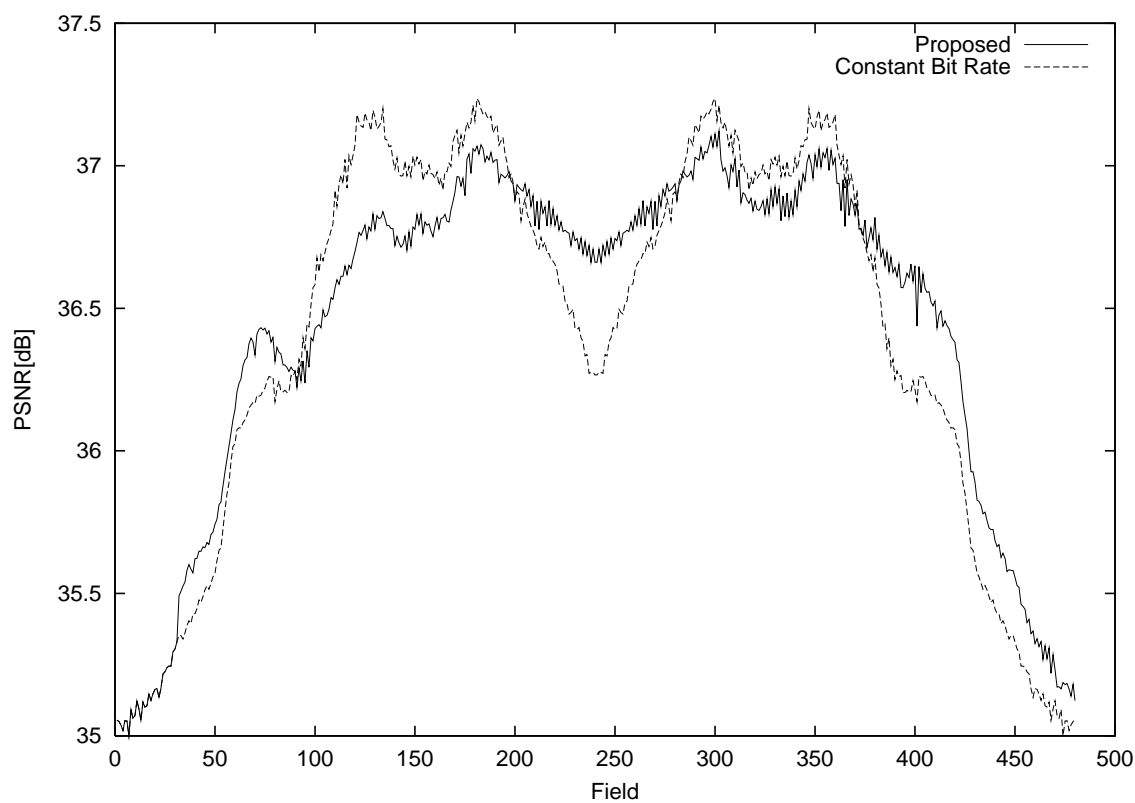


図 4-8: 等ビット配分方式と提案手法の PSNR の比較 -Streetcar- (バッファサイズ 10Mbit)

表 4-2: PSNR の平均と分散 (Streetcar)

| Label | PSNR の平均 | PSNR の分散 |
|-------------------|----------|----------|
| constant bit rate | 36.41 | 0.485 |
| proposed | 36.44 | 0.364 |

- Europeanmarket

第 300 フレームを開始点とした 240 フィールドに対して，
第 240 フィールドが中心となるように同 240 フィールドを繋げた
480 フィールド．

定めたバッファサイズの制限下でのシステムの動作を確認するために，バッファサイズを 10Mbit と設定し，この制限下で，符号量制御を行うこととした．品質変動については，各フィールドに等ビットを配分する方式と提案方式の 2 つの方式で比較を行う．

Streetcar と EuropeanMarket のそれぞれについて，品質変動の比較として，各フィールドでの PSNR の変化の様子を図 4-8，図 4-9 に示す．また，シーケンス全体の PSNR の平均と分散を表 4-2，表 4-3 に示す．

図 4-8，図 4-9 より，等ビット配分時と比較して，提案方式を用いることで，各フィール

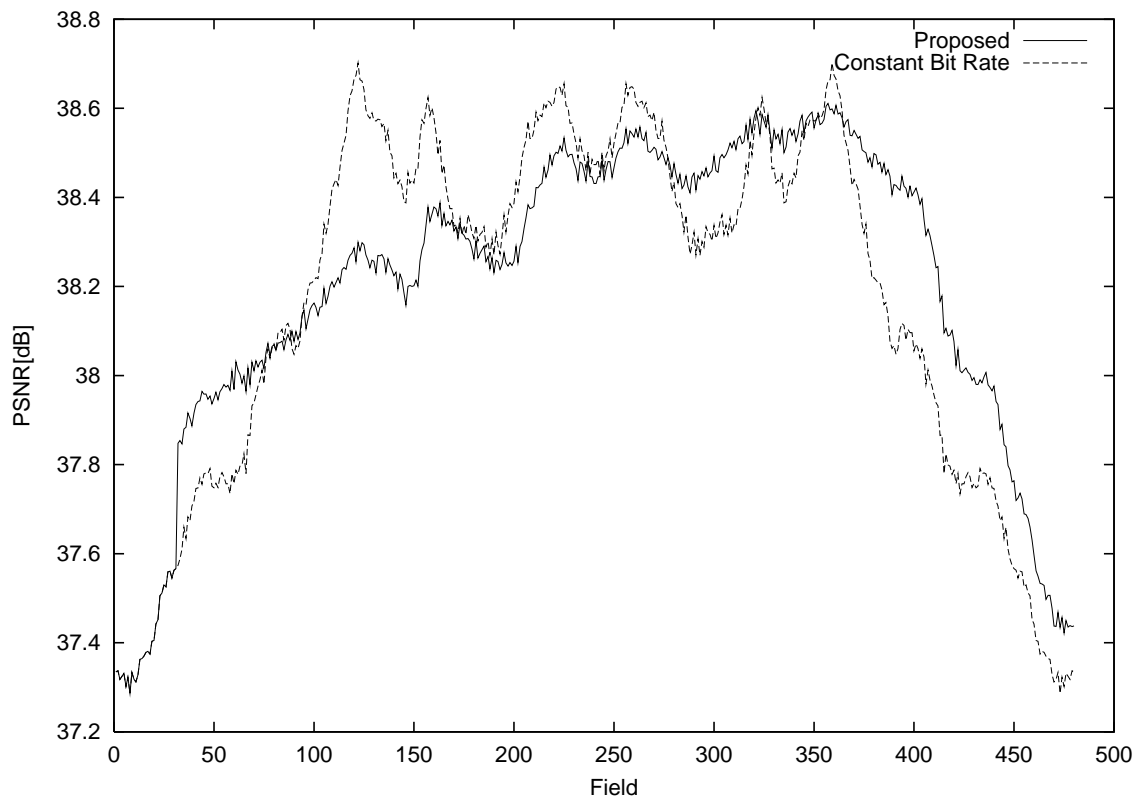


図 4-9: 等ビット配分方式と提案手法の PSNR の比較 -EuropeanMarket-
(バッファサイズ 10Mbit)

ドの PSNR が滑らかに変化している様子が見られる。また、表 4-2、表 4-3 より、提案手法が等ビット配分時と同程度の平均 PSNR の値を保持したままで、PSNR の分散が 70 ~ 80 % に減少している様子が見られる。平均 PSNR が同等なのは、各フィールドに可変ビットを与えながらも、目標符号量に対する過不足を考慮して、ビット割り当てを決定したことによる。また、後続フィールドの特性を予測しつつ、隣接フィールド間の量子化ステップの変化幅を制御したことによって、PSNR の分散が抑えられたと言える。

さらに、その際のバッファの占有量の変化を図 4-10、図 4-11 に示す。

図 4-10、図 4-11 から、各フィールドに可変ビットを与えるため、限られたバッファサイズの下でバッファの占有量に変化している様子が見られる。バッファ占有量が、オーバーフローとアンダーフローの限界の位置まで動かないのは、当該フィールドに対してだけでなく、後

表 4-3: PSNR の平均と分散 (EuropeanMarket)

| Label | PSNR の平均 | PSNR の分散 |
|-------------------|----------|----------|
| constant bit rate | 38.17 | 0.169 |
| proposed | 38.20 | 0.119 |

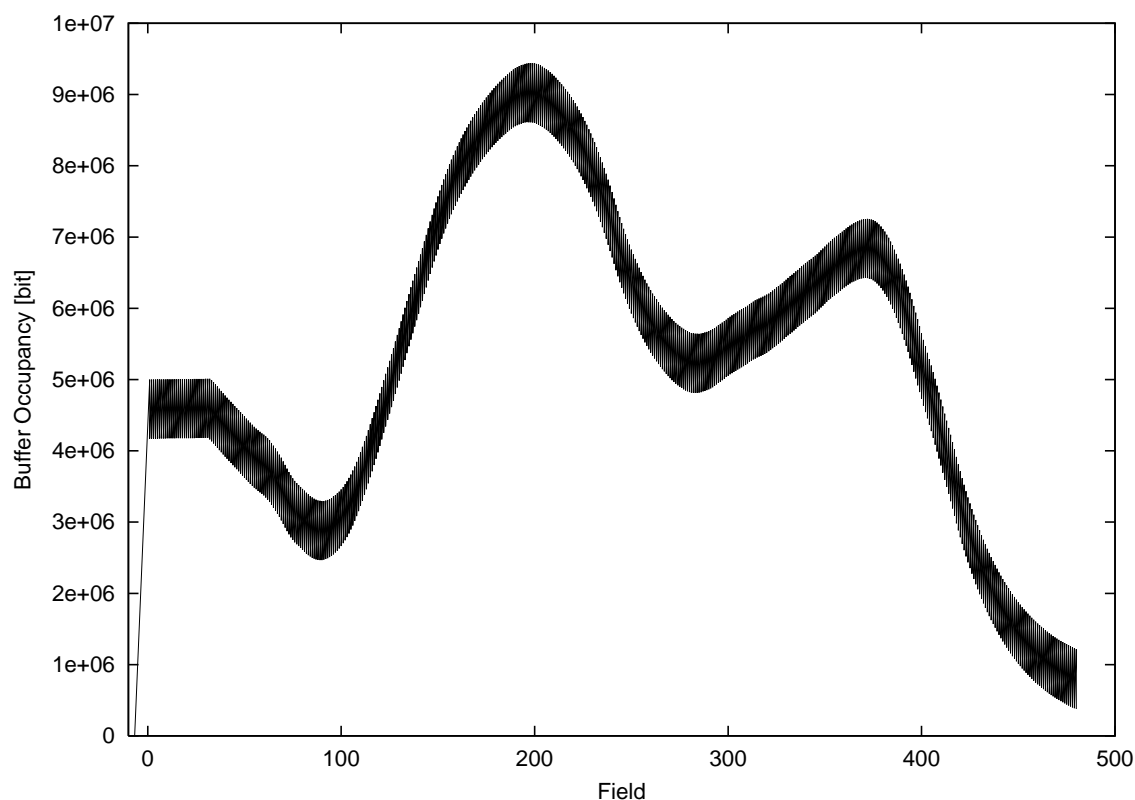


図 4-10: 提案手法によるバッファ占有量の遷移 (Streetcar-
(バッファサイズ 10Mbit))

続 N フィールドを予測して、アンダーフローもしくはオーバーフローしないように割り当てビットを決定した結果である。

次に、提案方式を用いる際の遅延について考察する。本実験では、ビットレートとして 50Mbps を想定してるので、10Mbit の VBV バッファサイズによって生じる遅延は 200msec となる。JPEG2000 を用いることによるコーデック遅延を推定する。各条件におけるコーデック遅延を比較したものを表 4-4 に示す。本方式では、符号化処理にかかる時間は瞬時であると仮定する。したがって、コーデック遅延としては、バッファリング遅延に加えて、エンコーダにおける符号化対象画像の入力待ち時間を考える。対象画像の入力待ち時間は、インターレース 60Hz で符号化することから 16.68msec となる。まず、JPEG2000 を提案方式を用いず、各フィールドに等ビットを与えて符号化した場合、バッファリング遅延は 16.68msec となり、入力待ち時間と合計すると 33.3msec となる。次に、動画像としての品質変動を考慮した提案方式を用いるとバッファリング遅延は 200msec であり、対象画像の入力待ち時間とバッファリング時間を加算すると 216.68msec となる。

提案方式を用いることで、各フィールドに等ビットを配分する場合と比較して、符号化遅延は増加する。しかし、許容される遅延の下、バッファサイズを決定することによって、

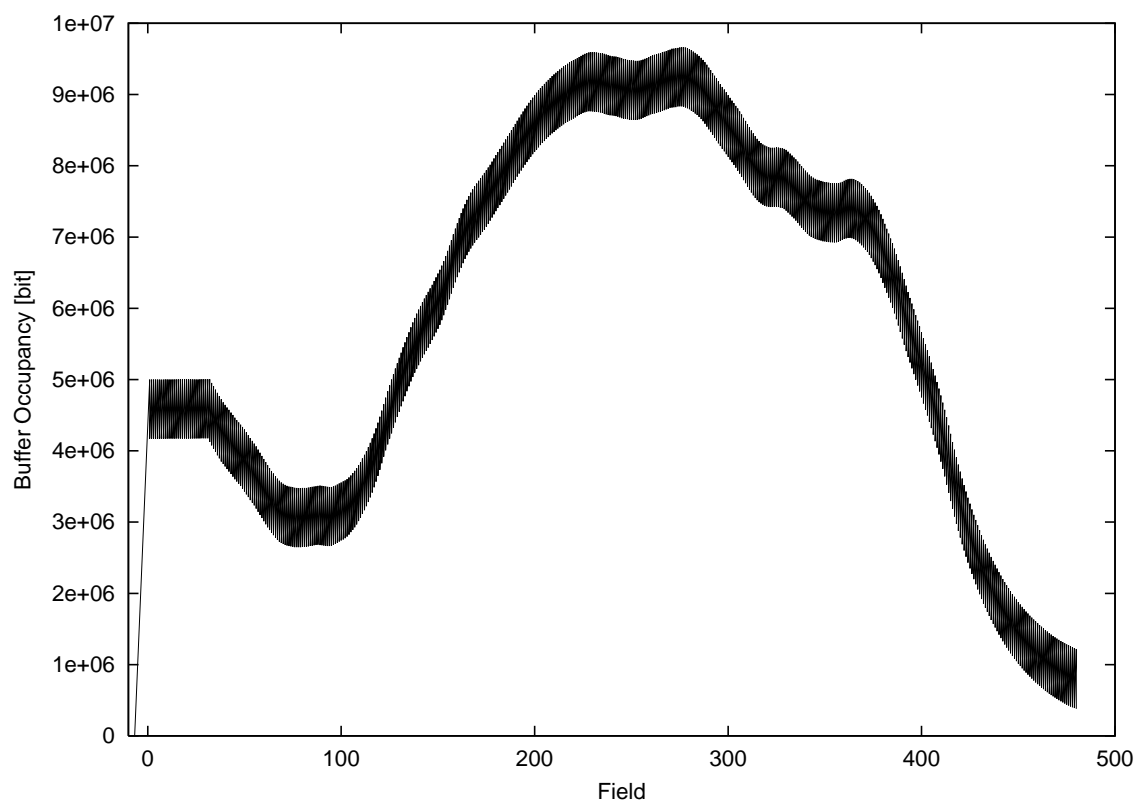


図 4-11: 提案手法によるバッファ占有量の遷移 -EuropeanMarket-
(バッファサイズ 10Mbit)

表 4-4: コーデック遅延量の推定値

| Label | 入力待ち時間 [msec] | バッファリング遅延 [msec] | 合計遅延 [msec] |
|-------------------|------------------|---------------------|----------------|
| Constant bit-rate | 16.68 | 16.68 | 33.3 |
| Proposed | 16.68 | 200 | 216.68 |

提案方式が等ビット配分時と同等の品質を維持しながら、品質の変動を低減できることがわかる。

以上の実験より提案方式は、定めたバッファサイズの下で、品質の変動を低減する効果があることがわかった。提案方式における品質変動はバッファサイズに依存しており、バッファサイズが無限であれば、品質変動はより低減される。

コーデック遅延については、第3章で示したように入力画像を分割して符号化することによって低減することが可能である。したがって、コーデック遅延を低減するためには、分割符号化方式は非常に有効である。しかし、動画としての品質の変動を考慮するためには、本方式のようにバッファサイズを定め、その範囲で遅延を許容することによって品質の変動を

抑えることが可能である。

4.5 おわりに

本章では、まず動画像符号化の際に各フィールドに対して適切なビット配分が行われないと、動画像としての品質が劣化することを説明した。さらに、第3章で提案した方式を用いる際に、各フィールドの品質が変動しないようなビット配分方法を提案した。具体的には、符号化済みの過去フィールドの情報を用いて、未来の後続フィールドの情報を予測する。後続予測によって、フィールドを複数バッファリングすることなく低遅延で、当該フィールドに割り当てるビット量を決定した。さらに、バッファサイズの制限の下、隣接フィールドの量子化ステップの変動に制限をかけることによって、フィールド間の品質の変動を低減した。最後に実験によって、提案方式の性能評価を行った。提案方式は、定められた遅延の下で、各フィールドの品質を劣化させることなく、変動を抑えることができることを明らかにした。

第5章

結論

本章では本論文を総括し、今後の検討課題を述べる。

5.1 総括

本研究は、HDTVの素材伝送を対象とし、JPEG2000の枠組の中で特にレート制御の最適化に着目し、低遅延かつ高効率な圧縮符号化を実現するためのものである。提案技術の導入により、JPEG2000本来の高い符号化性能を維持しつつ、コーデック遅延量の大幅な削減が達成されることを確認した。従来一般的なJPEG2000符号化実装手法は動画像を扱う上で、量子化歪が空間的に偏って分布することに起因して主観品質が劣化するというリスクを抱えていた。これに対し、提案レート制御手法はこの問題を解消できることを示した。また、フレームごとの品質が大きく変動することによって動画としての品質が低下する問題があった。これに対し、提案方式は、許容される範囲でバッファを設定することで動画像全体として品質を劣化することなく、品質の変動を低減することが可能であることを示した。

以下に、本論文の各章の内容を述べる。

第1章では、本研究の背景と目的、および本論文の構成について述べた。

第2章では、本研究で用いる符号化方式であるJPEG2000について説明した。JPEG2000の概要と符号化処理のアルゴリズムとその技術について述べた。特に、分割符号化における分割単位であるタイルとプレシントについて説明を行った。さらに、JPEG2000の動画像フォーマットであるMotion JPEG2000の概要を述べた。

第3章では、JPEG2000のタイルそしてプレシントを単位として、分割符号化することによって符号化遅延を低減することができることを示した。その際、各分割領域に対するビット配分が品質に大きく影響を与えることを示した。そして動画像として品質を維持するために、Explicit量子化とポスト量子化を組み合わせる符号量制御方式を提案した。さらに、実験によって提案方式の性能評価を行った。提案方式は、入力画像を非分割で符号化する場合と比較して品質を大きく劣化させることなく、遅延を低減できることを示した。

第4章では、まず動画像符号化の際に各フレームに対して適切なビット配分が行われないと、動画像としての品質が劣化することを説明した。さらに、第3章で提案した方式を用い

る際に、各フレームの品質が変動しないようなビット配分方法を提案した。具体的には、符号化済みの過去フレームの情報をを用いて、未来の後続フレームの情報を予測する。後続予測によって、フレームを複数バッファリングすることなく低遅延で、当該フレームに割り当てるビット量を決定した。さらに、バッファサイズの制限の下、隣接フレームの量子化ステップの変動に制限をかけることによって、フレーム間の品質の変動を低減した。最後に実験によって、提案方式の性能評価を行った。提案方式は、定められた遅延の下で、各フレームの品質を劣化させることなく、変動を抑えることができることを明らかにした

第5章は、結論であり、本論文の今後の課題について述べている。

5.2 今後の課題

今後の課題として EBCOT 処理の短縮が挙げられる。JPEG2000 符号化処理の中で、最も処理に時間がかかるのは、RD 最適化を含む EBCOT 処理である。本論文では、コーデック遅延として主に、入力画像の読み込み時間とバッファリング時間に注目して研究を行った。一方で、符号化処理自体に注目すれば、複雑な EBCOT 処理にはまだまだ処理時間を短縮する余地があると考えられる。コーデック遅延の一因として、検討を続ける必要がある。

謝辞

本研究の機会及び素晴らしい実験環境を与えて下さり、貴重な時間を割いて研究の方向性を御指導頂きました渡辺 裕 教授に心から感謝致します。

本研究は、株式会社 KDDI 研究所との委託研究「JPEG2000 および H.264/AVC に基づく高精細動画の高品質伝送方式に関する研究」(2004年9月1日から2005年3月31日まで)、共同研究「高精細動画に対する JPEG2000 ベース高性能符号化方式に関する研究」(2005年9月1日から2006年3月31日まで)による。共同研究の機会を与えていただき、貴重なご意見を頂くなど、お世話になった株式会社 KDDI 研究所に感謝致します。特に、研究を進めるにあたり、お忙しい中たびたび時間をとって頂き、研究の進め方から文章の書き方まで丁寧に御指導下さった内藤 整氏に深く感謝致します。

また、学部時代の指導教授であり、研究の方向性について、数々の有意義な御意見、御助言を賜りました早稲田大学国際情報通信研究科 富永 英義教授に深く感謝致します。

貴重な御意見、様々な御提案を頂いた AVS ゼミの皆様に御礼申し上げます。

同じ分野の研究を行っていて、貴重な意見を頂いた石川 孝明氏に心から感謝致します。

英語執筆の際に、時間を割いて入念な添削を行ってくれたばかりか、私を、インドという素晴らしい国へ導いてくれた、Abhay Gatophande 氏に心から感謝致します。

学部時代からの同級生でいろいろと教えてくれた河村 圭氏と常松 祐一氏に心から感謝致します。

学部時代からお世話になった富永研究室の皆様に、御礼申し上げます。特に、3年間公私にわたり、共に頑張ってきた同級生のみなに感謝致します。

最後に、私をここまで育てて下さった家族に深く感謝します。

平成 18 年 3 月 15 日

参考文献

- (1) 桜井, 小川, “地上デジタル放送の技術動向と将来展望,” 東芝レビュー, Vol.58 No.12, pp2-6, 2003
- (2) 杉本, 荒関, “地上デジタル放送の動向,” NEC 技法, Vol.57 No.4, pp3-7, 2004
- (3) 久下哲郎, “放送におけるコンテンツ伝送・配信技術,” 映像情報メディア学会誌, Vol.58 No.2, pp175-182, 2004
- (4) 八島由幸, “高品質 HDTV 映像通信・配信サービスを支える符号化伝送技術,” NTT 技術ジャーナル, pp8-11, 9, 2003 .
- (5) ISO/IEC 13818-2:2000 — ITU-T Rec. H.262, “Information technology - Generic coding of moving pictures and associated audio information: Video,” 2000
- (6) ISO/IEC 14496-10, “Advanced Video Coding,” ITU-T Rec. H.264, 2003.
- (7) ISO/IEC JTC1/SC29/WG1 N1890, “JPEG2000 part I Final Draft International Standard,” Sep. 2000.
- (8) D. Marpe et al. “Performance evaluation of Motion-JPEG2000 in comparison with H.264/AVC operated in intra coding mode,” in Proc. SPIE Conf. on Wavelet Applications in Industrial Processing, Oct. 2003.
- (9) Digital Cinema Initiatives, “Digital Cinema System Specification version 4.3,” Dec. 2004.
- (10) 貴家仁志, “JPEG2000 を中心とした画像圧縮技術の新しい流れ,” Interface, pp46-58, Jan. 2002.
- (11) 貴家仁志, 渡辺修, “JPEG2000 符号化アルゴリズムの要素技術,” Interface, pp59-71, Jan. 2002.
- (12) K. Berkner, E.L. Schwartz, “Removal of Tile Artifacts Using Projection Onto Scaling Functions for JPEG 2000,” Proceedings of IEEE ICIP'2002, Rochester New York, 2002.
- (13) M. HASHIMOTO, K. MATSUO, A. KOIKE, “JPEG2000 encoder for reducing tiling artifacts and accelerating the coding process,” International Conference on Image Processing, MP-L1.7, 2003
- (14) R. Rosenbaum, D. Taubman, “Merging images in JPEG2000-domain, IASTED-VIIP2003, Benalmadena/Spain, Sep. 2003.

- (15) J. C. Rountree, T. J. Flohr, M. W. Marcellin, "Report on core experiment codeff5: scan-based processing mode," ISO/IEC JTC1/SC29/WG1 N1594, Mar. 2000.
- (16) T. J. Flohr, M. W. Marcellin, J. C. Rountree, "Scan-based processing with JPEG-2000," Proc. of SPIE, Appl. of Digital Image Proc., pp 347-355, July, 2000.
- (17) ISO/IEC JTC1/SC29/WG1 N2250, "Motion JPEG2000 Final Draft International Standard 1.0," Sep. 2001.
- (18) 福原隆浩, "JPEG2000/Motion-JPEG2000の技術概要と応用(前編)," Interface, Nov. 2002.
- (19) 福原隆浩, "JPEG2000/Motion-JPEG2000の技術概要と応用(後編)," Interface, Dec. 2002.
- (20) 久下哲郎, "Motion-JPEG2000規格化の現状と課題," 映像情報メディア学会誌, Vol.54 No.12, pp1695-1702, 2000.
- (21) W. Chan, A. Becker, "Efficient Rate Control for Motion JPEG2000," Proceedings of DCC2004, 2004.
- (22) ISO/IEC JTC1/SC29 WG1 N2144, "Motion JPEG2000 Verification Model ver.4.1," June 2001.
- (23) S. Naito, A. Koike, S. Matsumoto, "Optimal JPEG2000 Encoder Mechanism for Low Delay and Efficient Distribution of HDTV Programs," Proceedings of ICASSP2004, May 2004.
- (24) T. Kuge, "Wavelet Picture Coding and Its Several Problems of the Application to the Interlace HDTV and the Ultra-high Definition Images," ICIP2002, Vol. III, pp.217-220, Sep. 2002.
- (25) 内藤 整, 小池 淳, 和田正裕, 松本修一, 羽鳥好律, "ひずみ最小型レート制御の高度利用に基づく MPEG-2 ビット配分の最適化," 信学論 D-II, vol.J86-D-II, no.11, pp.1565-1574, Nov. 2003.
- (26) "KaKadu software,"
<http://www.kakadusoftware.com/>
- (27) 佐野雄磨, 内藤 整, 渡辺 裕, "JPEG2000 Precinct 構造を用いた低遅延 HDTV 符号化方式," 情処学研報, 2004-AVM-46 Oct. 2004.
- (28) M.Hamdi, J.W.Roberts, and P. Rolin, "Rate Control for VBR Video Coders in Broad-Band Networks," IEEE Journal on Selected Areas in Communications, Vol.15, No6, Aug. 1997.

- (29) J.Bai, Q.Liao, X.Lin, X.Zhuang, "Rate-distortion model based rate control for real-time VBR video coding and low delay communications," *Signal Processing, Image Communication* 17, pp.187-199, 2002.
- (30) A.Jagmohan, K.Ratakonda, "MPEG4 One-Pass VBR Rate Control for Digital Storage," *IEEE Transactions on Circuits and Systems for Video Thecnology*, Vol.13, No5, May 2003.
- (31) J. C. Dagher, A. Bilgin, M. W. Marcellin, "Resource-Constrained Rate Control for Motion JPEG2000," *IEEE Transactions on Image Processing*, Vol. 12, No. 12, pp. 1522- 1529, Dec. 2003.
- (32) ISO/IEC 13818-2, "Information Technology Generic coding of moving pictures and associated audio information-Part2:Video," 1994.

付録 A

バッファリング遅延量の導出

以下に、バッファモデル導入時における送信側および受信側で発生するバッファリング遅延量の総和を推定する。受信側において、検証バッファのサイズ、検証バッファよりタイル t ($t = 0, 1, \dots$) が引き抜かれた直後の占有量をそれぞれ、 L_V 、 $v(t)$ とするとき式 (A.1)、式 (A.2) が成立する。 v_0 、 B_W 、 $tile_cnt$ 、 $S(i)$ の定義は本文中 3.5. に従うこととする。

$$v(t) = v_0 + t \times B_W / tile_cnt - \sum_{i=0}^t S(i) \quad (A.1)$$

$$0 \leq v(t) \leq L_V \quad (A.2)$$

次に送信側においても同様に検証バッファを規定する。検証バッファにタイル t ($t = 0, 1, \dots$) が書き込まれた直後の占有量 $vs(t)$ は式 (A.3) により算出される。 vs_0 は先頭タイルの書込みが行われる直前の占有量である。検証バッファの破綻防止のため、式 (A.4) に示す条件が常に満足される必要があるが、これは vs_0 として式 (A.5) に従う値を選択することで実現できる。よって、送信側でのストリーム送出待ち時間、および受信側での復号開始待ち時間の合計を T_W とおくと、同値は式 (A.6) に示すとおり、検証バッファサイズを符号化レートで除した値により表せる。

$$vs(t) = vs_0 + \sum_{i=0}^t S(i) - t \times B_W / tile_cnt \quad (A.3)$$

$$0 \leq vs(t) \leq L_V \quad (A.4)$$

$$vs_0 = L_V - v_0 \quad (A.5)$$

$$\begin{aligned} T_W &= \frac{v_0 + vs_0}{bit_rate} \\ &= \frac{L_V}{bit_rate} \end{aligned} \quad (A.6)$$

図一覧

| | | |
|------|--|----|
| 1-1 | HDTV の無線伝送 | 2 |
| 1-2 | JPEG2000 による HDTV 符号化 | 3 |
| 2-1 | JPEG2000 の符号化アルゴリズム | 7 |
| 2-2 | 2 分割フィルタによる信号の分析・合成システム | 9 |
| 2-3 | 分解レベルが 2 の場合の 2 次元空間に対するサブバンド分解 | 9 |
| 2-4 | 各サブバンドの配置 | 10 |
| 2-5 | Lena 画像の 3 段階 Wavelet 変換 | 10 |
| 2-6 | ビットプレーン分解と符号化パスへの分解 | 13 |
| 2-7 | コードブロックデータのレイヤへの分割の例 | 14 |
| 2-8 | タイルの分割法 (左) とタイル境界の不連続性 (右) | 15 |
| 2-9 | タイル構造のサブバンド分割 | 16 |
| 2-10 | プレシント構造のサブバンド分割 | 16 |
| 2-11 | Motion JPEG2000 のピクチャ構造 | 17 |
| 2-12 | MPEG-2 のピクチャ構造 | 18 |
| 2-13 | 3 コンポーネントの場合のピクチャ構造 | 19 |
| 2-14 | 符号化コードストリーム構造 | 19 |
| 2-15 | フレームベースの YUV コンポーネントのピクチャ構造 | 20 |
| 2-16 | フレームベース符号化時の Wavelet 変換 | 20 |
| 2-17 | フィールドベース符号化時の Wavelet 変換 (preceding_field_first=1) | 21 |
| 2-18 | フィールドベース符号化時の Wavelet 変換 (preceding_field_first=0) | 21 |
| 2-19 | JP2 のファイル構造 | 22 |
| 2-20 | Movie を再生する際の QuickTime のしくみ | 24 |
| 2-21 | 複数の track を持つ movie の例 | 25 |
| 2-22 | QuickTime の基本的な Box 構造 | 26 |
| 2-23 | moov による時間管理 | 26 |
| 2-24 | jp2 と mj2 の構造 | 27 |
| 2-25 | MJ2 ファイルフォーマットでの画像と音声の同期 (コントロールフロー) | 28 |
| 2-26 | MJ2 ファイルフォーマットでの画像と音声の同期 (データフロー) | 28 |
| 3-1 | 入力画像非分割時の読み込みと符号化 | 31 |
| 3-2 | 入力画像分割時の読み込みと符号化 | 32 |

| | | |
|------|--|----|
| 3-3 | Rate,Distortion, の関係 | 33 |
| 3-4 | 想定する符号化器 | 34 |
| 3-5 | ビデオフィールドに対するタイル分割処理 | 35 |
| 3-6 | タイル構造 A) における検証バッファ占有量の時間遷移 | 43 |
| 3-7 | タイル構造 B) における検証バッファ占有量の時間遷移 | 44 |
| 3-8 | プレシントの生成に必要な画像エリア | 46 |
| 3-9 | プレシント構造 C) のレイアウト | 48 |
| | | |
| 4-1 | foreman1 原画像 | 53 |
| 4-2 | foreman2 原画像 | 53 |
| 4-3 | foreman1 0.5bpp で符号化 | 53 |
| 4-4 | foreman2 0.5bpp で符号化 | 53 |
| 4-5 | Q 一定時と 一定時の品質変動 (Streetcar) | 55 |
| 4-6 | Q 一定時と 一定時の品質変動 (Whaleshow) | 56 |
| 4-7 | バッファ占有量の遷移の制限 | 60 |
| 4-8 | 等ビット配分方式と提案手法の PSNR の比較 -Streetcar- (バッファサイズ 10Mbit) | 62 |
| 4-9 | 等ビット配分方式と提案手法の PSNR の比較 -EuropeanMarket- (バッファサイズ 10Mbit) | 63 |
| 4-10 | 提案手法によるバッファ占有量の遷移 -Streetcar- (バッファサイズ 10Mbit) | 64 |
| 4-11 | 提案手法によるバッファ占有量の遷移 -EuropeanMarket- (バッファサイズ 10Mbit) | 65 |

表一覧

| | | |
|------|---------------------------------|----|
| 1-1 | 主な映像フォーマット | 2 |
| 2-1 | JPEG2000 の各パート | 7 |
| 2-2 | 5 × 3 フィルタの係数 (順変換) | 11 |
| 2-3 | 5 × 3 フィルタの係数 (逆変換) | 11 |
| 2-4 | 9 × 7 フィルタのフィルタ係数 (順変換) | 11 |
| 2-5 | 9 × 7 フィルタのフィルタ係数 (逆変換) | 11 |
| 2-6 | QuickTime で movie を再生するときの重要な機能 | 25 |
| 2-7 | Box のタイプと階層構造 | 30 |
| 3-1 | 符号化条件 | 36 |
| 3-2 | 分割符号化の品質に与える影響 | 36 |
| 3-3 | 分割符号化の遅延に与える影響 | 36 |
| 3-4 | 符号化条件 2 | 40 |
| 3-5 | 対象とするタイル構造 | 41 |
| 3-6 | 提案方式による PSNR の結果 | 42 |
| 3-7 | 従来方式による PSNR の結果 | 42 |
| 3-8 | 所要バッファサイズの推定値 | 45 |
| 3-9 | コーデック遅延量の推定値 | 45 |
| 3-10 | 対象とするプレシнкт構造 | 47 |
| 3-11 | 提案方式による PSNR の結果 | 48 |
| 3-12 | 従来方式による PSNR の結果 | 49 |
| 3-13 | 所要バッファサイズの推定値 | 49 |
| 3-14 | コーデック遅延量の推定値 | 49 |
| 4-1 | 符号化条件 | 61 |
| 4-2 | PSNR の平均と分散 (Streetcar) | 62 |
| 4-3 | PSNR の平均と分散 (EuropeanMarket) | 63 |
| 4-4 | コーデック遅延量の推定値 | 65 |

研究業績

1. 佐野 雄磨, 内藤 整, 渡辺 裕
“JPEG2000 Precinct 構造による低遅延 HDTV 符号化方式,”
FIT2004, J-076, 2004.9
2. 佐野 雄磨, 内藤 整, 渡辺 裕
“JPEG2000 Precinct 構造を用いた低遅延 HDTV 符号化方式,”
オーディオビジュアル複合処理研究会 46-02, 2004.10
3. 佐野 雄磨, 内藤 整, 渡辺 裕
“Motion JPEG2000 低遅延符号化のための時間方向ビット配分,”
オーディオビジュアル複合処理研究会 47-24, 2004.12
4. 河村 圭, 佐野 雄磨, 伊谷裕介, 加藤幸一, 後藤崇行, 山本勇樹
“映像現場訪問記 情報通信研究機構本庄情報通信研究開発支援センター,”
映像情報メディア学会誌, Vol59, No.9, pp1297-1299, 2005.9
5. 佐野 雄磨, 内藤 整, 渡辺 裕
“JPEG2000 による HDTV 低遅延号化のための符号量制御方式,”
FIT2005, J-062, 2005.9
6. 佐野 雄磨, 内藤 整, 渡辺 裕
“品質変動低減のための JPEG2000 符号量制御方式,”
オーディオビジュアル複合処理研究会 51-17, 2005.12
7. 内藤 整, 佐野 雄磨, 渡辺 裕, 小池 淳, 松本 修一
“低遅延高効率 HDTV 伝送のための JPEG2000 符号化制御の最適化,”
電子情報通信学会論文誌, VolJ88-D2, No.12, pp.2311-2320, 2005.12