

平成17年度 修士論文

Computational Complexity Reduction
for H.264/MPEG-4 AVC Video Coding

H.264/MPEG-4 AVC ビデオ符号化における
演算量削減手法に関する研究

指導教授 渡辺 裕 教授

2006年3月

早稲田大学大学院 国際情報通信研究科
国際情報通信研究学専攻

4204A149-6

常 松 祐 一

Yuichi TSUNEMATSU

目次

第1章	序論	1
1.1	研究の背景	1
1.2	研究の目的	2
1.3	本論文の構成	3
第2章	動画像符号化技術	5
2.1	まえがき	5
2.2	動画像符号化技術	5
2.2.1	動画像符号化技術の必要性	5
2.2.2	基本処理構成	6
2.2.3	標準化の歴史	7
2.3	H.264/MPEG-4 AVC	10
2.3.1	標準化の経緯	10
2.3.2	アプリケーション	10
2.3.3	プロファイルとレベル	11
2.3.4	符号化要素技術	12
2.3.5	演算量と符号化効率への寄与	16
2.4	むすび	17
第3章	Intra 予測処理の演算量削減手法	26
3.1	まえがき	26
3.2	Intra 予測処理演算量削減の従来手法	26
3.3	Intra 予測処理演算量削減の提案手法	26
3.4	提案手法の評価	28
3.4.1	実験条件	28
3.4.2	実験結果	29
3.4.3	考察	29
3.5	むすび	30
第4章	Inter 予測処理の演算量削減手法	35
4.1	まえがき	35
4.2	Inter 予測処理演算量削減の従来手法	35

4.3	Inter 予測処理演算量削減の提案手法	36
4.4	提案手法の評価	38
4.4.1	実験条件	38
4.4.2	実験結果	38
4.4.3	考察	39
4.5	むすび	40
第5章	組み合わせ時の性能評価	45
5.1	まえがき	45
5.2	全探索時の性能評価	45
5.2.1	実験条件	45
5.2.2	実験結果	45
5.2.3	考察	57
5.3	FastME 使用時の性能評価	62
5.3.1	FastME の探索手法	62
5.3.2	実験条件	62
5.3.3	実験結果	62
5.3.4	考察	72
5.4	むすび	72
第6章	結論	77
6.1	総括	77
6.2	今後の課題	78
6.2.1	最適な符号化処理単位の検討	78
6.2.2	動き補償の改善	78
	謝辞	80
	参考文献	81
	図一覧	84
	表一覧	86
	研究業績	87

第1章

序論

1.1 研究の背景

近年画像処理の分野におけるデジタル技術の発展が著しい。半導体技術，通信技術，高密度実装技術の進展により画像関連デバイスの進化が急速に進んでいる。デジタルカメラ，デジタルビデオカメラといったデジタル画像の入力が可能な撮像系機器の普及，編集に利用するPCの大幅な性能向上と低価格化，安価に利用出来る広帯域ネットワークの普及に伴い，入力から編集，出力に至る全ての過程をデジタル処理出来る環境が整いつつある。デジタル画像データは“撮影”，“編集・加工”，“記録・出力”，“伝送”，“保存・蓄積”，“見る”という一連のやりとりを循環して利用出来るようになってきた。

編集・加工作業では早くからデジタル技術が画像処理に用いられてきたが，今後はコンテンツの配信も全てデジタル環境へ移行していくことが予想される。デジタル画像処理技術を用いるアプリケーションの範囲もテレビ電話やテレビ会議，ストリーミング配信など多様化してきている。ネットワークを介して遠く離れた地域に瞬時にデータを送ることが出来るなど，デジタル技術があって初めて実現できたアプリケーションは多く，その恩恵は計り知れない。やりとりされるコンテンツは以前のように静止画だけでなく，動画像も近年流通量が増加している。デジタル技術は機器の小型化，高機能化，データの機器間共有を可能とし，デジタル機器は単体で機能する時代から，相互に強調して動作するネットワーク型のシステムへと変わりつつある。

今後流通する動画像コンテンツに起こる変化として予想出来ることに，高解像度化と携帯端末への対応の2点が挙げられる。ネットワークの広帯域化，蓄積メディアの大容量化，地上波テレビのアナログ放送から地上波デジタル放送への移行，液晶，プラズマ，SED(Surface-conduction Electron-emitter Display:表面電界ディスプレイ)といった表示装置の技術発展などを背景に，近年動画像コンテンツの高解像度化が進んでいる。新たに発売されるテレビやPC用ディスプレイはフルHD規格に対応しているものも少なくなく，値段も日々下がっており一般家庭にフルHD規格に対応した表示装置が広まりつつある。テレビの地上波アナログ放送が終了する2011年には一般家庭にフルHD規格に対応したテレビが十分普及していることも考えられる。表示装置のフルHD規格対応に伴い，撮像系，制作・編集環境も高解像度化が進んでいる。放送以外の分野でもデジタル化が進んでおり，映画の分野では

4000×2000[pixel] サイズの画像までカバーする DCI 規格が策定された [1] . 30G バイト以上のデータが書き込み可能な次世代光ディスクも今後展開される予定であり, 高解像度で撮影されたコンテンツがそのまま編集され, 流通され, 縮小されることなく一般家庭において表示されることも十分に予想出来る. NHK では 8000×4000[pixel] サイズのスーパーハイビジョンを 2020 年までに一般家庭に普及させることを目的として研究を開始しており, 今後とも動画像コンテンツの高画質化は進むと考えられる [2] .

次に動画像コンテンツの携帯端末への対応について述べる. 次々と新製品が発売される携帯電話には様々な機能が追加され, 地上波テレビ放送が受信可能なものや外部メモリに保存した動画像を再生する機能が付いているものもある. また携帯ゲーム機においても動画像の再生機能が追加されたり, 音楽や動画像コンテンツの再生に特化した携帯機器も登場しつつある. インターネットを用いての動画像コンテンツ配信もすでに開始されており, 対応する機器も増えている. 今はネットワークの帯域幅や, メモリの記録容量の問題から短編のコンテンツが主流であるが, 将来は映画など長編コンテンツのやりとりが一般的になることも考えられる.

以上より今後の動画像データは QCIF サイズの小さな携帯向けコンテンツから HD 規格以上のサイズの高解像度コンテンツまで幅広く対応する必要があると言える. 動画像コンテンツの需要は今後も増えると予想され, デジタルカメラが静止画を手軽に楽しむことが出来るようにしたのと同様, デジタルビデオカメラを初めとした機器が動画像を楽しむことをより簡単にする可能性がある. HDD 搭載のレコーダーも普及しつつあり, 誰もが好きなコンテンツをいつでも, どこでも楽しめる環境が整いつつある.

こうした環境を実現するために様々な画像処理技術が検討され, 実際に利用されてきた. 主だったものに限定してもノイズ除去, 画像認識, 高詳細化, レート制御など様々な分野にわたる. 中でも符号化技術は長くに渡り研究が進められており, 重要な役割を果たしている. デジタル画像はデータ量が大きく圧縮無しにやりとりすることは困難である. 符号化技術が無ければ現在の画像流通環境は実現できなかったと言える. 符号化技術にはあらゆる機器での対応やさらなる符号化効率の向上, フィルムを置き換える品質の高画質化といった要求がある.

1.2 研究の目的

本研究は動画像の符号化処理において効率的な符号化処理を行い, 演算量を削減する手法について検討を行う. 2003 年に MPEG(Moving Picture Experts Group) と ITU-T(International Telecommunication Union - Telecommunication Standardization Sector, 旧 CCITT) の合同標準化部会である JVT(Joint Video Team) で標準化された H.264/MPEG-4 AVC(以後 H.264 と表記) では符号化効率を高めるためにこれまでの標準化では採用が見送られてきた多くの演算量を必要とする符号化機能が多数採用されている [3]. 符号化効率の改善を積み重ねて, 高い符号化効率を実現していることにより演算量が非常に多くなった.

標準化仕様通りにエンコーダーを実装すると, H.264 の演算量は一般的に MPEG-2 の 10

倍以上になると言われている [4]。実際に符号化処理を行うハードウェアの性能はこれまでムーアの法則に従い15ヶ月で2倍となるペースで増加してきた。これまで利用されてきた符号化方式は符号化処理にかかる演算量の増加によりハードウェアの性能向上が追いついている状況であった。しかし H.264 での急激な演算量の増加によりムーアの法則によるハードウェアの性能向上を期待してはいつまでも H.264 が利用できない状況となっている。そこで不要な符号化処理を簡略化し、符号化した画像品質を落とすことなく効率的に処理する手法が必要とされている。

現在の動画像符号化方式は DCT(Discrete Cosine Transform) などの直交変換と動き補償を用いたハイブリッド方式が主流となっており、MPEG で標準化された MPEG-1, MPEG-2, MPEG-4, ITU-T で標準化された H.261, H.262, H.263, そして H.264 もハイブリッド方式の動画像符号化方式である [5]。ハイブリッド型の符号化方式では動き補償に関連する Inter 予測処理が特に多くの演算量を占める。参照フレームを探索し、動きベクトルを求める演算量の削減は以前から検討が行われてきており、効率的な探索手法がいくつも提案されている。H.264 からは新たに複数フレームの参照や 1/4 画素精度の動き補償、動き補償時の可変ブロックサイズが導入され、動き補償に関連する演算量が飛躍的に増加した。また画面内で画素値の予測を行う Intra 予測も多くの演算を必要とする。Intra 予測は多くの予測方向が存在し、全てのパターンを計算した上で利用する符号化モードを決定すると演算量が問題となる。他にも多くの演算量を必要とする符号化機能はいくつかあるが特にこの2つが占める割合が大きい。

また符号化処理に必要な演算量の増加は今後も続くことが予想される。ITU-T では次の動画像符号化方式の標準として H.265 の検討を開始した。まだ具体的な方式は決まっていないが H.264 と同じくハイブリッド方式で実現されるのであれば、同じく符号化演算量の問題に直面することが考えられる。また現在 JVT で標準化作業が行われているスケーラブル符号化方式の SVC(Scalable Video Coding) は H.264 ベースで標準化が進められている。H.264 における符号化演算量の問題点はそのま SVC においても問題となることが考えられ、H.265 以降も問題となることが考えられる [6, 7]。

そこで本論文では Inter 予測と Intra 予測に着目し、符号化演算量の削減を行う手法について検討する。提案手法では符号化処理サイズの違いに着目し、符号化効率の低下を抑えつつ符号化処理を簡略化することを目的とする。

1.3 本論文の構成

以下に次章以降の本論文の構成を示す。

第1章 本章であり、研究の背景およびその目的について述べている。

第2章 動画像符号化技術について述べる。始めに動画像符号化技術が登場した背景と圧縮技術の必要性を述べ、これまでの標準化の歴史と各標準符号化方式で利用されている符号化機能を紹介する。また H.264 以外に近年標準化が行われている VC-1 や SVC の概要に

についても述べる．次に本論文で扱う H.264 を取り上げ，標準化の経緯やアプリケーション，H.264 で新たに追加された符号化機能を紹介する．最後に現在の符号化技術で問題となっている演算量の問題を取り上げ，H.264 の Intra 予測と Inter 予測が符号化効率にどの程度寄与しているか，どの程度の演算量を占めているか基礎検討を行う．

第3章 Intra 予測の演算量削減手法について述べる．H.264 では符号化効率を高めるために新たに画面内で画素値の予測を行う Intra 予測が導入されたが，符号化演算量増加の一因となっている．始めに Intra 予測の演算量削減を行う従来手法を紹介し，次に本論文で提案する複数の処理サイズの違いを用いた Intra 予測の符号化演算量削減手法を述べる．提案手法では 4×4 画素単位の Intra 予測モード結果を利用して 8×8 と 16×16 画素単位の Intra 予測処理を制限する．最後に符号化演算量の削減度合いと符号量，PSNR の観点から提案手法の評価を行う．

第4章 Inter 予測の演算量削減手法について述べる．Inter 予測は H.264 以前の符号化方式でも利用されてきたが，符号化効率を高めるために複数参照フレームや可変ブロックサイズ動き補償， $1/4$ 画素精度動き補償などの機能が追加され符号化演算量増加の一因となっている．始めに Inter 予測の演算量削減の従来手法を紹介し，次に本論文で提案する Inter 予測の符号化演算量削減手法を述べる．提案手法では符号化対象のマクロブロックと探索中に参照するマクロブロックで画素値の差分を取り，この情報を利用することで探索する動き補償ブロックサイズの種類を効率よく削減する．最後に符号化演算量の削減度合いと符号量，PSNR の観点から提案手法の評価を行う．

第5章 第3章で提案する Intra 予測と第4章で提案する Inter 予測の演算量削減手法を組み合わせて用いた際の性能評価を行う．実験を行い，2つの提案手法を同時に利用することが出来ることを示す．また H.264 の符号化演算量削減手法として有名な FastME と提案手法を組み合わせたときの性能評価を行う [8]．符号化演算量の削減度合いと符号量，PSNR の観点から提案手法の評価を行う．また符号化後の画像と符号化モードの利用割合の変化を示して提案手法により主観的な品質劣化が無い事を示す．

第6章 本研究の総括と今後の課題について述べる．

第2章

動画像符号化技術

2.1 まえがき

本章では動画像符号化技術について述べる．始めに動画像符号化技術が登場した背景と圧縮技術の必要性を述べ，これまでの標準化の歴史と各標準符号化方式で利用されている符号化機能を紹介する．また H.264 以外に近年標準化が行われている VC-1 や SVC の概要についても述べる．次に本論文で扱う H.264 を取り上げ，標準化の経緯やアプリケーション，H.264 で新たに追加された符号化機能を紹介する．最後に現在の符号化技術で問題となっている演算量の問題を取り上げ，H.264 の Intra 予測と Inter 予測が符号化効率にどの程度寄与しているか，どの程度の演算量を占めているか基礎検討を行う．

2.2 動画像符号化技術

2.2.1 動画像符号化技術の必要性

文字，音声，静止画，動画像などのデジタルデータを統合して扱うマルチメディアがデジタル技術の進展により現実のものとなった．歴史的には 1980 年代半ばから取り組みが続けられているが一般的に利用されるようになったのはここ数年の出来事であると言える．実現を可能にした要因の一つにデジタル機器の性能向上とネットワークの広帯域化が挙げられる．これらにより，文字や静止画だけにとどまらず，音声，動画像も容易にやりとり出来るようになった．しかし手軽なデータのやりとりが実現すると，次は高品質なコンテンツが求められるようになり，扱うデータ量は増大していく傾向にある．

文字，音声，静止画と比較して動画像は非常に多くのデータ量を必要とする違いがある．ネットワーク帯域が広帯域化した現在でも全ての情報をそのままやりとりすることは困難である．SDTV(Standard Definition Television:標準テレビ)の動画像データを，デジタル静止画像を並べて表現したものとして計算するとビットレートは 124[Mbps] になり，HDTV(High Definition Television:高詳細テレビ)だと 746[Mbps] となる．これは現在普及している回線で最も高速な光ファイバーを用いてもやりとりすることが出来ないビットレートとなる．

そこでデジタル動画像の冗長な情報を削除し，データサイズを小さくしてやりとりしやすくする圧縮技術が必要となる．圧縮は完全に元の情報へ戻すことが可能な可逆圧縮と，圧

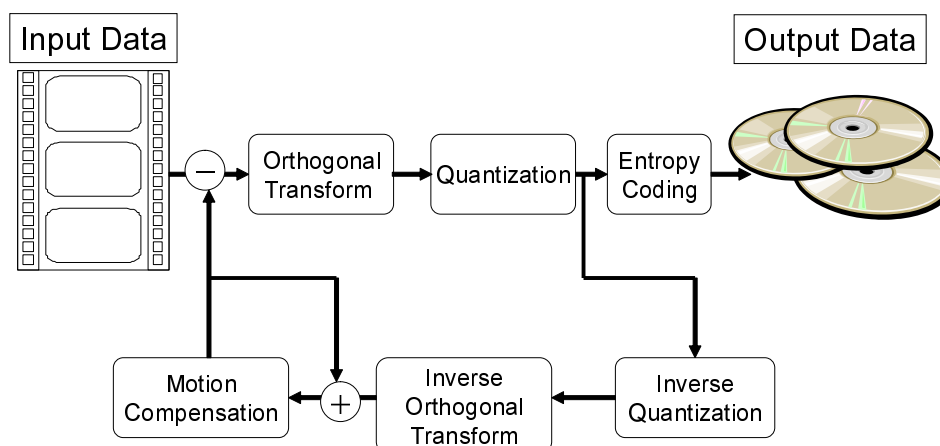


図 2-1: 動画像圧縮技術の基本処理構成

縮後の状態から元の情報を完全復元できない非可逆圧縮に大別出来る。可逆圧縮の動画像符号化方式の例として Motion-JPEG2000 [9] が、非可逆圧縮の動画像符号化方式の例として MPEG-1/MPEG-2/MPEG-4, H.264 が挙げられる [10, 11]。最新の可逆圧縮符号化方式を用いると元の半分程度のサイズへ動画像データを圧縮することが出来るが、非可逆圧縮符号化方式を用いると完全に情報を復元できない代わりに高い圧縮率を実現することが出来る。先に出た HDTV の動画像データを例にとると MPEG-2 では約 1/40 の 20[Mbps] に、H.264 では 1/80 の 10[Mbps] で符号化出来ると言われている。

2.2.2 基本処理構成

動画像の圧縮では変換と予測を行うことで元の画像信号に含まれる冗長性を削減する。特に以下に挙げられる性質を利用することが多い。

- ある画素値の周辺画素値との相関性。
- ある画素値の前後フレームとの相関性。
- 画素をグループとしてみたときのなだらかな値の変化。

現在主流となっている圧縮技術では空間方向と時間方向の両方の冗長性を削減しているが、空間方向では変換を、時間方向では予測を用いるのが一般的である。動画像の圧縮符号化は時間方向に働く動き補償と、その残差信号の符号化で構成される。一般的な動画像圧縮技術の基本処理構成を図 2-1 に示す。

入力画像は前後参照フレームとの差分を取り、その後直交変換、量子化、エントロピー符号化が施される。前後参照フレームとの差分はいくつかの小さいブロック単位で行われ、処理単位はマクロブロックと呼ばれる。参照フレームと同じ位置から差分が求められるのではなく、位置をずらして最も差分が小さくなる箇所で求められ、この処理を動き補償と呼ぶ。動き補償処理の原理を図 2-2 に示す。

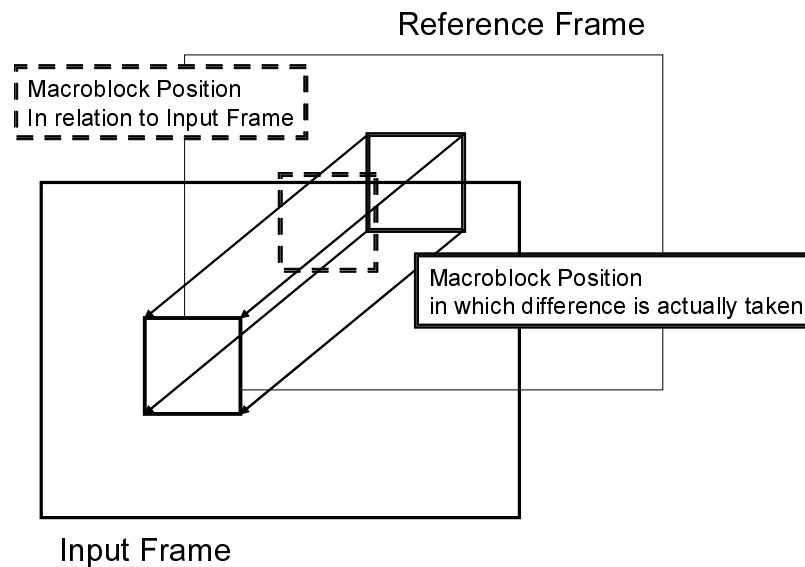


図 2-2: 動き補償の原理

動き補償処理で、差分を求めるのに最も適切な位置への移動量は動きベクトルと呼ばれ、残差信号と共に符号化される。動き補償の単位であるブロックサイズは 16×16 画素単位が一般的に用いられているが、 8×8 画素単位や 4×4 画素単位といったブロックサイズも最近の符号化方式では利用されるようになった。小さいブロックサイズを用いた方が差分情報が小さくなる可能性が高いが、ヘッダに記録する動きベクトル情報が増えるためどのブロックサイズを動き補償に利用した方が符号化効率が高くなるかを符号化時に判断する必要がある。動きベクトルを求める推定処理にはブロックマッチングがよく用いられる。ブロックマッチングは画素値差分の二乗誤差や絶対値誤差が最小となる位置を探索範囲内で位置をずらしながら探索する。

動き補償後の差分情報は直交変換された後、エントロピー符号化される。直交変換は周囲の画素値との相関性の性質を利用したもので、低周波成分にエネルギーを集中させることで符号化効率を高める役割がある。直交変換としてはDCTが多く用いられている。動き補償と直交変換を核とする符号化方式はハイブリッド符号化方式と呼ばれ、現在の動画像符号化方式で多く採用されている。

2.2.3 標準化の歴史

動画像符号化技術が利用されている分野は大きく以下の3つに分類することが出来る。

- 放送分野。
- 通信分野。
- 蓄積分野。

表 2-1: 国際標準の動画像符号化方式

ISO/IEC	MPEG-1	MPEG-2	MPEG-4 Part2	MPEG-4 Part10
Standardization Year	1993	1994	1998	2003
Picture Type	I/P/B	I/P/B	I/P/B	I/P/B
Interlace	-	Frame/Field	Frame	Frame/Field
Macroblock Size	16×16	16×16	16×16 8×8	16×16 ~ 4×4
MC Accuracy	1/0.5	0.5	0.5	0.25
Orthogonal Transform	8×8 DCT	8×8 DCT	8×8 DCT	4×4/8×8 Integer Transform
Application	VideoCD	Multipurpose	Mobile Contents	Multipurpose

ITU-T	H.261	H.262	H.263	H.264
Standardization Year	1989	Same as MPEG-2 Standard	1998	Same as MPEG-4 Part10 Standard
Picture Type	P		I/P/B	
Interlace	-		Frame	
Macroblock Size	16×16		16×16	
MC Accuracy	1		0.5	
Orthogonal Transform	8×8 DCT		8×8 DCT	
Application	Video Conference on ISDN		Mobile Contents	

動画像符号化の標準化を行う国際標準化機関は ISO(International Organization for Standardization)/IEC(International Electrotechnical Commission) やその下部組織である MPEG , ITU-T があるが、当初は分野ごとに標準化を行う機関が異なっていた。蓄積分野では MPEG が、通信分野では ITU-T が主導権をとって標準化を行ってきたが、近年では汎用的にどの分野でも利用出来る符号化方式が求められていることから共同で標準化を行うこともある。

ハイブリッド型の符号化方式は 1989 年に CCITT(当時) で標準化された H.261 から始まり、2003 年に標準化された H.264 までいくつか存在する。動画像符号化の標準化では符号化効率の向上と機能性の向上に重点が置かれ発展してきた。表 2-1 に国際標準の動画像符号化方式の概要をまとめたものを示す。ITU-T が標準化した規格には H.26x , MPEG が標準化した規格には MPEG-x という番号が割り振られている。H.262 と MPEG-2 , H.264 と MPEG-4 Part10 は ITU-T と MPEG が共同で標準化を行い、それぞれの機関で標準化番号が割り振られているが中身は同じものである。

1989 年に標準化された H.261 ではすでに動き補償と DCT を組み合わせたハイブリッド型の符号化アルゴリズムが採用されており、これは以降の標準にも引き継がれている。

1991 年に標準化された MPEG-1 は CD-ROM に動画を入れることを目的として標準化されたことから蓄積分野に特有の操作であるランダムアクセスや早送り、逆再生機能が追加された。符号化効率改善の点では未来に位置する参照フレームからの予測を行う B ピクチャや 1/2 画素精度動き補償が追加されている。

ITU-T と ISO/IEC JTC1 合同で標準化された H.262/MPEG-2 は汎用的な符号化方式として標準化された。現在でも DVD を始めとした蓄積分野や、日本における地上波デジタルテレビ放送など多くの用途に利用されている。放送分野で用いるためにインタレース対応がされ、フィールド/フレーム適応予測やフィールド/フレーム適応 DCT が機能追加された。多くの機能が追加されたことから利用出来る符号化機能をまとめたプロファイルと画像サイズを表すレベルによって標準のサブセットが定義されている。これは MPEG-2 対応端末の相互接続性を現実的に実現するためである。当初はテレビ放送向けに MPEG-2 が、HDTV 向けに MPEG-3 が標準化される予定であったが基本技術が同じであったことから MPEG-3 は欠番となり、MPEG-2 に一本化された。

H.263 はアナログ電話網におけるテレビ電話実現を目的として標準化され、H.261 を基に符号化効率向上の改善が行われた。具体的にはループフィルタが削除、シンタックスの再定義、3次元可変長符号の追加が行われている。

MPEG-4 Part2 では新たにオブジェクト単位の操作を目的として任意形状オブジェクト符号化機能が追加された。画面中で動作する物体をオブジェクトとし、オブジェクト単位で圧縮符号化が行われ、受信側で再構成して表示する仕組みになっている。

2003年に標準化された H.264/MPEG-4 Part10 では MPEG-2 や MPEG-4 からの大幅な圧縮率向上を目指して、これまでの標準化では採用が見送られてきた多くの符号化効率改善機能が追加されている。主立ったものに Intra 予測、可変ブロックサイズ動き補償、1/4画素精度動き補償、複数参照フレーム、算術符号化などがあり、ハイブリッド符号化における予測/変換処理をより細かく行うように機能改善が行われたと言える。

表 2-1 に挙げたもの以外にも、以下に挙げる動画像符号化方式もある。

VC-1 VC-1 は Microsoft 社により作成され SMPTE で標準化された動画像符号化方式である [12,13]。Microsoft 社は動画像符号化方式の開発を独自に進め、2003年に Windows Media Video9 を発表した。同社は Windows Media Video 9 を国際標準にするために SMPTE へ仕様を提案し、これが元になって VC-1 として標準化された。

VC-1 における符号化/復号化の処理内容は H.264 と大枠で類似している。しかし VC-1 では符号化効率と計算負荷の低減の両方を実現することを目的として H.264 にあるいくつかの機能が修正され、また H.264 には無い機能がいくつか追加されている。

VC-1 で動作が修正された機能に以下のものがある。

- 動き補償。
- 量子化・逆量子化。
- ループフィルタ。
- B ピクチャ符号化。

また H.264 に無く VC-1 で追加された機能に以下のものがある。

- 適応型ブロックサイズ直交変換。

- 16ビット処理を前提とした直交変換セット。
- 2層符号化。

SVC H.264の拡張としてJVTで標準化が行われているのがスケーラブル符号化を実現するSVCである。SVCでは空間、品質、時間の3つのスケーラブル機能を同時に利用することが可能であり、ワンソースマルチユースなど新たな動画像符号化の活用が検討されている。H.264が符号化効率を追求した符号化方式であるのに対し、SVCは機能性を追求した符号化方式であると言える。階層型Bピクチャなど新たに追加された符号化機能では符号化効率の改善も報告されている。

2.3 H.264/MPEG-4 AVC

2.3.1 標準化の経緯

H.264はITU-Tで1998年から検討が行われてきたH.26Lをベースに標準化が行われた。LはLong Termを意味し、長い期間をかけてH.263の半分のビットレートで同等の品質を実現する符号化方式の標準化を目的としたプロジェクトであった。2001年の夏にMPEGでも次世代動画像符号化の標準化作業を開始し、MPEG-4を含む当時検討対象となっていた6つの方式の中でH.26Lがもっとも高性能であったことから合同標準化部会であるJVTが発足し、標準化作業がITU-TとMPEGの合同で行われるようになった。2003年3月に最終企画案としてまとめ、ITU-TではH.264として、MPEGではMPEG-4 Part10 Advanced Video Codingとして標準化されたのがH.264である。

H.264では過去の符号化方式との互換性は考慮されず、演算複雑度の制限も度外視して検討が行われ、純粋に高い符号化効率が追求された。その結果として多くの符号化機能が搭載され、符号化効率も大きく向上した。H.264は特定のアプリケーションでのみ利用される符号化方式でなく、幅広い分野で利用できるよう検討が行われた。その結果、携帯端末やテレビ電話で利用される小さい画像サイズであるQCIFからHDTVといった大きい画像サイズでも従来の符号化方式を上回る符号化効率となっている。

2005年12月には高精細画像で主観画質を向上させる機能が追加されたHighプロファイルが標準化され、MPEG-4 AVCバージョン3として統合された。Highプロファイルでは色空間の拡張も行われ、業務用でも利用できる品質が実現できるようになった。

2.3.2 アプリケーション

2003年に標準化されたH.264はすでに様々な分野で利用されている。以下分野ごとにアプリケーションを紹介する。

通信 H.264を用いたテレビ電話/テレビ会議システムの普及が進んでいる。この分野では従来ITU-Tで標準化されたH.261やH.263が用いられてきたが、H.264を利用することで

より狭い帯域幅で高品質な動画像をやりとりすることが可能となる。主要メーカーの製品はすでに H.264 への対応が完了している。

テレビ電話/テレビ会議システム専用端末だけでなく、PC 上で動作するソフトウェアも H.264 への対応が進んでいる。中でも Apple 社が提供するビデオカンファレンスソフト iChat AV では Mac OS X Tiger に同梱されるバージョンから H.264 対応となった [14]。Apple 社の製品では他にも動画再生ソフトの QuickTime や携帯音楽再生プレーヤーである iPod も H.264 の再生に対応している [15,16]。

放送 2006 年 4 月 1 日に日本で開始される 1 セグメント放送で H.264 が採用された。これは地上波デジタルテレビ放送に割り当てられている周波数帯域の 1 セグメントを用いて携帯端末向けの放送を行うサービスであり、サービス名称を“ワンセグ”という [17]。通常の地上波デジタル放送に用いる符号化方式では MPEG-2 が採用されたが、携帯端末向けではより帯域幅が狭くなることなどから圧縮率の高い H.264 が採用された。

蓄積 次世代光ディスクでも H.264 の利用が進んでいる。青色レーザーを利用する次世代 DVD フォーマットで利用出来る符号化方式として MPEG-2、VC-1 と共に H.264 が採用された。他にも UMD(Universal Media Disc) で動画像を記録する規格である UMD Video でも H.264 が採用された [18]。

2.3.3 プロファイルとレベル

H.264 では多くの符号化機能が追加されたが、全ての機能に対応したエンコーダー/デコーダーの実装は困難であり、アプリケーションによっては必ずしも必要でない機能も存在する。そこで符号化機能をいくつかの集まりで定義したプロファイルが規定されている。プロファイルの考え方は MPEG-2 から採用されており、必要な機能が規定されたプロファイルを選択し、実装することで実装の複雑度を下げることには貢献している。

H.264 では現在 6 種類のプロファイルが定義されている。表 2-2 に H.264 で規定されているプロファイルを示す。

プロファイルは 6 種類あるが

- High 10 プロファイル。
- High 4:2:2 プロファイル。
- High 4:4:4 プロファイル。

は High プロファイルに対してそれぞれ 10 ビットコンポーネント対応、4:2:2 フォーマット対応、4:4:4 フォーマット対応といった機能が追加されたものであり、実質のプロファイルは大きく 4 種類であると言える。

基本となる Baseline プロファイルはテレビ電話やテレビ会議システム、携帯端末で用いられることを想定したプロファイルであり、H.264 の基本機能に加え、エラー耐性ツールが

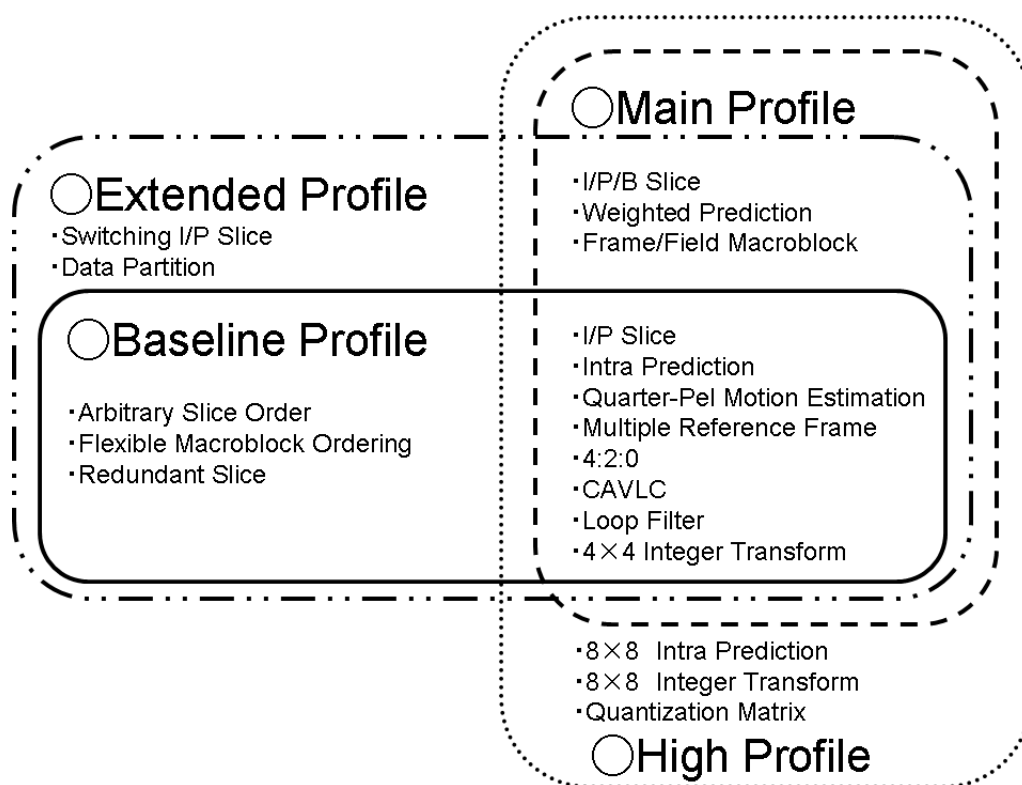


図 2-3: プロファイルの関係

追加されている。これにスイッチング I/P スライス，データ・パーティションの機能を追加したものが Extended プロファイルであり，IP ストリーミング用途を想定して規定されている。Main プロファイルは Baseline プロファイルからエラー耐性ツールが取り除かれており，B スライスや重み付け予測が利用出来る。これは蓄積メディアで利用されることを想定したプロファイルであることによる。High プロファイルは Main プロファイルに対して高精細画像の符号化の際に有効な機能が追加されている。

H.264 のプロファイルはあるプロファイルが特定のプロファイルのサブセットとなる階層構造を取っていない。これは階層構造を取っていた MPEG-2 や MPEG-4 と異なる点である。図 2-3 に各プロファイルで利用出来る符号化機能の関係を示す。また H.264 ではプロファイルとは別にデコーダの能力を示すレベルが規定されている。画像サイズやビットレート，最大フレームサイズ，デコーダで復号時に用いるバッファのサイズなどがレベルで規定される内容に当たる。

2.3.4 符号化要素技術

H.264 で新たに多くの符号化技術が追加された。本論文と関連の深い符号化機能を中心に以下説明を加える。図 2-4 に H.264 エンコーダの構成を示す。

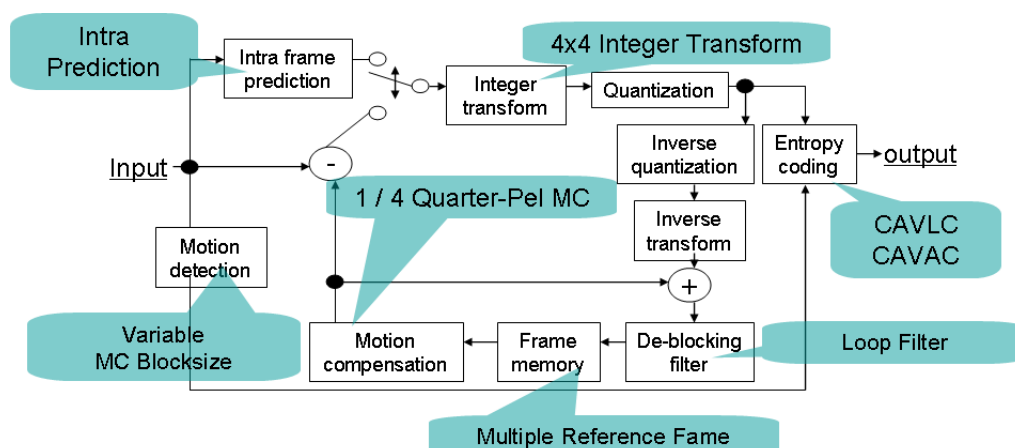


図 2-4: H.264 エンコーダの構成

Intra 予測 H.264 では画素領域において Intra 予測を行うことが出来、圧縮率の向上に大きく貢献している。全てのプロファイルで 4×4 画素単位と 16×16 画素単位の 2 種類の予測サイズが利用出来る。さらに High プロファイルでは新たに 8×8 画素単位の予測サイズも追加された。

Intra 予測は符号化対象マクロブロックの上側と左側に位置する符号化済みの画素値を用いて予測を行う。 16×16 画素単位では 4 通り、 4×4 画素単位と 8×8 画素単位では 9 通りの予測方向が定義されており、計 22 種類の予測モードが定義されている。各予測方向には利用頻度が高い順に小さい番号が付けられている。図 2-5 に予測方向とモード番号の関係を示す。符号化効率を向上させるためには発生符号量と符号化歪みを考慮した Rate-Distortion コストが最も小さくなる予測モードを選択する必要があるが、全ての予測モードを探索すると符号化演算量が増加する問題点がある。

Inter 予測 H.264 で新たに追加された Inter 予測に関連する機能を図 2-6 に示す。まず利用出来る参照フレーム数の増加がある。MPEG-2 や MPEG-4 では動き補償予測の際に参照フレームを P ピクチャで 1 枚、B ピクチャで 2 枚しかそれぞれ利用することができなかったが、H.264 では複数の参照フレームを利用出来るようになった。利用可能な参照フレーム数は利用するプロファイルとレベルにより定義されており、最大で 5 枚となっている。単純に全フレームに対して動き補償予測処理を行うと参照フレーム数に比例して符号化演算量が増加する。

さらに可変ブロックサイズ動き補償が導入された。これまで動き補償の単位は 16×16 画素単位で固定であったが、H.264 では 16×16 、 16×8 、 8×16 、 8×8 、 8×4 、 4×8 、 4×4 画素単位の 7 種類のブロックサイズが利用出来るようになった。ブロックサイズはまず 16×16 、 16×8 、 8×16 、 8×8 画素単位のうちのどのブロックサイズを利用するかを選択し、 8×8 画素単位を選択した時のみ 8×4 、 4×8 、 4×4 画素単位のブロックサイズが利用出来るように定義されている。動き補償に利用出来るブロックサイズの種類が増えたことから、全てのブロック

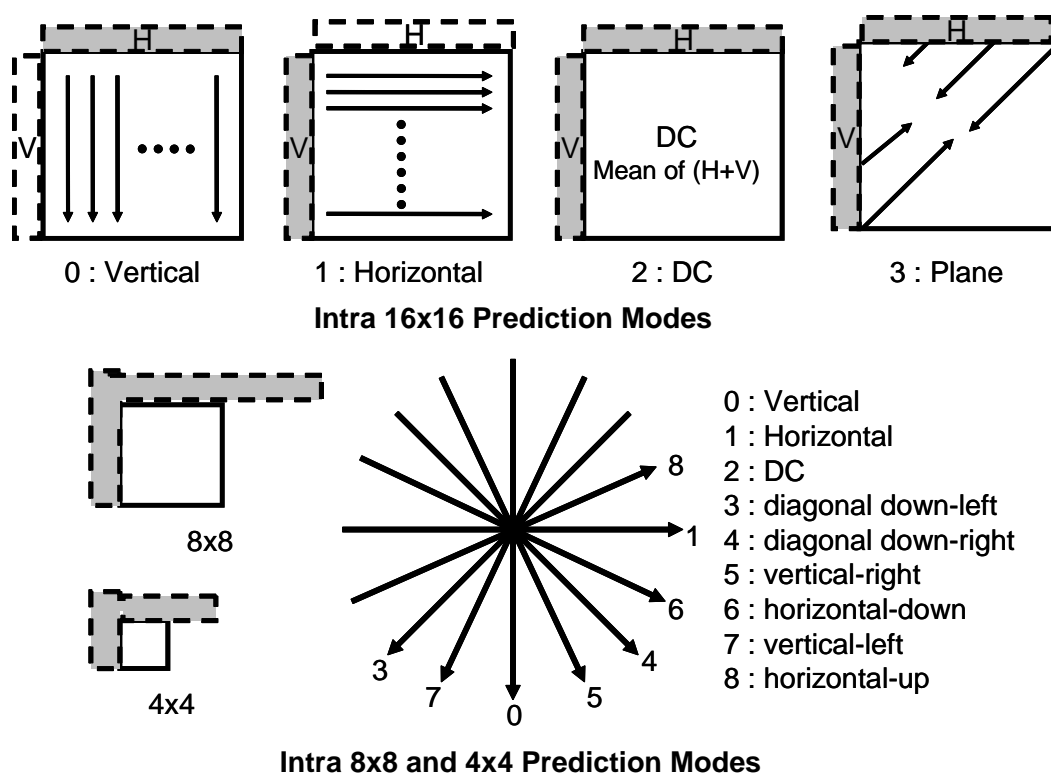


図 2-5: Intra 予測の予測方向とモード番号

サイズの組み合わせに対して動き補償予測処理を行うと符号化演算量が増加する。

最後に動き補償の精度が従来の 1/2 画素精度から 1/4 画素精度へと向上した。1/2 画素精度の画素値は 6 タップの FIR フィルタ*により計算され、1/4 画素精度の画素値は整数精度の画素値と 1/2 画素精度の画素値を用いて平均値フィルタで作成される。これに伴い従来の探索範囲と同じ範囲を探索する場合、4 倍の探索点について計算しなければならない。

整数変換 H.264 では直交変換が、これまで用いられてきた 8×8 画素単位 DCT から 4×4 画素単位の整数変換へと変更された。これには以下の利点がある。

- 演算の有効桁数が少なくなり、ハードウェア化が容易である。
- IDCT ミスマッチが発生しない。
- 符号化処理の最小単位である 4×4 画素単位と整合性がとれる。

High プロファイルでは 4×4 画素単位整数変換に加えて、8×8 画素単位整数変換も利用出来るようになっている。これは 8×8 画素単位整数変換が 4×4 画素単位整数変換と比較して高周波成分の再現に優れていることから追加された。

整数変換に用いられる行列は DCT に近い性能を発揮し、かつ 16 ビット精度の演算処理で高速演算アルゴリズムが利用出来るように設計されている。

*係数は (1, -5, 20, 20, -5, 1)/32

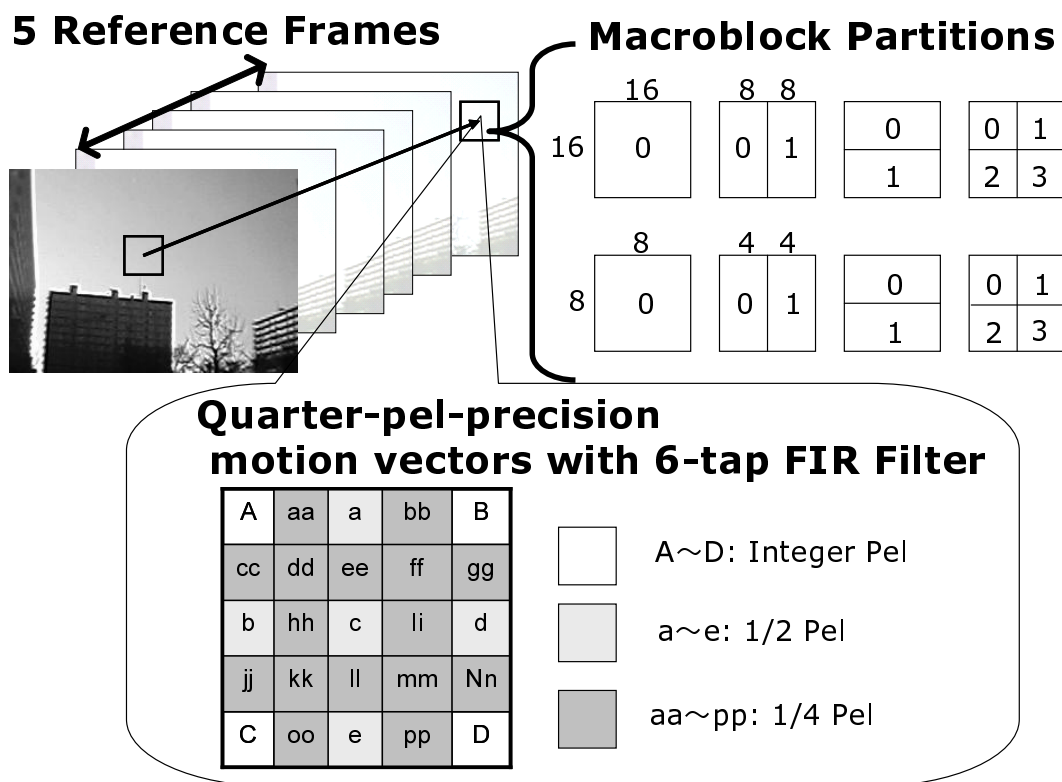


図 2-6: H.264 で Inter 予測に追加された機能

デブロッキングフィルタ 低ビットレートで符号化を行うとブロック歪みが発生し、動き補償処理を通して歪みが伝搬する問題点がある。そこでH.264ではフレームメモリに復号画像を保存する処理に先立って、ブロック歪みを軽減させるデブロッキングフィルタを適用することが出来る。デブロッキングフィルタは以下の3通りに施すことが出来る。

- ブロック境界とマクロブロック境界にデブロッキングフィルタを適用する。
- マクロブロック境界にのみデブロッキングフィルタを適用する。
- デブロッキングフィルタを適用しない。

これはブロック歪みの程度が同一画像においても箇所によって異なることに因る。デブロッキングフィルタは処理量が多く、特にデコーダに占める割合が高い。

エントロピー符号化 H.264では以下の3種類のエントロピー符号化を利用することが出来る。

- 指数ゴロム符号
- CAVLC
- CABAC

指数ゴロム符号は基本的なシンタックス要素の符号化に用いられ、CAVLCとCABACが直交変換後の係数値の符号化に用いられる。CAVLCは可変長符号化方式であり、ジグザグスキャンされた直交変換係数を係数値と連続する0の長さにより符号化を行う。CABACは符号量を理論的な限界まで圧縮することが出来る算術符号化方式であり、演算量が増加することからMainプロファイルでのみ利用可能である。

2.3.5 演算量と符号化効率への寄与

H.264の参照ソフトウェアであるJMを用いてIntra予測とInter予測が占める処理時間の割合と符号化効率への寄与を調査した[19]。実験条件を表2-3に示す。

QPが28の時の処理全体に占める処理時間の割合を図2-7に示す。参照ソフトウェアのJMでは規格に忠実に符号化処理を行うため、画像によって多少の違いは見られるがIntra予測とInter予測が処理時間に占める割合はほとんど同じであると言える。BaselineプロファイルではIntra予測が全体の1割前後、Inter予測が7割から8割を占めている。またHighプロファイルではIntra予測が全体の2割前後、Inter予測が6割前後を占めている。BaselineプロファイルとHighプロファイルでは符号化処理に要する時間が2倍近く離れていることが図2-7から分かる。

次にHighプロファイル利用時のIntra予測のモード利用割合を図2-8に示す。画像により割合が多少異なるものの、予測処理単位が小さくなるにつれて利用割合が高くなっていることが見て取れる。利用割合は4×4画素単位が6割前後、8×8画素単位が2割から3割、残りが16×16画素単位となっている。図2-7と合わせて考えると処理単位が細かいほど利用割合が高く、処理時間も多くを占める傾向があると言える。

同じくHighプロファイルにおけるInter予測のブロックサイズ利用割合を図2-9に示す。H.264ではP/BピクチャにおいてもIntra予測を利用することが出来、わずかながら利用されている。またCopyは直前フレームの同位置のマクロブロックの画素値をそのままコピーして利用するモードである。動きが多いForemanやMobile&Calendarでは8×8画素単位以下のブロックサイズが利用されることが多く、Container Ship, News, Silent, Parisといった動きが少ない画像ではCopyの割合が多いことが分かる。

表2.3.5に利用する符号化機能を制限したときの処理時間を、図2-10と図2-11にレート対歪み特性を示す。処理時間は全ての符号化機能を使用(Original)、Intra予測で利用できるサイズを16×16画素単位だけに制限(Intra16x16)、Inter予測で利用できるブロックサイズを8×8画素単位までに制限(Inter8x8)と16×16画素単位だけに制限(Inter16x16)、Intra予測とInter予測で利用できるサイズを16×16画素単位だけに制限(16x16)で比較した。Originalと16x16では2倍以上の差となっていることが分かる。表2.3.5よりIntra予測とInter予測で利用できる複数の処理単位が演算量増加の一因となっていることが見て取れる。またIntra予測とInter予測ではInter予測の方が複数の処理単位による演算量増加の割合が大きい。

レート歪み特性では曲線が左上にあるほど符号化効率が優れていることを意味するが、全ての符号化機能を利用したOriginalに最も近い性能を發揮しているのはInter予測のブロック

サイズを 16×16 画素単位から 8×8 画素単位までに制限したときであることが分かる。Intra 予測のサイズを 16×16 画素単位のみで制限した条件が続き、Inter 予測のブロックサイズを 16×16 画素単位に制限した条件、Intra 予測と Inter 予測のブロックサイズの両方を 16×16 画素単位に制限した条件が最も符号化効率が悪いという結果が出ている。このことから Intra 予測と Inter 予測で利用出来る複数の処理単位が符号化効率にも大きく影響していると言える。

最後に Baseline プロファイルと High プロファイルの符号化効率の違いを図 2-12 に示す。画像によって異なるものの High プロファイルを利用した方が Baseline プロファイル利用時と比較して PSNR で $0.5[\text{dB}]$ から $1[\text{dB}]$ ほど PSNR が改善されていることが分かる。

2.4 むすび

本章では動画像符号化技術について述べた。画像符号化技術が登場した背景と圧縮技術の必要性を述べ、符号化技術の標準化の歴史と各標準符号化方式で利用されている符号化機能を紹介した。次に本論文で扱う H.264 を取り上げ、新たに追加された符号化機能や利用用途を中心に紹介した。最後に H.264 において Intra 予測と Inter 予測が符号化効率にどの程度寄与していて、どの程度の演算量を占めているかについて基礎検討を行った。

表 2-2: H.264 で規定されているプロファイル

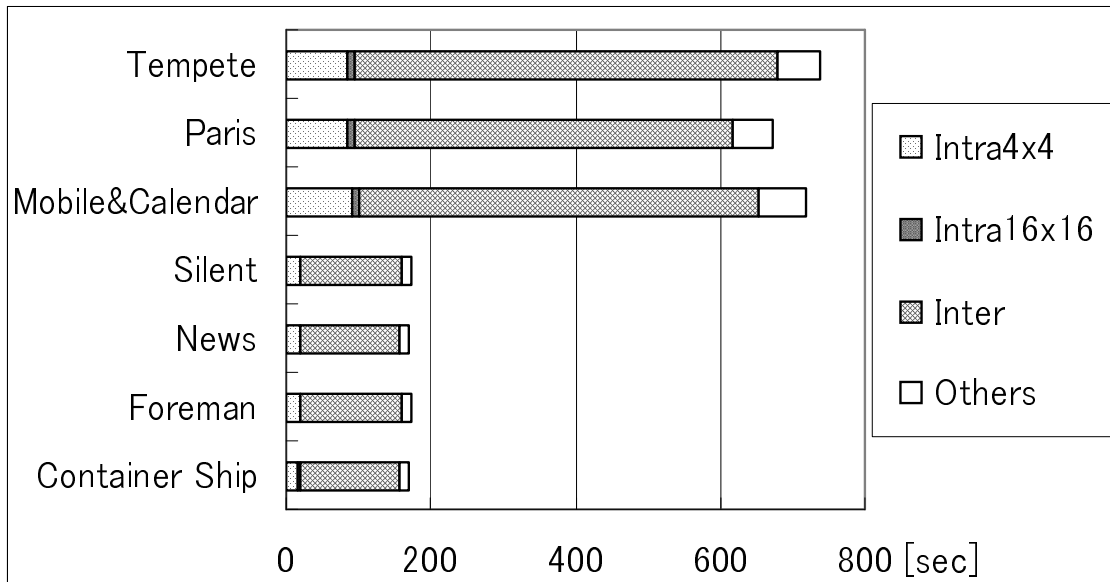
Profile Name	Application	Elemental Technology
Baseline	Mobile Application	4:2:0
	TV Phone	I/P Slice
	TV Conference	CAVLC
		Error Tolerance Tool (ASO, FMO, RS)
Main	Broadcasting	4:2:0
	Disk Media	I/P/B Slice
		CAVLC, CABAC
		Weighted Prediction
Extended	Streaming	4:2:0
		I/P/B Slice
		CAVLC
		Data Partition
		Weighted Prediction
		Switching I/P Slice
High	Studio	4:2:0, 4:0:0
		8×8 Integer Transform
		Intra8×8 Prediction
		Quantization Matrix
High 10	Studio	High Profile + 10 bit Component
High 4:2:2	Studio	High Profile + 4:2:2
High 4:4:4	Studio	High Profile + 4:4:4

CAVLC : Context-based Adaptive Variable Length Code

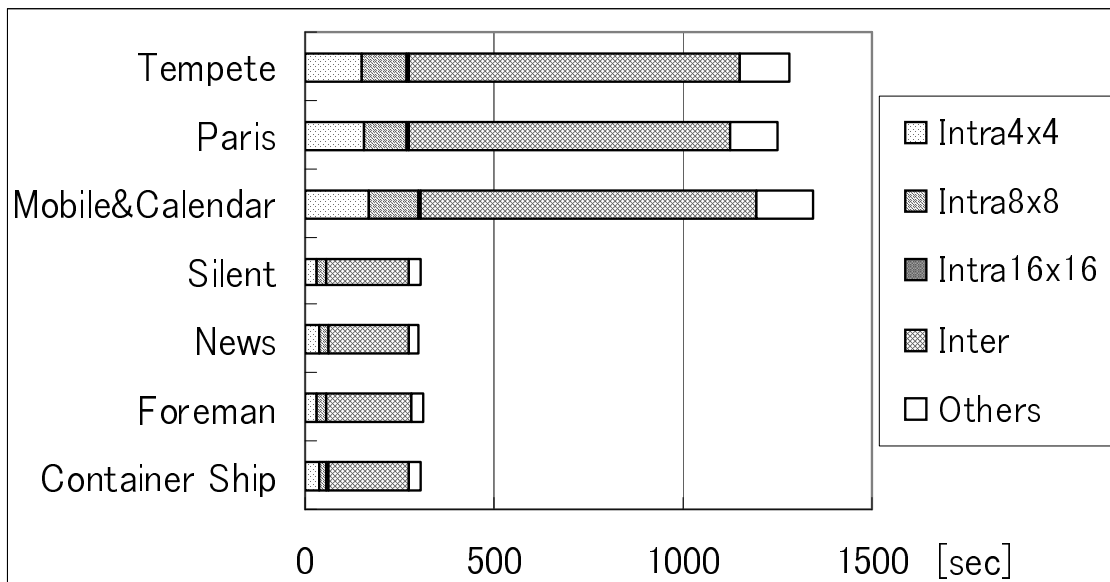
CABAC : Context-based Adaptive Binary Arithmetic Code

表 2-3: 実験条件

JM Version	10.1
Profile	Baseline & High Profile
Input Sequence	QCIF Size: Container Ship,Foreman,News,Silent CIF Size: Mobile&Calendar,Paris,Tempete
QP	I , P:22 ~ 28
Number of Frames	150
Frame Structure	N=15, M=1(Baseline Profile) N=15, M=3(High Profile)
Frame Rate	QCIF:15fps CIF:30fps
Optimization	RD-Optimization
MC Accuracy	Quarter Pel
Entropy Coding	CAVLC(Baseline Profile) CABAC(High Profile)



(a) Baseline Profile



(b) High Profile

図 2-7: H.264 符号化処理時間の割合

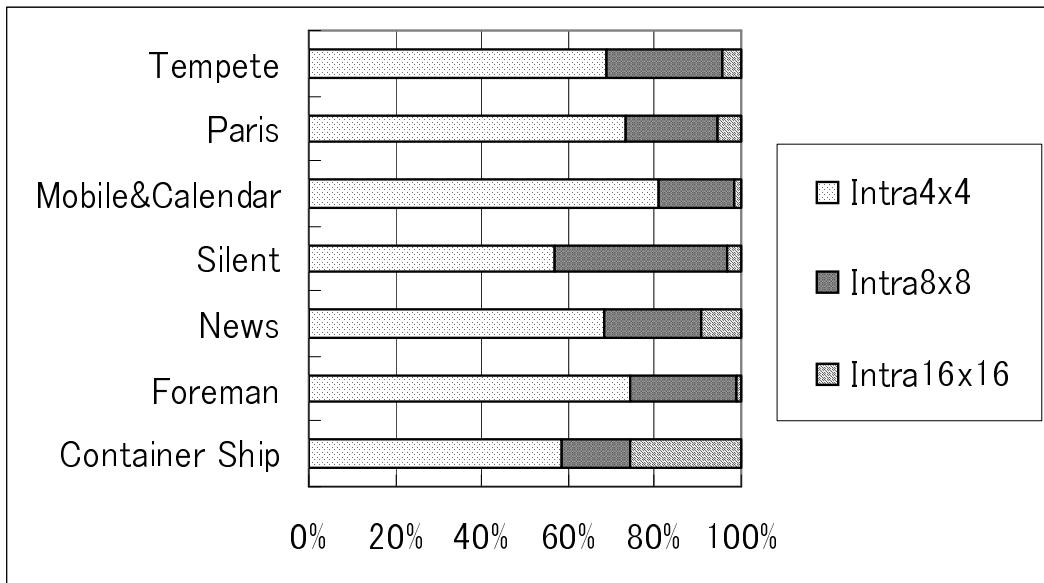


図 2-8: Intra 予測のモード利用割合

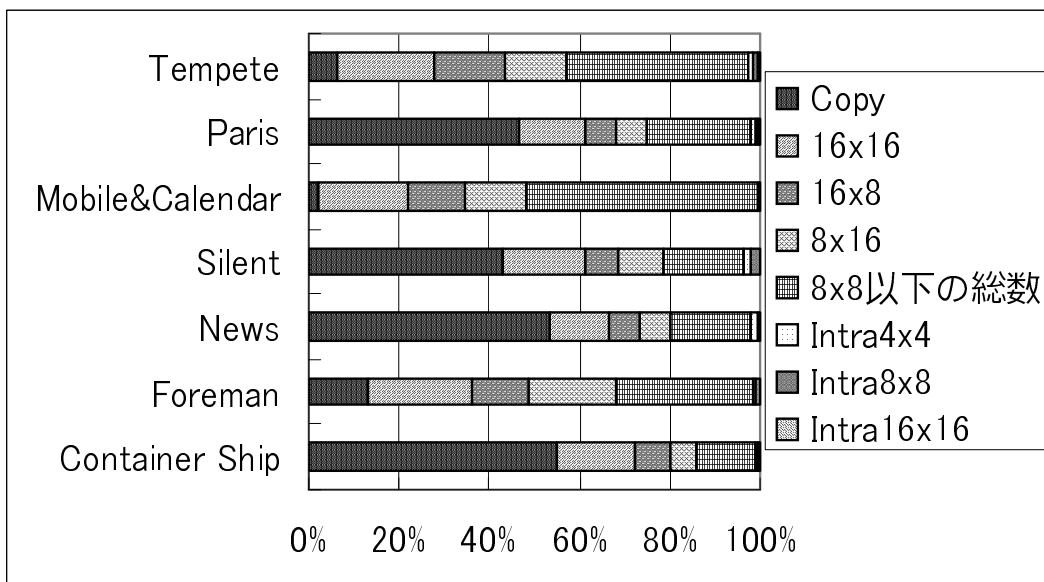


図 2-9: Inter 予測のブロックサイズ利用割合

表 2-4: 符号化機能を制限したときの処理時間

Baseline Profile

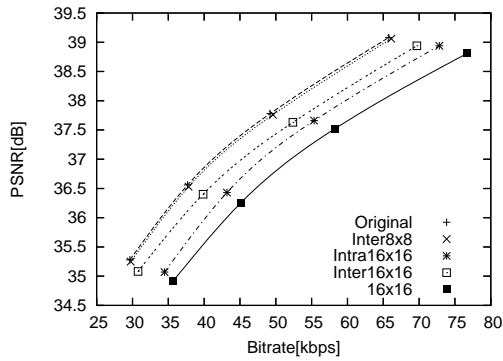
	Original	Intra16x16	Inter8x8	Inter16x16	16x16
Container Ship	183.965	162.912	146.931	95.963	76.986
Foreman	190.003	170.259	144.629	96.883	81.883
News	184.428	165.637	143.968	97.408	77.466
Silent	186.087	187.260	444.953	264.547	76.573
Mobile&Calendar	782.016	675.571	652.189	412.205	307.987
Paris	741.322	646.342	596.418	389.428	491.925
Tempete	791.876	668.416	591.885	404.947	304.814

[sec]

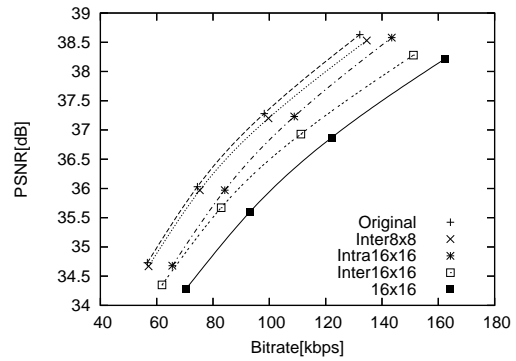
High Profile

	Original	Intra16x16	Inter8x8	Inter16x16	16x16
Container Ship	325.104	259.508	287.223	165.817	95.944
Foreman	332.421	265.393	298.035	162.640	98.204
News	333.753	259.931	290.354	163.596	95.175
Silent	341.848	261.011	287.753	161.697	97.167
Mobile&Calendar	1476.280	1122.199	1270.530	775.968	395.756
Paris	1354.336	1069.280	1330.172	691.695	462.705
Tempete	1399.359	1063.246	1287.283	700.998	385.048

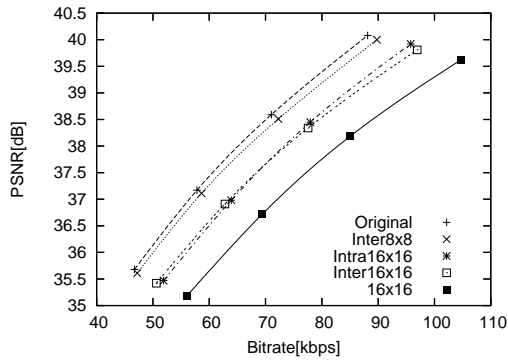
[sec]



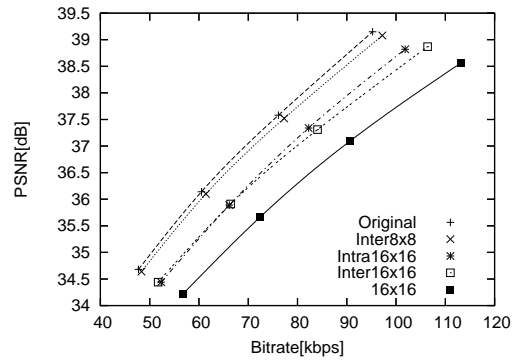
(a) Container Ship



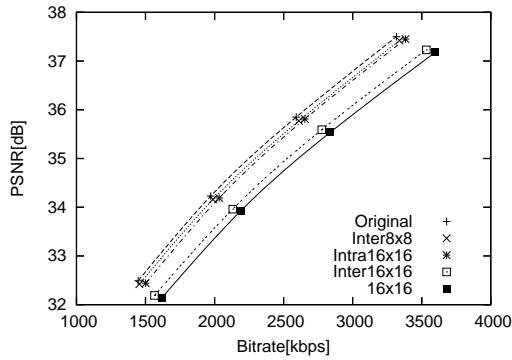
(b) Foreman



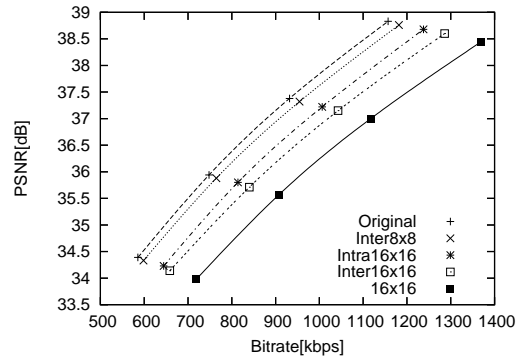
(c) News



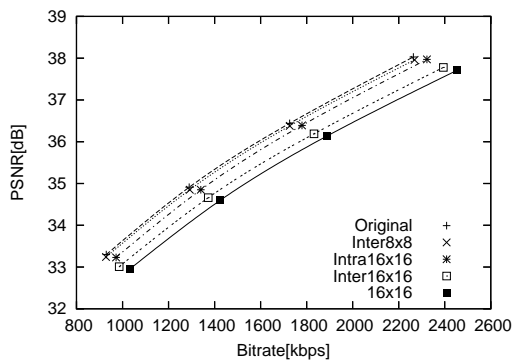
(d) Silent



(e) Mobile&Calendar

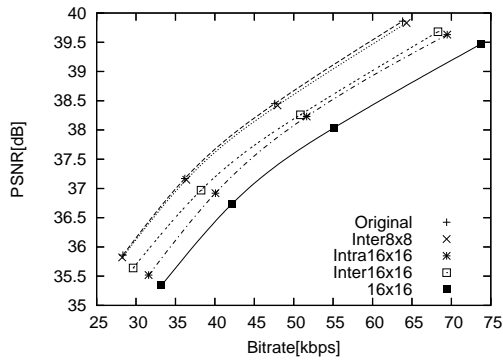


(f) Paris

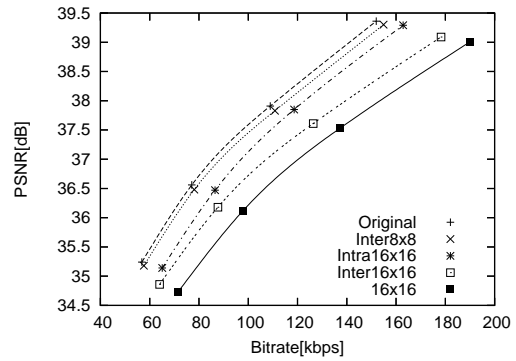


(g) Tempete

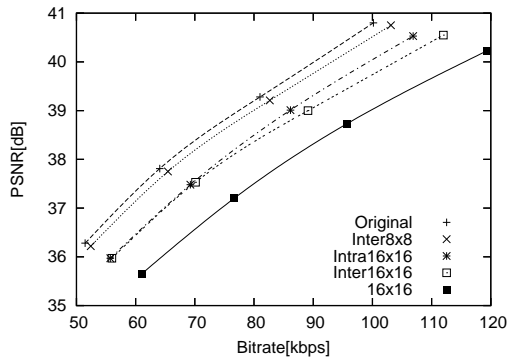
図 2-10: 各機能の符号化効率への寄与 (Baseline プロファイル)



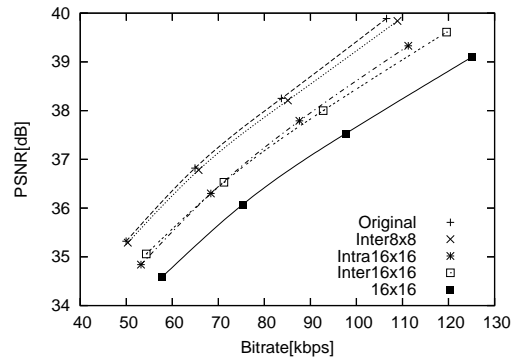
(a) Container Ship



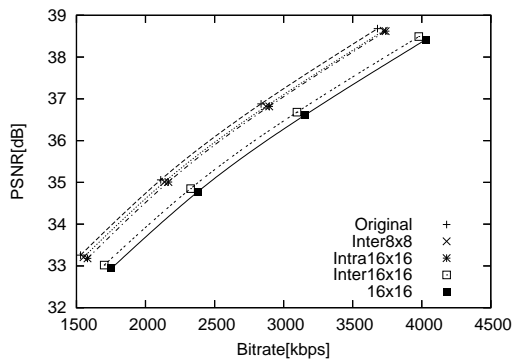
(b) Foreman



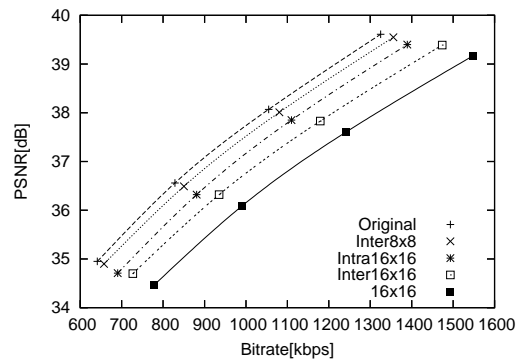
(c) News



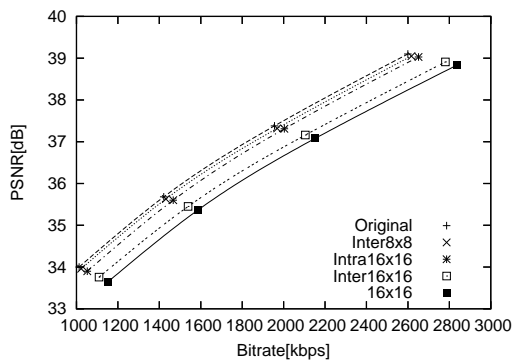
(d) Silent



(e) Mobile&Calendar

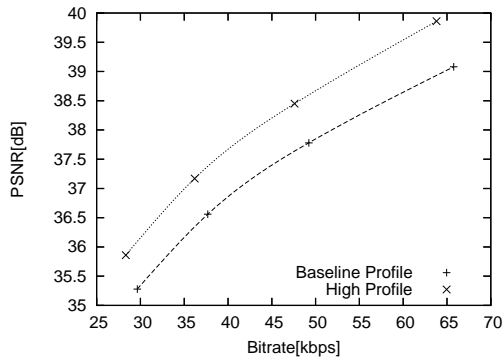


(f) Paris

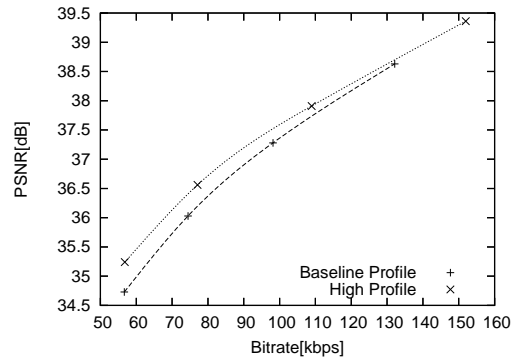


(g) Tempete

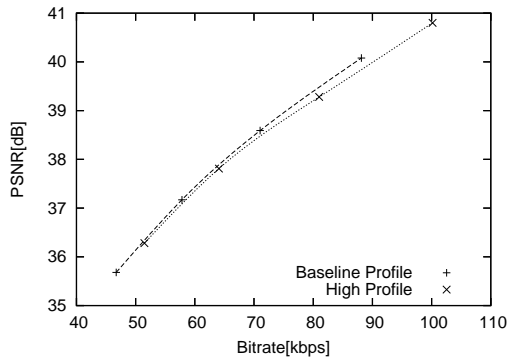
図 2-11: 各機能の符号化効率への寄与 (High プロファイル)



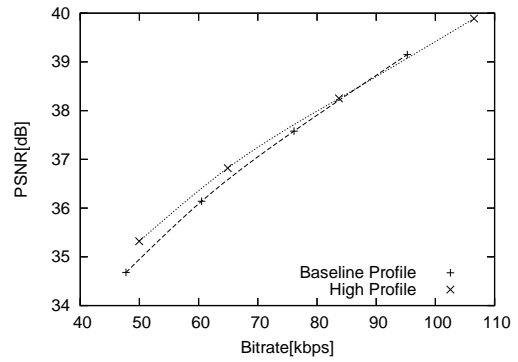
(a) Container Ship



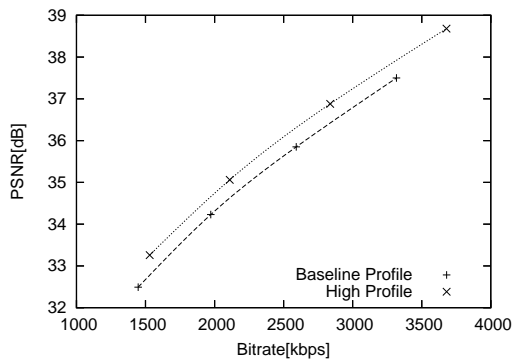
(b) Foreman



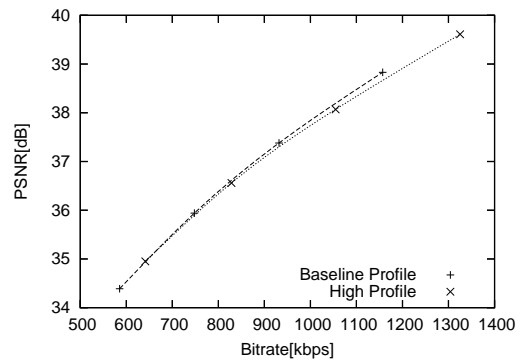
(c) News



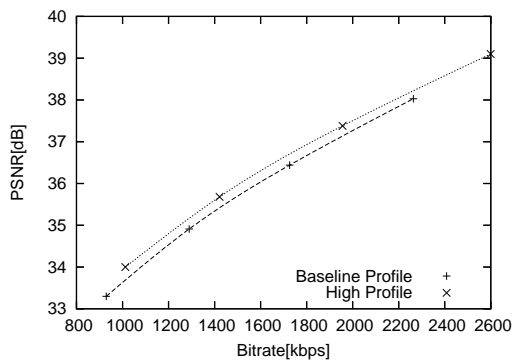
(d) Silent



(e) Mobile&Calendar



(f) Paris



(g) Tempete

図 2-12: Baseline プロファイルと High プロファイルの符号化効率比較

第3章

Intra 予測処理の演算量削減手法

3.1 まえがき

本章では Intra 予測の演算量削減手法について述べる。H.264 では符号化効率を高めるために新たに画面内で画素値の予測を行う Intra 予測が導入されたが、符号化演算量増加の一因となっている。始めに Intra 予測の演算量削減を行う従来手法を紹介し、次に本論文で提案する複数の処理サイズの違いを用いた Intra 予測の符号化演算量削減手法を述べる。提案手法では 4×4 画素単位の Intra 予測モード結果を利用して 8×8 と 16×16 画素単位の Intra 予測処理を制限する。最後に符号化演算量の削減度合いと符号量、PSNR の観点から提案手法の評価を行う。

3.2 Intra 予測処理演算量削減の従来手法

2.3.5 の検討結果より、Intra 予測は符号化処理全体の 1 割から 2 割を占めることが分かった。演算量は処理単位が小さくなるほど多くの割合を占めるようになり、 4×4 画素単位の演算量が最も多く、 8×8 画素単位、 16×16 画素単位の順に演算量が小さくなる。このことから、これまでに提案されてきた Intra 予測の演算量削減手法の多くは 4×4 画素単位の予測処理を削減するものが多い。

文献 [20] ではエッジ検出を利用した Intra 予測の 4×4 画素単位処理の高速化手法を提案している。Intra 予測の予測方向と画像中のエッジの関連性に着目し、始めにエッジ検出を行うことで探索する Intra 予測モードを削減する。また文献 [21] では予測画素値との SAD (Sum of Absolute Difference) が最小の予測モードを選択する手法を提案している。Intra 予測の演算量は Intra 予測の予測モードの数と RD 最適化の演算量が多い点に寄るところが大きいことから、RD 最適化によるモード選択ではなく、SAD が最小となる予測モードを選択する手法をとっている。

3.3 Intra 予測処理演算量削減の提案手法

参照ソフトウェアの JM では 4×4 画素単位と 8×8 画素単位の予測処理では RD 最適化を行ってモード決定を行い、 16×16 画素単位の予測処理では SAD 最小の予測モードを採用す

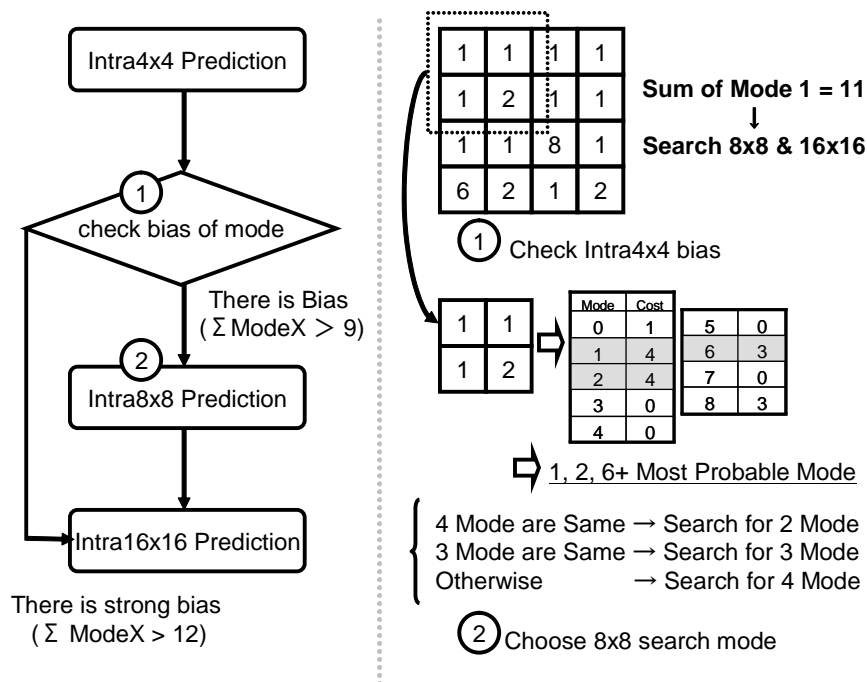


図 3-1: Intra 予測の提案手法

るようにしている [19] . 16×16 画素単位の予測処理でも RD 最適化によるモード決定を行えば演算量はさらに増加することが考えられる . また High プロファイルで利用出来る 8×8 画素単位の演算量も 4×4 画素単位予測処理に近い割合を占める .

Intra 予測では 4×4 画素単位の利用割合が最も多く , 利用割合が少ない 8×8 画素単位と 16×16 画素単位の予測処理を削減する手法が良いと言える . 各マクロブロック中の 4×4 画素単位の予測結果モードが 0 , 1 , 2 のいずれかとなったブロック数の総計と , JM において内部でモード決定と予測サイズ決定に利用されるコスト値の関係について調査した結果 , 4×4 画素単位の予測結果が特定のモードに偏ると 16×16 画素単位のコスト値が小さくなることがわかった [22] . これは 8×8 画素単位のコスト値にも見られる傾向である . そこで 4×4 画素単位の予測結果が特定のモードに偏ったかどうかを調査することで , 8×8 画素単位と 16×16 画素単位の予測処理を削減する手法を提案する . 提案手法を図 3-1 に示す .

提案手法では初めに 4×4 画素単位の予測処理を行い , 以降の探索は 4×4 画素単位の予測結果モードの偏りを見て判断する . ある特定のモードが 8 つ以上選択された場合のみ 16×16 画素単位の予測処理を行う . 4×4 画素単位と 16×16 画素単位の予測処理では予測方向の数が異なるため , 同じ予測方向で利用割合が大きい 0 , 1 , 2 と 3~8 で判断を分ける . どのモードに偏っているかで 16×16 画素単位での予測モードを選択する . 具体的には以下のように選択する .

- モード 0 に偏り モード 0 , 2 , 3 を探索する .
- モード 1 に偏り モード 1 , 2 , 3 を探索する .

表 3-1: 8×8 探索モード決定に用いるコスト値

Intra4x4 Mode	Intra8x8 Search Mode
0	0,2,5,7
1	1,2,6,8
2	0,1,2
3	2,3,7,8
4	2,4,5,6
5	0,2,4,5
6	1,2,4,6
7	0,2,3,7
8	1,2,8

- モード2に偏り 4種類のモードを探索する。

モード0とモード1は90度異なる予測方向であり、片方に強く偏りが見られるときにはもう片方の利用割合が小さいことから探索モードから除外する。逆に4×4画素単位の予測結果で特定のモードが8つ以上選択された無かった時は16×16画素単位の予測を行わない。これにより16×16画素単位が本来最適であるマクロブロックの8割を処理することが出来る。

Highプロファイルで8×8画素単位の予測が利用出来る場合も同じく4×4画素単位の予測結果を利用する。特定のモードが12以上選択された場合、8×8画素単位の探索を省略して16×16画素単位の処理のみを行う。特定のモードが12以上選択されず8×8画素単位の処理を行う場合、8×8画素単位のブロックサイズに対応する位置にある4×4画素単位の予測結果4つのモードの偏りを見て2から4つのモードを探索する。探索するモードの決定は表3-1を用いる。4×4画素単位の予測結果1つに対して表3-1の右に挙げられた8×8画素単位の予測モードのコスト値を1足すようにし、コストが大きい方から順に選択する。探索モード数は4×4画素単位の予測結果の偏りの強さに応じて変化させるようにし、4つとも同じモードの時は2つのモードを、3つが同じモードの時は3つのモードをそれぞれ探索する。

3.4 提案手法の評価

3.4.1 実験条件

H.264の参照ソフトウェアであるJMに提案手法を実装し性能の確認を行った[19]。実験条件を表3-2に示す。実験はオリジナルのJM(Original)、提案手法(Intra Proposed)、Intra予測のブロックサイズを16×16だけに限定したもの(Intra16×16)の3つに対してそれぞれ行った。

表 3-2: 実験条件

JM Version	10.1
Profile	Baseline & High Profile
Input Sequence	QCIF Size: Container Ship,Foreman,News,Silent CIF Size: Mobile&Calendar,Paris,Tempete
QP	I, P:22 ~ 28
Number of Frames	150
Frame Structure	N=15, M=1(Baseline Profile) N=15, M=3(High Profile)
Frame Rate	QCIF:15fps CIF:30fps
Optimization	RD-Optimization
Entropy Coding	CAVLC(Baseline Profile) CABAC(High Profile)

3.4.2 実験結果

QP を 28 に設定した時の Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化を表 3-3 に, High プロファイル使用時の PSNR, 符号量, 演算量の変化を表 3-4 にそれぞれ示す.

また QP を変化させて取得した実験結果を元に作成したレート歪み特性を示す. 図 3-2 に Baseline プロファイル使用時の結果を, 図 3-3 に High プロファイル使用時の結果を示す.

3.4.3 考察

表 3-3 と表 3-4 より提案手法を用いることで PSNR の低下と符号量の増加を抑えつつ演算量を削減できることが分かる. Intra 予測に利用できる処理単位を 16×16 画素単位だけに制限するとヘッダ量が減少するため符号量は小さくなっているが残差信号成分が増加するため PSNR も低下してしまっている. 演算量の変化はオリジナル JM の演算量を 100%としたときの提案手法における演算量を表している. 今回は探索したモード数を値として用いた. 演算量は 8×8 画素単位の処理が 35%から 40%, 16×16 画素単位の処理が 35%から 70%に削減できた.

また図 3-2 と図 3-3 から提案手法は全探索に近い符号化性能を発揮していることが分かる.

3.5 むすび

本章では Intra 予測の演算量削減手法について述べた。Intra 予測演算量削減の従来手法を紹介し、本章で提案する複数の処理サイズの違いを活用した Intra 予測の符号化演算量削減手法を述べた。提案手法では 4×4 画素単位の予測モード結果を利用して 8×8 と 16×16 画素単位の予測処理を制限する。提案手法の性能評価を符号量、PSNR の観点から評価を行い、提案手法を用いることで符号化効率を大きく低下させることなく、 8×8 画素単位の演算量を約 40%、 16×16 画素単位の演算量を 35% から 70% の範囲で削減出来ることを示した。

表 3-3: Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化

Container Ship			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.56	37.69	-
Intra16x16	36.43	43.18	-
Intra Proposed	36.54	37.85	74.94

Foreman			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.03	74.45	-
Intra16x16	35.97	84.15	-
Intra Proposed	36.03	74.30	51.43

News			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	37.17	57.82	-
Intra16x16	36.98	63.91	-
Intra Proposed	37.20	57.84	62.70

Silent			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.14	60.5	-
Intra16x16	35.89	66.17	-
Intra Proposed	36.15	60.63	37.31

Mobile&Calendar			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	34.23	1972.85	-
Intra16x16	34.19	2033.59	-
Intra Proposed	34.23	1974.94	44.38

Paris			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	35.94	747.96	-
Intra16x16	35.80	813.97	-
Intra Proposed	35.95	748.56	62.11

Tempete			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	34.91	1289.62	-
Intra16x16	34.85	1340.04	-
Intra Proposed	34.91	1290.33	55.63

表 3-4: High プロファイル使用時の PSNR, 符号量, 演算量の変化

Container Ship				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	37.17	36.19	-	-
Intra16x16	36.92	40.07	-	-
Intra Proposed	37.15	36.02	73.1	35.09

Foreman				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	36.56	77.09	-	-
Intra16x16	36.47	86.55	-	-
Intra Proposed	36.55	76.95	46.27	39.36

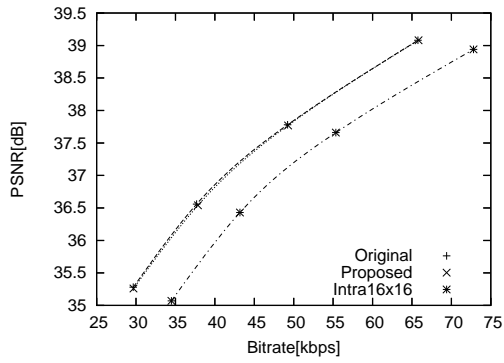
News				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	37.81	64.07	-	-
Intra16x16	37.48	69.25	-	-
Intra Proposed	37.8	63.91	58.96	37.66

Silent				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	36.82	64.92	-	-
Intra16x16	36.3	68.37	-	-
Intra Proposed	36.8	65.18	34.45	41.32

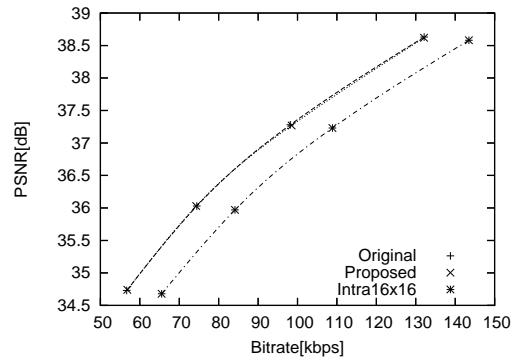
Mobile&Calendar				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	35.06	2109.75	-	-
Intra16x16	35.01	2166.06	-	-
Intra Proposed	35.06	2109.35	40.12	41.51

Paris				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	36.56	828.61	-	-
Intra16x16	36.32	881.02	-	-
Intra Proposed	36.55	828.23	58.14	37.49

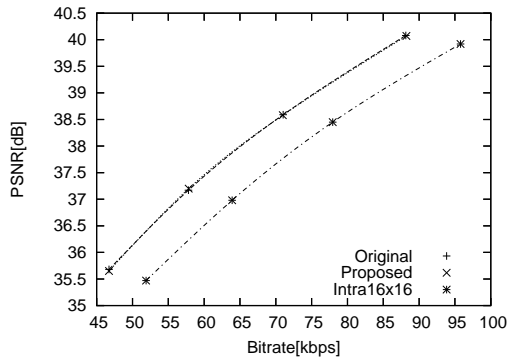
Tempete				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra16x16	Intra8x8
Original	35.68	1421.44	-	-
Intra16x16	35.6	1467.09	-	-
Intra Proposed	35.68	1424.4	53.54	39.45



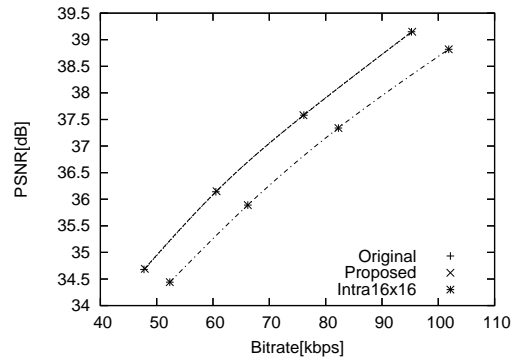
(a) Container Ship



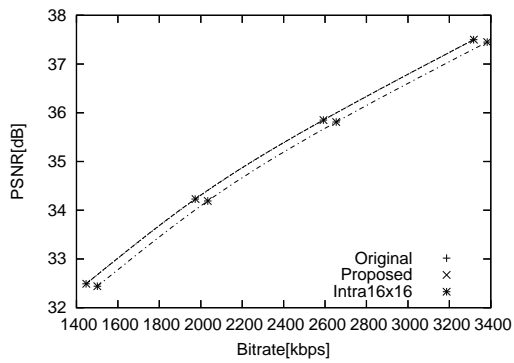
(b) Foreman



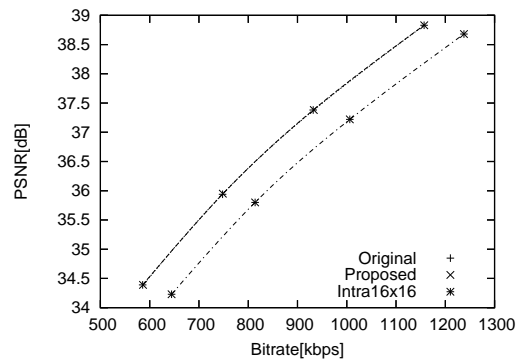
(c) News



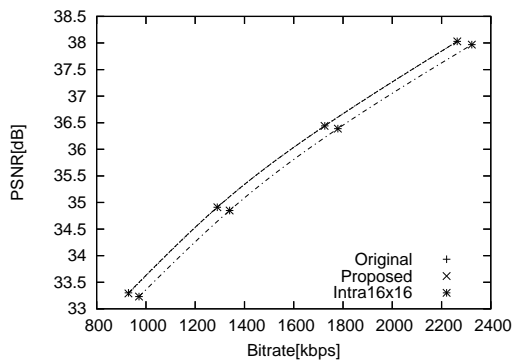
(d) Silent



(e) Mobile&Calendar

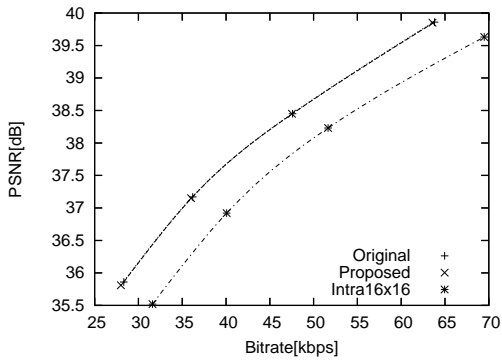


(f) Paris

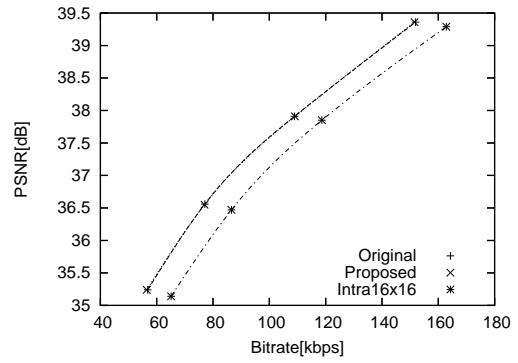


(g) Tempete

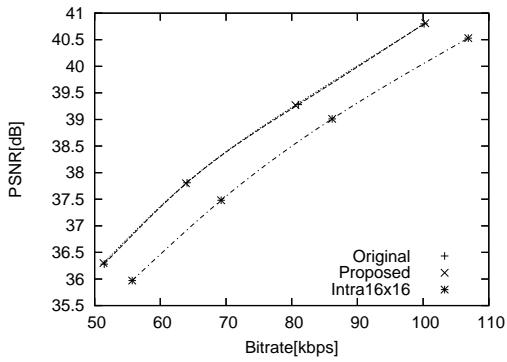
図 3-2: Baseline プロファイル使用時のレート対歪み特性



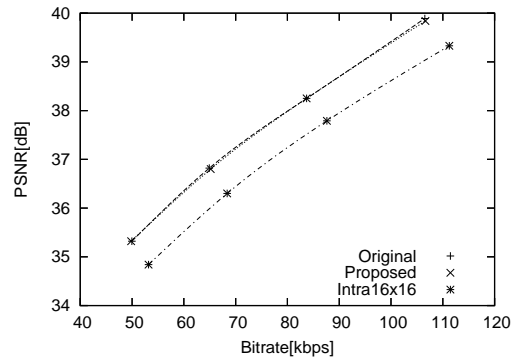
(a) Container Ship



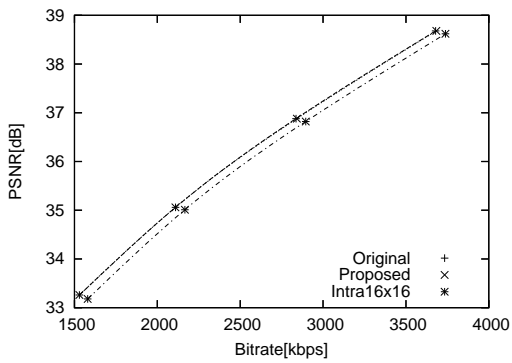
(b) Foreman



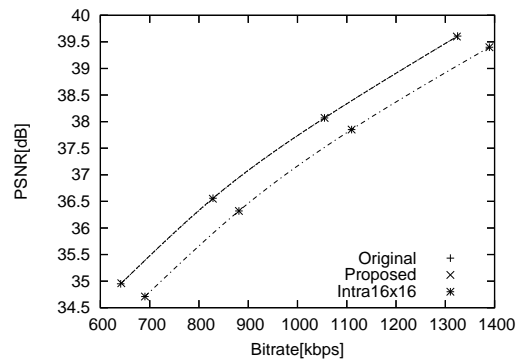
(c) News



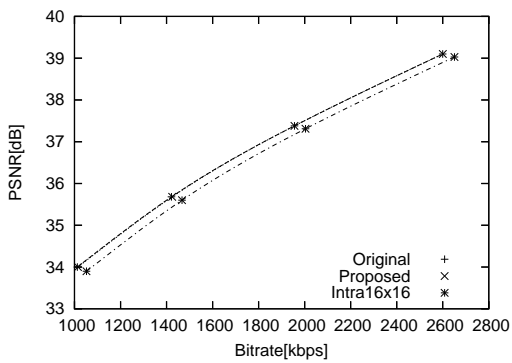
(d) Silent



(e) Mobile&Calendar



(f) Paris



(g) Tempete

図 3-3: High プロファイル使用時のレート対歪み特性

第4章

Inter 予測処理の演算量削減手法

4.1 まえがき

本章では Inter 予測の演算量削減手法について述べる。Inter 予測は H.264 以前の符号化方式でも利用されてきたが、符号化効率を高めるために複数参照フレームや可変ブロックサイズ動き補償、1/4 画素精度動き補償などの機能が追加され符号化演算量増加の一因となっている。始めに Inter 予測の演算量削減の従来手法を紹介し、次に本論文で提案する Inter 予測の符号化演算量削減手法を述べる。提案手法では符号化対象のマクロブロックと探索中に参照するマクロブロックで画素値の差分を取り、この情報を利用することで探索する動き補償ブロックサイズの種類を効率よく削減する。最後に符号化演算量の削減度合いと符号量、PSNR の観点から提案手法の評価を行う。

4.2 Inter 予測処理演算量削減の従来手法

2.3.5 の検討結果より、Inter 予測は符号化処理全体の 6 割から 8 割を占めることが分かった。演算量は処理単位が小さくなるほどに多く占めるようになり、 4×4 画素単位の演算量が最も多く、 8×8 画素単位、 16×16 画素単位の順に演算量が小さくなっていた。これまでに提案されてきた Inter 予測の演算量削減手法は大きく以下の 3 つに分けられる。

- 動きベクトル探索の演算量削減手法。
- 複数参照フレーム選択に関する演算量削減手法。
- 可変ブロックサイズの選択に関する演算量削減手法。

またこれらとは別にハードウェア実装による高速化も検討されている。

動きベクトル探索の演算量削減手法は MPEG-2 の時代から検討されており、縮小画像を用いた階層探索や探索点を飛び飛びに選択するダイヤモンドサーチといった手法が提案されている [23]。H.264 では参照ソフトウェアの JM にも実装されている FastME がよく知られており、ダイヤモンドサーチをさらに発展させた Unsymmetric-Cross Multi-Hexagon Grid Search という探索点の選択手法をとる [8]。他にもブロックサイズ毎に探索点の選び方を変

える手法 [24] や、縮小画像で符号化した結果得られた情報を用いて符号化を行う手法が提案されている [25] .

参照フレームの選択では確率モデルに基づき、探索フレームの選択を行う手法 [26] や、直前フレームとの動き補償結果を用いて閾値処理を行い、残りの参照フレームの探索を行うかを決定する手法 [27] が提案されている．参照フレームに関する演算量削減では各フレーム毎に動きベクトル情報を蓄えて探索範囲を絞る手法などもある [28] .

可変ブロックサイズの選択手法ではブロックサイズの違いに着目したものが多く、中でも 4×4 画素単位や 8×8 画素単位といった小さいブロックサイズの動き補償結果から動きベクトルを見て、大きいブロックサイズへと統合するアプローチが多い [29–31, 33] . またちょうど中間のサイズに当たる 8×8 画素単位の予測結果からブロックサイズを分割、統合するアプローチもある [34] . ブロックサイズの違いを利用しない可変ブロックサイズ選択手法ではブロックサイズの分割方向が画像のエッジ方向に沿うと仮定し、エッジ抽出を行いブロックサイズ選択に利用する手法がある [35] .

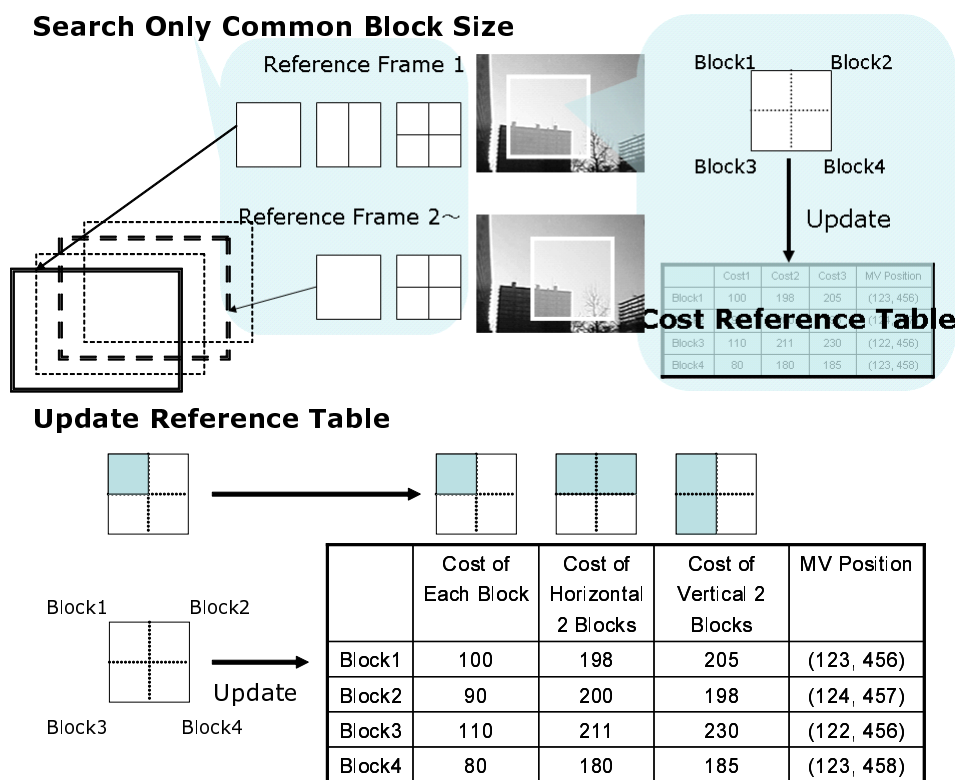
4.3 Inter 予測処理演算量削減の提案手法

16×16 画素単位の動き予測結果からブロックサイズを分割しながら探索していく手法を用いた場合以下の利点がある .

- 削減出来る探索回数が多い .
- 高い精度の動きベクトルが期待出来る .

Inter 予測の演算量の多くはブロックサイズが小さくなるほど探索回数が多くなるため、全体の演算量に占める割合が増える . 分割しながら探索していく手法では早い段階で処理を打ち切った際の削減出来る演算量が多くなり、またブロックサイズが小さい程、ブロックマッチングを用いた際の動きベクトルの探索精度は低くなる . よって本論文ではブロックサイズを分割しながら動き補償を行う手法を提案する . 提案する探索ブロックサイズ選択手法を図 4-1 に示す . 16×16 画素単位から探索を開始し、探索中のブロックサイズの $1/4$ となるブロックサイズ単位 (16×16 画素単位の探索時は 8×8 画素単位、 8×8 画素単位の探索時は 4×4 画素単位) ごとに、それぞれの箇所において画素値差分の絶対和である SAD の値と動きベクトルの大きさから求めるコスト値の和が最小となる探索位置を参照テーブルに記録する . ある程度の数の探索点を探索した後、参照テーブルに記録されたコスト値が最小となる位置の揃い方を見てその後の探索ブロックサイズを選択する . 4 箇所が同じ位置を指している場合は現在探索しているブロックサイズで探索処理を打ち切る . 縦横どちらかに 2 箇所ずつ揃っている場合は揃っている向きに合わせて 2 つに分割したサイズの探索を行う . 位置が揃っているかどうかは整数画素精度で判断した . なお以降の実験では 100 箇所探索する毎にこの処理を加えた .

探索中のブロックサイズにおいて全ての範囲の探索が終了した場合、以下の基準に従って以降の処理で探索するブロックサイズの選択を行う . 参照テーブルで 3 つまたは 2 つの位置



Search Block Size Decision

Number of blocks where coordinates are same	Processing
4	Quit Search Process
3	Search 16x8, 8x16 and 8x8
2	Search 16x8 or 8x16
None	Search 8x8 only

図 4-1: Inter 予測の提案手法

が揃っている場合、全ての種類のブロックサイズを探索する。参照テーブルで指す位置が4つバラバラの場合は探索中のブロックサイズの1/4となるブロックサイズの探索のみを行う。また2分割の判断精度を向上させるため、1/4のブロックサイズ単位から計算した最小コストと、参照テーブルに記録された2分割単位のコストを比較し、もし2分割単位のコストの方が小さければ2分割のブロックサイズを探索する。16×16画素単位から8×8画素単位への分割、8×8画素単位から4×4画素単位への分割は同じ手法を用いる。

2枚目以降の参照フレームを探索する際も同様のブロックサイズ選択処理を行うが、1枚目に探索を行った基準フレームにおいて選択されなかったブロックサイズについては探索を行わない。

マクロブロック内のエッジに基づく分割では符号化効率と無関係にブロックサイズが分割され動き補償の性能が低下することが考えられる。図4-2にMobile&CalendarのPピクチャにおけるブロックサイズ分割形状を示す。図4-2より分かるように、最適なブロックサ



図 4-2: Mobile&Calendar におけるブロックサイズ分割形状

イズ分割形状は必ずしも画像のエッジ方向に沿わないことがある．提案手法ではブロックサイズ分割の判定に画素値差分の SAD と動きベクトルから求めるコスト値を用い，PSNR が高くなるようにブロックサイズが分割されるようにする．

4.4 提案手法の評価

4.4.1 実験条件

H.264 の参照ソフトウェアである JM に提案手法を実装し性能の確認を行った [19]．実験条件を表 4-1 に示す．実験はオリジナルの JM(Original)，Inter 予測のブロックサイズを 16×16 画素単位だけに限定したもの (Inter 16×16) と 8×8 画素単位までに制限したもの (Inter 8×8)，全てのブロックサイズが利用出来る条件で提案手法を適用したもの (Inter Proposed) と 8×8 画素単位までに制限した条件で提案手法を適用したもの (Inter 8×8 Proposed) の 5 つに対してそれぞれ行った．

4.4.2 実験結果

QP を 28 に設定した時の Baseline プロファイル使用時の PSNR，符号量，演算量の変化を図 4-2 に，High プロファイル使用時の PSNR，符号量，演算量の変化を図 4-3 にそれぞれ

表 4-1: 実験条件

JM Version	10.1
Profile	Baseline & High Profile
Input Sequence	QCIF Size: Container Ship,Foreman,News,Silent CIF Size: Mobile&Calendar,Paris,Tempete
QP	I, P:22 ~ 28
Number of Frames	150
Frame Structure	N=15, M=1(Baseline Profile) N=15, M=3(High Profile)
Frame Rate	QCIF:15fps CIF:30fps
MC Block Size	16×16 ~ 4×4
Number of Reference Frames	5
Search Range	16
Optimization	RD-Optimization
MC Accuracy	Quarter Pel
Entropy Coding	CAVLC(Baseline Profile) CABAC(High Profile)

れ示す。

またQPを変化させて取得した実験結果を元に作成したRD曲線を示す。図4-3にBaselineプロファイル使用時の結果を、図4-4にHighプロファイル使用時の結果を示す。

4.4.3 考察

表4-2と4-3より提案手法を用いることでPSNRの低下と符号量の増加を抑えつつ演算量を削減できることが分かる。演算量の変化はオリジナルJMの演算量を100%としたときの提案手法における演算量を表している。今回は探索したブロックサイズ数を値として用いた。演算量は全てのブロックサイズを利用出来る条件下で8%から32%に、8×8画素単位までに制限した条件下で40%から60%に削減することができた。全てのブロックサイズを利用できる条件の方が削減度合いが大きい結果は処理の打ち切りによる効果であると考えられる。提案手法では16×16画素単位、8×8画素単位で以降の処理を打ち切りにする分岐処理が入っているが、より大きい処理単位で以降の処理の打ち切りを行うと演算量削減度合いが大きくなる。利用できるブロックサイズが増えるほど、提案手法による演算量削減度合いは大きくなることが予想される。

また図 3-2 と図 3-3 から提案手法は全探索に近い符号化性能を発揮していることが分かる。

4.5 むすび

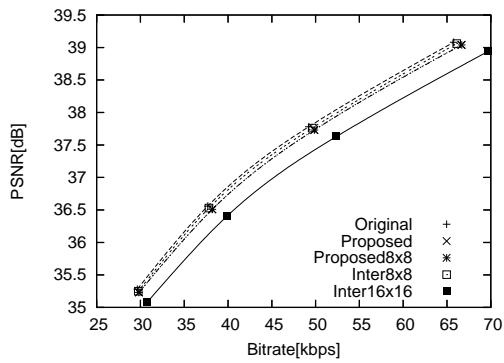
本章では Inter 予測の演算量削減手法について述べた。これまでに提案されている Inter 予測の演算量削減手法を述べ、本章で提案する複数の処理サイズの違いを活用した Inter 予測の符号化演算量削減手法を述べた。提案手法では符号化対象のマクロブロックと探索中に参照するマクロブロックで画素値の差分をとり、用いることで探索する動き補償ブロックサイズを効率よく削減する。提案手法の性能評価を符号量、PSNR の観点から評価を行い、提案手法を用いることで符号化効率を大きく低下させることなく、Inter 予測で探索するブロックサイズの種類を全てのブロックサイズを利用出来る条件下で 8% から 32% に、 8×8 画素単位までに制限した条件下で 40% から 60% に削減することができた。

表 4-2: Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化

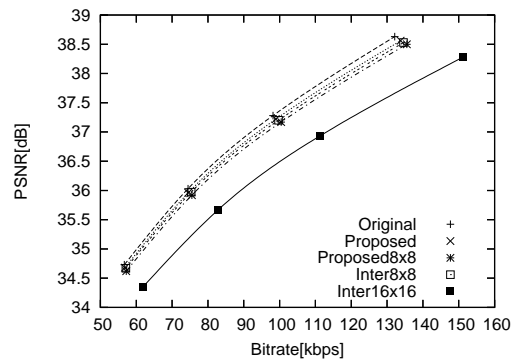
Container Ship			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.56	65.76	-
Inter16x16	36.40	39.85	-
Inter8x8	36.53	37.83	-
Inter8x8 Proposed	36.51	38.21	38.49
Inter Proposed	36.51	38.21	9.43
Foreman			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.03	74.45	-
Inter16x16	35.67	82.87	-
Inter8x8	35.97	75.31	-
Inter8x8 Proposed	35.92	75.52	63.44
Inter Proposed	35.96	74.52	34.91
News			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	37.17	57.82	-
Inter16x16	36.91	62.78	-
Inter8x8	37.11	58.63	-
Inter8x8 Proposed	37.09	58.66	42.91
Inter Proposed	37.12	58.10	14.65
Silent			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.14	60.5	-
Inter16x16	35.91	66.44	-
Inter8x8	36.10	61.41	-
Inter8x8 Proposed	36.08	61.38	47.93
Inter Proposed	36.11	60.83	19.47
Mobile&Calendar			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	34.23	1972.85	-
Inter16x16	33.96	2131.15	-
Inter8x8	34.16	1988.40	-
Inter8x8 Proposed	34.13	2005.12	59.36
Inter Proposed	34.16	1993.81	26.46
Paris			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	35.94	747.96	-
Inter16x16	35.71	840.12	-
Inter8x8	35.88	764.64	-
Inter8x8 Proposed	35.86	767.60	48.89
Inter Proposed	35.89	755.83	20.16
Tempete			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	34.91	1289.62	-
Inter16x16	34.66	1372.68	-
Inter8x8	34.85	1292.09	-
Inter8x8 Proposed	34.82	1303.24	59.18
Inter Proposed	34.84	1302.43	27.07

表 4-3: High プロファイル使用時の PSNR, 符号量, 演算量の変化

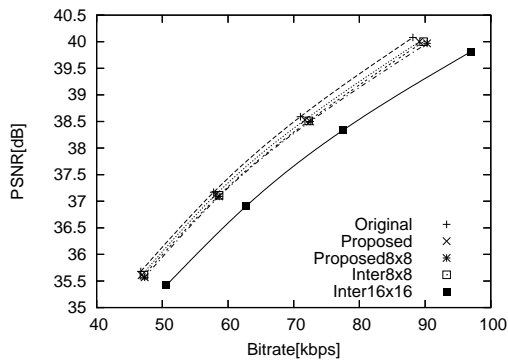
Container Ship			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	37.17	36.19	-
Inter16x16	36.97	38.24	-
Inter8x8	37.15	36.38	-
Inter8x8 Proposed	37.11	36.31	26.25
Inter Proposed	37.1	36.33	8.28
Foreman			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.56	77.09	-
Inter16x16	36.18	87.68	-
Inter8x8	36.48	78.1	-
Inter8x8 Proposed	36.42	79.22	56.33
Inter Proposed	36.49	78.46	32.94
News			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	37.81	64.07	-
Inter16x16	37.53	70.11	-
Inter8x8	37.75	65.46	-
Inter8x8 Proposed	37.69	65.56	31.81
Inter Proposed	37.73	64.66	13.16
Silent			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.82	64.92	-
Inter16x16	36.53	71.22	-
Inter8x8	36.78	65.71	-
Inter8x8 Proposed	36.73	66.45	36.86
Inter Proposed	36.76	65.58	17.01
Mobile&Calendar			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	35.06	2109.75	-
Inter16x16	34.85	2326.24	-
Inter8x8	35.01	2137.07	-
Inter8x8 Proposed	34.96	2174.7	51.43
Inter Proposed	34.99	2157.03	28.32
Paris			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	36.56	828.61	-
Inter16x16	36.32	935.24	-
Inter8x8	36.49	850.27	-
Inter8x8 Proposed	36.45	856.11	38.15
Inter Proposed	36.49	839.23	17.94
Tempete			
	PSNR[dB]	Bitrate[kbps]	Complexity[%]
Original	35.68	1421.44	-
Inter16x16	35.45	1538.83	-
Inter8x8	35.63	1431.57	-
Inter8x8 Proposed	35.59	1456.2	51.61
Inter Proposed	35.61	1449.86	28.1



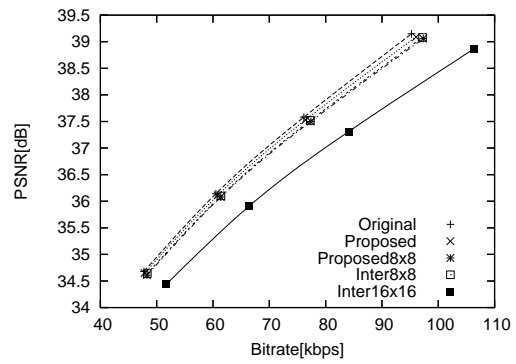
(a) Container Ship



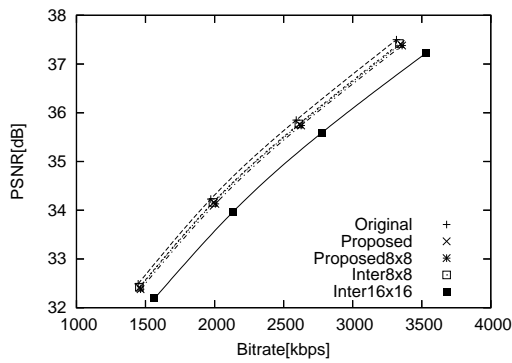
(b) Foreman



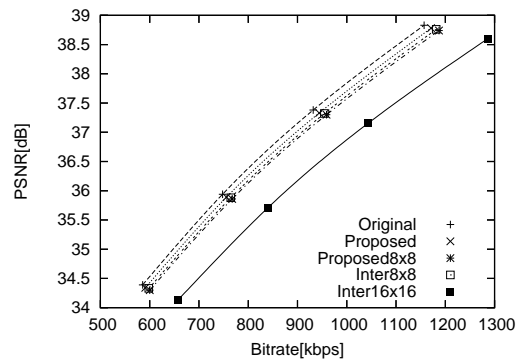
(c) News



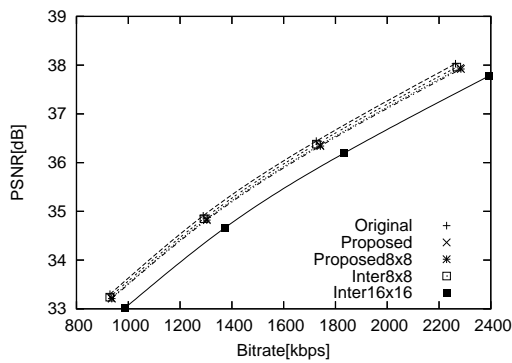
(d) Silent



(e) Mobile&Calendar

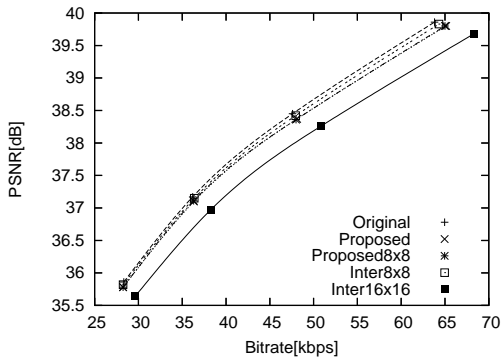


(f) Paris

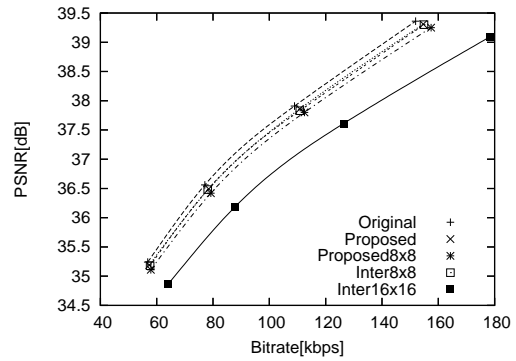


(g) Tempete

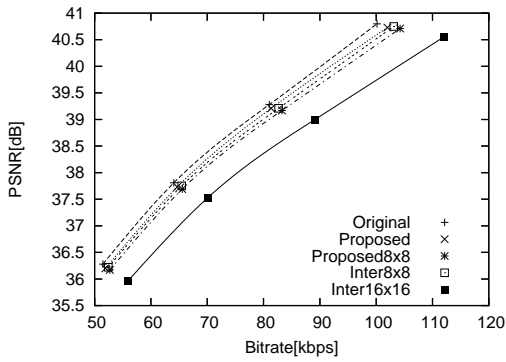
図 4-3: Baseline プロファイル使用時のレート対歪み特性



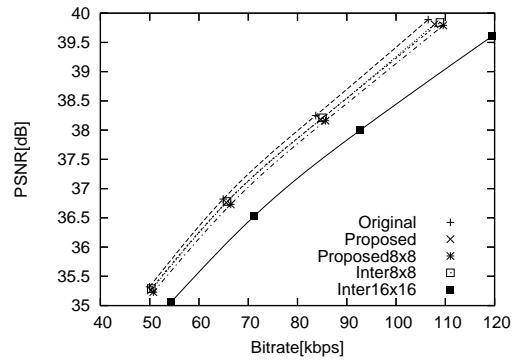
(a) Container Ship



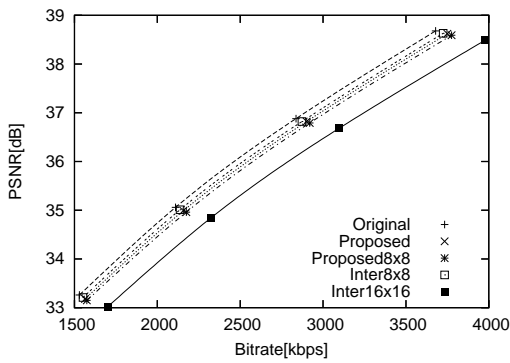
(b) Foreman



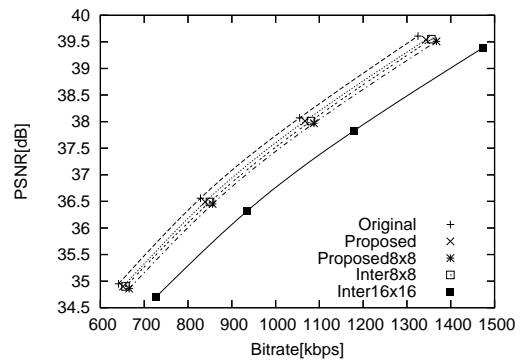
(c) News



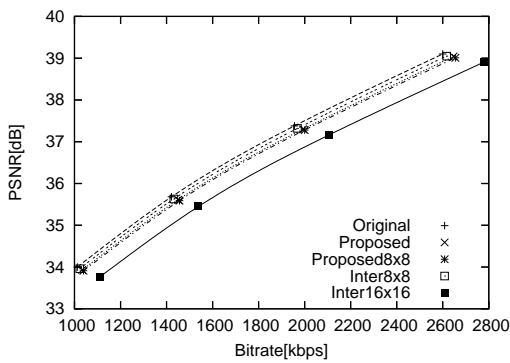
(d) Silent



(e) Mobile&Calendar



(f) Paris



(g) Tempete

図 4-4: High プロファイル使用時のレート対歪み特性

第5章

組み合わせ時の性能評価

5.1 まえがき

本章では第3章で提案した Intra 予測の演算量削減手法と、第4章で提案した Inter 予測の演算量削減手法を組み合わせ用いた際の性能評価を行う。実験を行い、2つの提案手法を同時に利用することが出来ることを示す。また H.264 の符号化演算量削減手法として有名な FastME と提案手法を組み合わせたときの性能評価を行う。符号化演算量の削減度合いと符号量、PSNR の観点から提案手法の評価を行う。また符号化後の画像と符号化モードの利用割合の変化を示し、提案手法により主観的な品質劣化が生じないことを示す。

5.2 全探索時の性能評価

5.2.1 実験条件

H.264 の参照ソフトウェアである JM に Intra 予測と Inter 予測の提案手法を実装し性能の確認を行った [19]。実験条件を表 5-1 に示す。実験はオリジナルの JM(Original)、Intra 予測と Inter 予測のブロックサイズを 16×16 画素単位だけに限定したもの (16×16)、全てのブロックサイズが利用出来る条件で Intra 予測と Inter 予測に対する提案手法を適用したものの (Proposed) の3つに対してそれぞれ行った。

5.2.2 実験結果

QP を 28 に設定した時の Baseline プロファイル使用時の PSNR、符号量、演算量の変化を図 5-2 に、High プロファイル使用時の PSNR、符号量、演算量の変化を図 5-3 にそれぞれ示す。

また QP を変化させて取得した実験結果を元に作成した RD 曲線を示す。図 5-1 に Baseline プロファイル使用時の結果を、図 5-2 に High プロファイル使用時の結果を示す。

原画像と JM オリジナルで符号化した画像、Intra 予測と Inter 予測の処理サイズを 16×16 画素単位だけに制限して符号化した画像、提案手法を用いて符号化した画像の比較を図 5-3 から 5-8 に示す。比較は QCIF サイズの画像 Foreman と CIF サイズの画像 Mobile&Calendar に対して行う。比較対象の画像は 1 フレーム目 (I ピクチャ)、12 フレーム目 (B ピクチャ)、

表 5-1: 実験条件

JM Version	10.1
Profile	Baseline & High Profile
Input Sequence	QCIF Size: Container Ship,Foreman,News,Silent CIF Size: Mobile&Calendar,Paris,Tempete
QP	I , P:22 ~ 28
Number of Frames	150
Frame Structure	N=15, M=1(Baseline Profile) N=15, M=3(High Profile)
Frame Rate	QCIF:15fps CIF:30fps
MC Block Size	16×16 ~ 4×4
Number of Reference Frames	5
Search Range	16
Optimization	RD-Optimization
MC Accuracy	Quarter Pel
Entropy Coding	CAVLC(Baseline Profile) CABAC(High Profile)

13 フレーム目 (P ピクチャ) をそれぞれ選択した .

表 5-2: Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化

Container Ship				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	36.56	65.76	-	-
16x16	36.26	45.17	-	-
Proposed	36.51	38.34	74.91	9.42

Foreman				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	36.03	74.45	-	-
16x16	35.60	93.12	-	-
Proposed	35.98	74.57	51.34	34.93

News				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	37.17	57.82	-	-
16x16	36.72	69.42	-	-
Proposed	37.13	58.17	62.93	14.74

Silent				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	36.14	60.5	-	-
16x16	35.66	72.37	-	-
Proposed	36.11	60.69	36.71	19.67

Mobile&Calendar				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	34.23	1972.85	-	-
16x16	33.92	2187.85	-	-
Proposed	34.15	1992.22	44.40	26.50

Paris				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	35.94	747.96	-	-
16x16	35.57	907.74	-	-
Proposed	35.89	757.61	62.12	20.28

Tempete				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original	34.91	1289.62	-	-
16x16	34.60	1425.20	-	-
Proposed	34.84	1301.77	55.46	27.40

表 5-3: High プロファイル使用時の PSNR, 符号量, 演算量の変化

Container Ship

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	37.17	36.19	-	-	-
16x16	36.74	42.2	-	-	-
Proposed	37.09	36.34	73.2	35.06	8.13

Foreman

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	36.56	77.09	-	-	-
16x16	36.11	97.68	-	-	-
Proposed	36.48	78.61	46.36	39.37	32.83

News

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	37.81	64.07	-	-	-
16x16	37.2	76.64	-	-	-
Proposed	37.71	64.2	59.22	37.71	13.35

Silent

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	36.82	64.92	-	-	-
16x16	36.08	989.37	-	-	-
Proposed	36.74	65.52	34.55	41.29	16.92

Mobile&Calendar

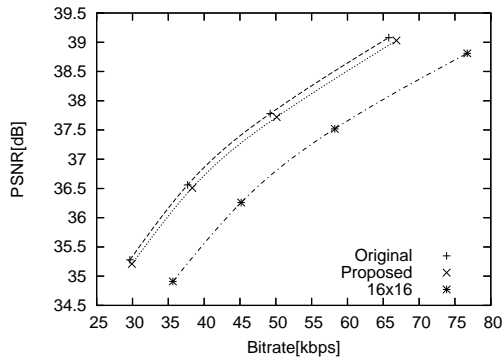
	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	35.06	2109.75	-	-	-
16x16	34.78	2380.29	-	-	-
Proposed	34.99	2155.69	40.18	41.49	28.22

Paris

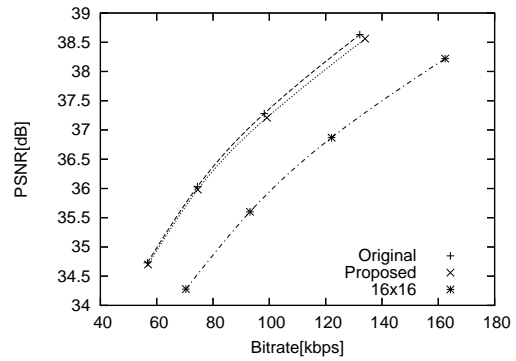
	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	36.56	828.61	-	-	-
16x16	36.08	989.37	-	-	-
Proposed	36.48	838.8	58.13	37.5	17.96

Tempete

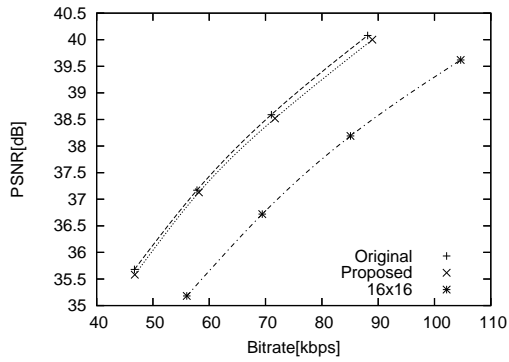
	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original	35.68	1421.44	-	-	-
16x16	35.36	1584.95	-	-	-
Proposed	35.61	1450.48	53.39	39.46	28.19



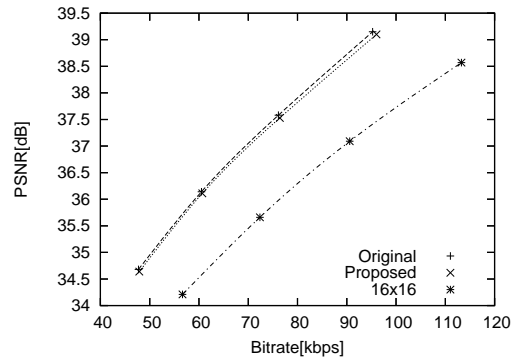
(a) Container Ship



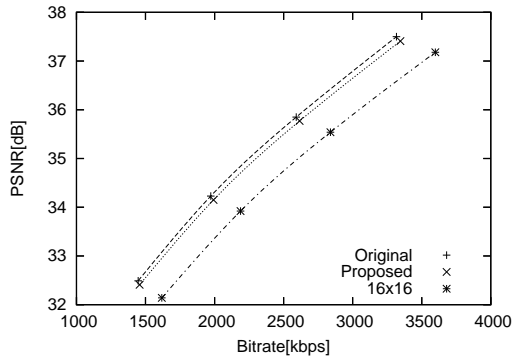
(b) Foreman



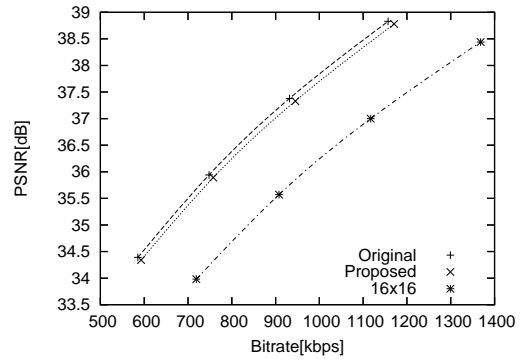
(c) News



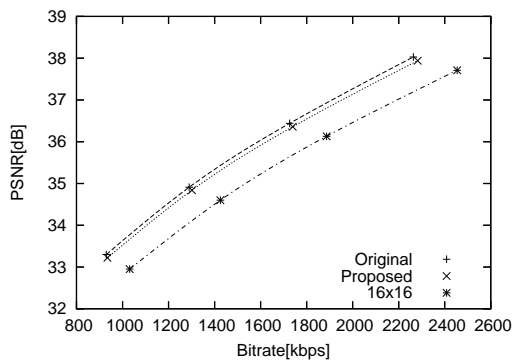
(d) Silent



(e) Mobile&Calendar

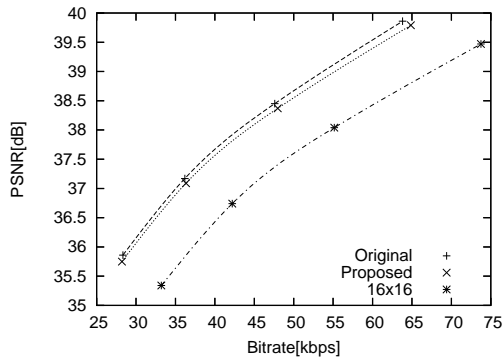


(f) Paris

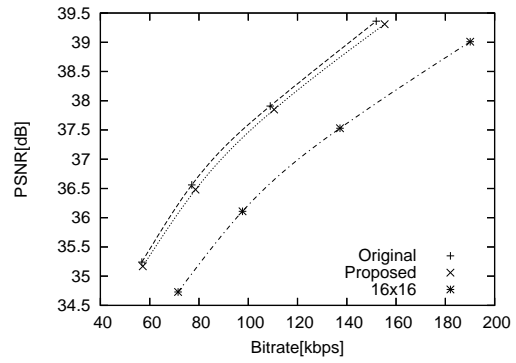


(g) Tempete

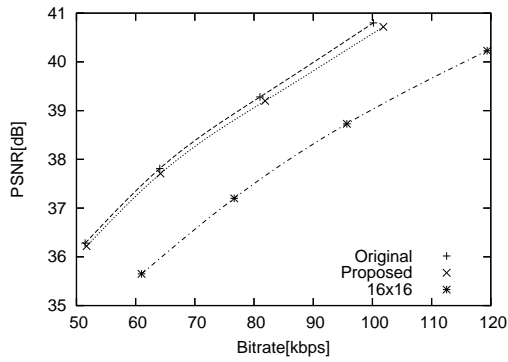
図 5-1: Baseline プロファイル使用時のレート対歪み特性



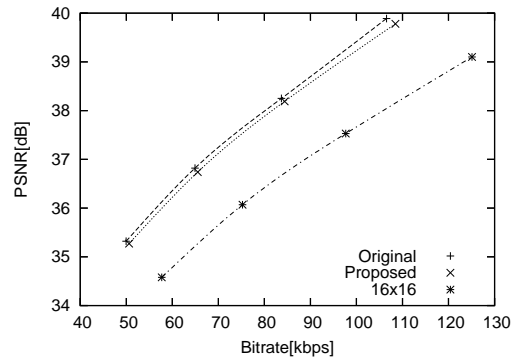
(a) Container Ship



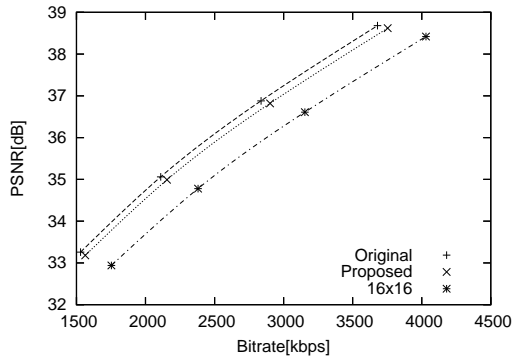
(b) Foreman



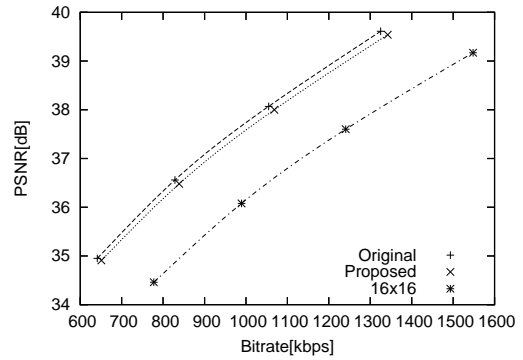
(c) News



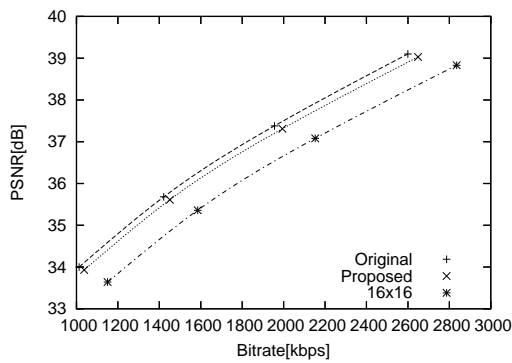
(d) Silent



(e) Mobile&Calendar



(f) Paris



(g) Tempete

図 5-2: High プロファイル使用時のレート対歪み特性



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-3: Foreman Frame No.1(I ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-4: Foreman Frame No.12(B ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-5: Foreman Frame No.13(P ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-6: Mobile&Calendar Frame No.1(I ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-7: Mobile&Calendar Frame No.12(B ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-8: Mobile&Calendar Frame No.13(P ピクチャ)

最後にIピクチャとPピクチャにおける利用モード割合の変化を表5-4から表5-7に示す。

5.2.3 考察

表5-2と表5-3より提案手法を用いることでPSNRの低下と符号量の増加を抑えつつ演算量を削減できていることが分かる。提案手法を用いた場合、JMオリジナルと比較して符号量が小さく抑えられているが、これはInter予測において 16×16 や 8×8 画素単位で処理が打ち切られた際に、大きいブロックサイズが積極的に利用されることが影響していると考えられる。演算量の変化はオリジナルJMの演算量を100%としたときの提案手法における演算量を表している。Intra予測では探索したモード数、Inter予測では探索したブロックサイズ数を値として用いた。演算量はIntra予測で 8×8 画素単位の処理が35%から40%、 16×16 画素単位の処理が35%から70%に削減できた。またInter予測は8%から34%に削減することができた。これは提案手法を別々に適用したときとほぼ同じ割合である。

また図5-3から図5-8に示した符号化した画像の比較と、5-4から5-7に示した利用モード割合の変化からも、提案手法により特定のモードに利用が偏ったり主観的な劣化が生じていないことが見て取れる。利用割合ではPピクチャで 16×16 画素単位の利用割合が増加しているが、これは提案手法における処理の打ち切りに寄るものと考えられる。

表 5-4: Baseline プロファイル使用時の QCIF サイズ画像における利用モード割合の変化

		Container Ship			Foreman		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	643	0	724	952	0	955
	Intra16x16	347	990	266	38	990	35
P Picture	Copy	10953	11142	10911	3298	3443	3318
	16x16	1453	2678	1781	4566	10375	5229
	16x8	567	0	554	1652	0	1446
	8x16	436	0	567	2272	0	2380
	8x8	1216	0	36	5375	0	3878
	8x4	131	0	5	801	0	558
	4x8	326	0	7	1749	0	1152
	4x4	3	0	0	151	0	120
	8x8 以下の総数	419	0	12	2019	0	1427
	Intra4x4	0	0	1	28	0	26
	Intra16x16	32	40	34	25	42	34

		News			Silent		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	859	0	874	955	0	975
	Intra16x16	131	990	116	35	990	15
P Picture	Copy	10165	10479	10247	8799	9117	8794
	16x16	1232	3354	1472	1815	4594	2057
	16x8	576	0	540	772	0	807
	8x16	667	0	906	900	0	1056
	8x8	2995	0	1615	3857	0	2608
	8x4	464	0	233	535	0	360
	4x8	1218	0	666	1251	0	977
	4x4	107	0	102	101	0	87
	8x8 以下の総数	1196	0	654	1436	0	1008
	Intra4x4	23	0	26	117	0	125
	Intra16x16	1	27	15	21	149	13

表 5-5: Baseline プロファイル使用時の CIF サイズ画像における利用モード割合の変化

		Mobile & Calendar			Paris		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	3851	0	3861	3584	0	3680
	Intra16x16	109	3960	99	376	3960	280
P Picture	Copy	3033	3886	3193	34170	35036	34284
	16x16	17163	51486	22892	6981	20078	8541
	16x8	9819	0	9902	2987	0	3606
	8x16	9300	0	13482	3149	0	4339
	8x8	41939	0	15102	19205	0	10504
	8x4	6107	0	2254	3450	0	1823
	4x8	14993	0	5601	8017	0	4576
	4x4	1217	0	647	1056	0	709
	8x8 以下の総数	16064	0	5901	7932	0	4403
	Intra4x4	16	0	16	156	0	171
	Intra16x16	45	68	54	65	326	96

		Tempete		
		Original	16x16	Proposed
I Picture	Intra4x4	3698	0	3728
	Intra16x16	262	3960	232
P Picture	Copy	7137	8463	7417
	16x16	17758	46197	22141
	16x8	9778	0	9491
	8x16	8577	0	11627
	8x8	29797	0	10264
	8x4	4951	0	1479
	4x8	9637	0	3091
	4x4	583	0	274
	8x8 以下の総数	11242	0	3777
	Intra4x4	640	0	673
	Intra16x16	308	780	314

表 5-6: High プロファイル使用時の QCIF サイズ画像における利用モード割合の変化

		Container Ship			Foreman		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	581	0	612	735	0	773
	Intra8x8	154	0	37	242	0	188
	Intra16x16	255	990	341	13	990	29
P Picture	Copy	2167	2449	2242	515	508	507
	16x16	686	1472	808	924	3420	1232
	16x8	329	0	437	498	0	457
	8x16	226	0	401	754	0	778
	8x8	1577	0	120	3210	0	2492
	8x4	153	0	18	316	0	284
	4x8	321	0	30	1217	0	865
	4x4	5	0	0	85	0	63
	8x8 以下の総数	514	0	42	1207	0	926
	Intra4x4	0	0	0	26	0	33
	Intra8x8	9	0	2	28	0	10
	Intra16x16	29	39	28	8	32	17

		News			Silent		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	678	0	731	564	0	677
	Intra8x8	220	0	128	392	0	283
	Intra16x16	92	990	131	34	990	30
P Picture	Copy	2124	2348	2175	1702	1953	1758
	16x16	519	1563	642	732	1933	799
	16x8	255	0	277	289	0	353
	8x16	270	0	360	388	0	459
	8x8	1804	0	1077	2031	0	1281
	8x4	222	0	104	209	0	131
	4x8	794	0	493	583	0	449
	4x4	44	0	46	33	0	27
	8x8 以下の総数	716	0	430	714	0	472
	Intra4x4	63	0	63	50	0	73
	Intra8x8	11	0	10	79	0	42
	Intra16x16	2	49	3	6	74	4

表 5-7: High プロファイル使用時の CIF サイズ画像における利用モード割合の変化

		Mobile & Calendar			Paris		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	3211	0	3512	2893	0	3015
	Intra8x8	674	0	353	843	0	696
	Intra16x16	75	3960	95	224	3960	249
P Picture	Copy	373	408	411	7347	7943	7467
	16x16	3093	15339	5180	2331	7543	2789
	16x8	2029	0	2809	1124	0	1397
	8x16	2136	0	4130	1087	0	1513
	8x8	21158	0	7830	8501	0	5118
	8x4	2242	0	1039	1347	0	937
	4x8	8633	0	3781	4140	0	2956
	4x4	483	0	270	360	0	285
	8x8 以下の総数	8129	0	3230	3587	0	2324
	Intra4x4	39	0	41	236	0	263
	Intra8x8	12	0	4	84	0	31
	Intra16x16	29	93	35	44	354	56

		Tempete		
		Original	16x16	Proposed
I Picture	Intra4x4	2722	0	3243
	Intra8x8	1061	0	498
	Intra16x16	177	3960	219
P Picture	Copy	1008	1109	1023
	16x16	3360	14416	5002
	16x8	2518	0	3185
	8x16	2129	0	3596
	8x8	16489	0	7110
	8x4	2182	0	859
	4x8	6708	0	2502
	4x4	317	0	165
	8x8 以下の総数	6424	0	2659
	Intra4x4	133	0	171
	Intra8x8	164	0	69
	Intra16x16	104	315	135

5.3 FastME 使用時の性能評価

5.3.1 FastME の探索手法

JM で実装されている探索方式は、探索点の中心から螺旋状に探索を行っていくスパイラルサーチの手法をとっている。これは一般的な画像において探索開始点に近い動きベクトルが多く、ある程度探索を行った時点で以降の探索を打ち切ることで探索点を削減出来ることから有効な手法である。しかし JM では打ち切り処理は実装されておらず、実質的に探索範囲を左上から右下まで走査する全探索と同じになってる。

探索範囲を広く取り、かつ探索点を抑えるためにダイヤモンドサーチというアルゴリズムがあるが、そのままでは RD 最適化との相性が悪く、符号化効率が低下する問題点がある。そこで探索点を決定するパターンを複数組み合わせる FastME が提案された [8]。FastME は Unsymmetric-Cross Multi-Hexagon Grid Search というアルゴリズムを用いる。これは目標コストよりも十分に小さい動きベクトルを発見できた場合はダイヤモンドサーチだけが実行され、見つけれなかった場合だけ周囲を広く探していくという処理を行う。

そこで提案手法が飛び飛びに探索点を走査する Unsymmetric-Cross Multi-Hexagon Grid Search と組み合わせても利用出来ることを示す。

5.3.2 実験条件

H.264 の参照ソフトウェアである JM に提案手法を実装し性能の確認を行った [19]。実験条件を表 5-8 に示す。実験はオリジナルの JM で FastME を用いたもの (Original FME)、FastME を用いて Intra 予測と Inter 予測のブロックサイズを 16×16 画素単位だけに限定したもの (16×16 FME)、全てのブロックサイズが利用出来る条件で FastME を用いて Intra 予測と Inter 予測に対する提案手法を適用したもの (Proposed FME) の 3 つに対してそれぞれ行った。

5.3.3 実験結果

QP を 28 に設定した時の Baseline プロファイル使用時の PSNR、符号量、演算量の変化を図 5-9 に、High プロファイル使用時の PSNR、符号量、演算量の変化を図 5-10 にそれぞれ示す。

また QP を変化させて取得した実験結果を元に作成した RD 曲線を示す。図 5-9 に Baseline プロファイル使用時の結果を、図 5-10 に High プロファイル使用時の結果を示す。

原画像と JM オリジナルで符号化した画像、Intra 予測と Inter 予測の処理サイズを 16×16 画素単位だけに制限して符号化した画像、提案手法を用いて符号化した画像の比較を図 5-11 から 5-14 に示す。比較は QCIF サイズの画像 Foreman と CIF サイズの画像 Mobile&Calendar に対して行う。1 フレーム目の I ピクチャは FastME を利用しない条件時と同一なので結果を省略し、比較対象の画像は 12 フレーム目 (B ピクチャ)、13 フレーム目 (P ピクチャ) をそれぞれ選択した。

表 5-8: 実験条件

JM Version	10.1
Profile	Baseline&High Profile
Input Sequence	QCIF Size: Container Ship,Foreman,News,Silent CIF Size: Mobile&Calendar,Paris,Tempete
QP	I , P:22 ~ 28
Number of Frames	150
Frame Structure	N=15, M=1(Baseline Profile) N=15, M=3(High Profile)
Frame Rate	QCIF:15fps CIF:30fps
MC Block Size	16×16 ~ 4×4
Number of Reference Frames	5
Search Range	16
Optimization	RD-Optimization
MC Accuracy	Quarter Pel
Entropy Coding	CAVLC(Baseline Profile) CABAC(High Profile)
Search Mode	FastME(UMHexagon Search)

表 5-9: Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化

Container Ship				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	36.57	37.77	-	-
16x16 FME	36.26	45.18	-	-
Proposed FME	36.55	37.84	74.91	48.41

Foreman				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	36.05	74.03	-	-
16x16 FME	35.61	93.23	-	-
Proposed FME	36.00	74.11	51.15	30.96

News				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	37.19	57.61	-	-
16x16 FME	36.72	69.57	-	-
Proposed FME	37.18	57.99	62.73	48.80

Silent				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	36.15	60.64	-	-
16x16 FME	35.67	72.64	-	-
Proposed FME	36.14	61.24	36.79	42.65

Mobile&Calendar				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	34.26	1972.05	-	-
16x16 FME	33.92	2189.27	-	-
Proposed FME	34.21	1980.81	44.22	20.49

Paris				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	35.96	746.19	-	-
16x16 FME	35.57	910.33	-	-
Proposed FME	35.94	756.33	62.14	43.61

Tempete				
	PSNR[dB]	Bitrate[kbps]	Complexity[%]	
			Intra	Inter
Original FME	34.94	1286.36	-	-
16x16 FME	34.60	1426.73	-	-
Proposed FME	34.91	1289.76	55.55	24.81

表 5-10: High プロファイル使用時の PSNR, 符号量, 演算量の変化

Container Ship

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	37.17	36.13	-	-	-
16x16 FME	36.74	42.24	-	-	-
Proposed FME	37.14	35.99	73.24	34.92	38.32

Foreman

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	36.56	76.62	-	-	-
16x16 FME	36.1	97.8	-	-	-
Proposed FME	36.52	77.69	46.69	39.37	29.35

News

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	37.85	64.03	-	-	-
16x16 FME	37.19	76.02	-	-	-
Proposed FME	39.61	64.64	59.32	37.72	40.16

Silent

	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	36.83	64.87	-	-	-
16x16 FME	36.06	75.38	-	-	-
Proposed FME	36.79	65.91	34.59	41.23	30.12

Mobile&Calendar

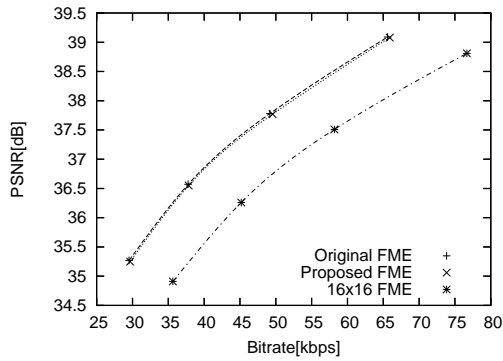
	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	35.08	2098.91	-	-	-
16x16 FME	34.78	2378.57	-	-	-
Proposed FME	35.04	2122.81	40.13	41.5	22.38

Paris

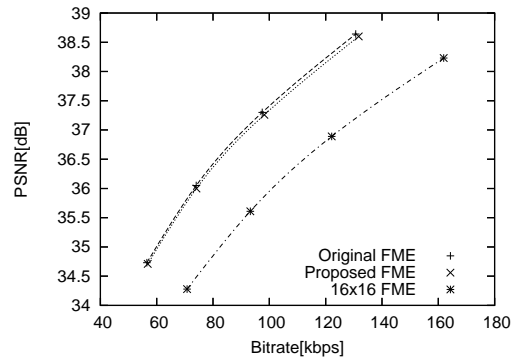
	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	36.57	825.58	-	-	-
16x16 FME	36.08	991.92	-	-	-
Proposed FME	36.54	838.61	57.95	37.57	31.37

Tempete

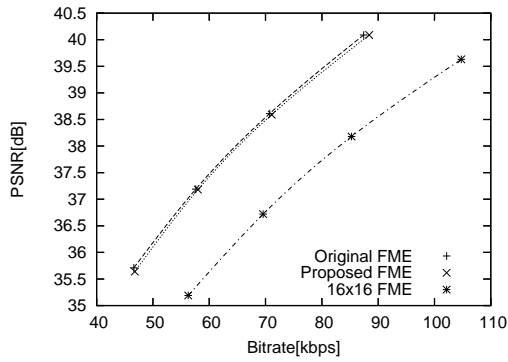
	PSNR[dB]	Bitrate[kbps]	Complexity[%]		
			Intra16x16	Intra8x8	Inter
Original FME	35.7	1412.04	-	-	-
16x16 FME	35.36	1588.58	-	-	-
Proposed FME	35.67	1430.07	53.51	39.44	26.42



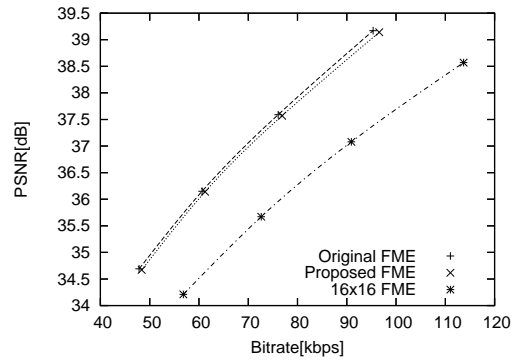
(a) Container Ship



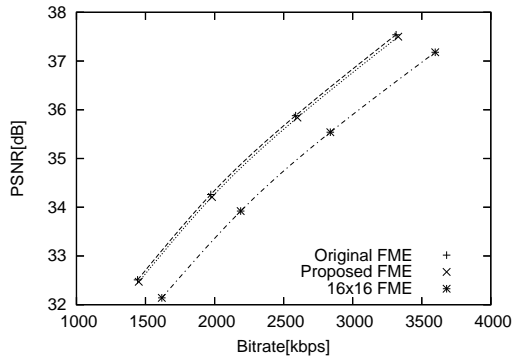
(b) Foreman



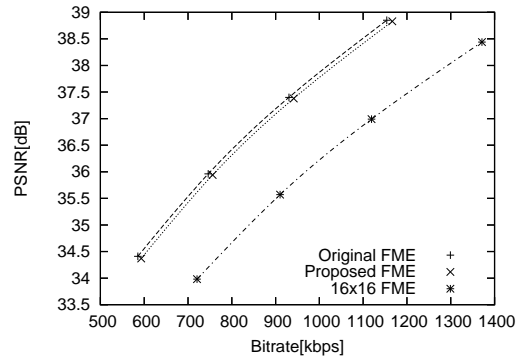
(c) News



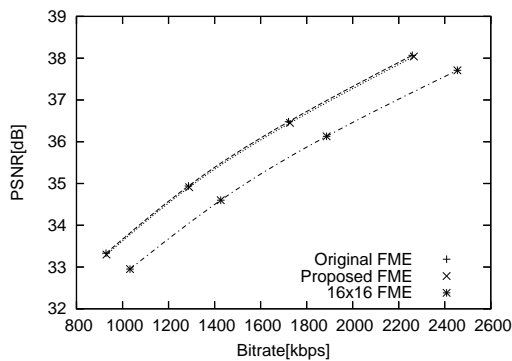
(d) Silent



(e) Mobile&Calendar

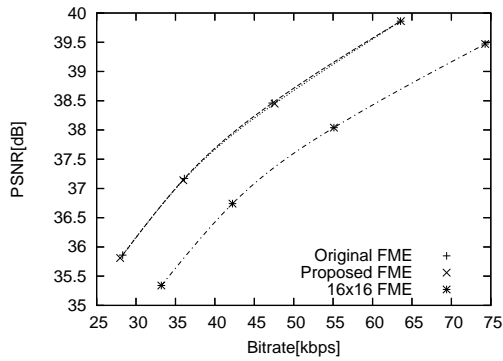


(f) Paris

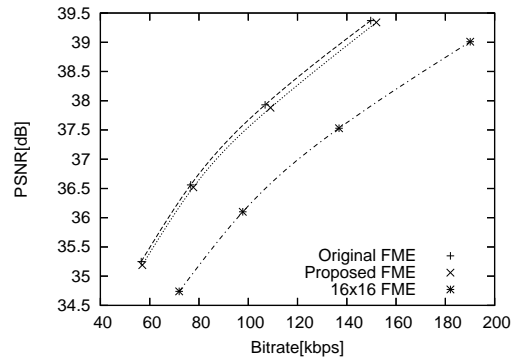


(g) Tempete

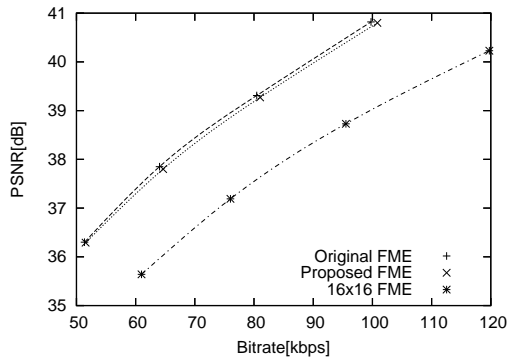
図 5-9: Baseline プロファイル使用時のレート対歪み特性



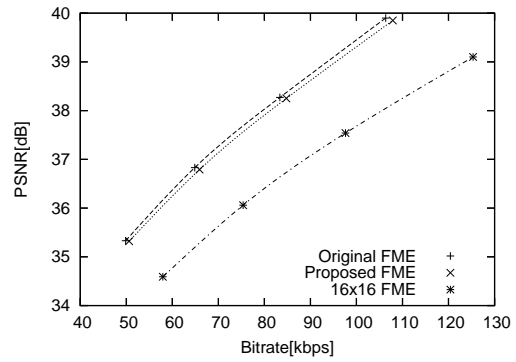
(a) Container Ship



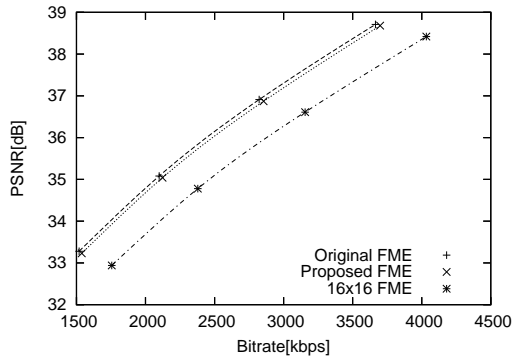
(b) Foreman



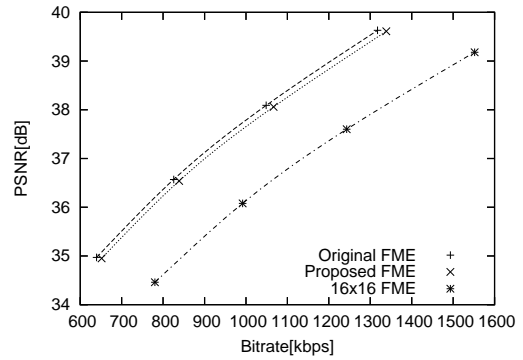
(c) News



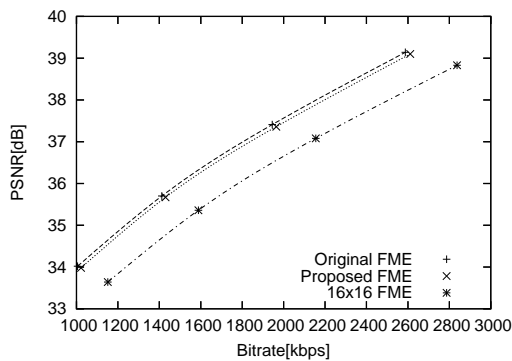
(d) Silent



(e) Mobile&Calendar



(f) Paris



(g) Tempete

図 5-10: High プロファイル使用時のレート対歪み特性



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-11: Foreman Frame No.12(B ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-12: Foreman Frame No.13(P ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-13: Mobile&Calendar Frame No.12(B ピクチャ)



(a) Original(Not Encoded)



(b) JM Original



(c) 16x16



(d) Proposed

図 5-14: Mobile&Calendar Frame No.13(P ピクチャ)

最後にIピクチャとPピクチャにおける利用モード割合の変化を表5-11から5-14に示す。

5.3.4 考察

表5-9と表5-10より提案手法を用いることでPSNRの低下と符号量の増加を抑えつつ演算量を削減できていることが分かる。提案手法ではJMオリジナルと比較して符号量が小さく抑えられているが、これはInter予測において 16×16 や 8×8 画素単位で処理が打ち切られた際に大きいブロックサイズが積極的に利用されることが影響していると考えられる。演算量の変化はオリジナルJMの演算量を100%としたときの提案手法における演算量を表している。Intra予測では探索したモードの数、Inter予測では探索したブロックサイズ数を値として用いた。演算量はIntra予測で 8×8 画素単位の処理が35%から40%、 16×16 画素単位の処理が35%から70%に削減できた。またInter予測は20%から40%に削減することができた。スパイラルサーチに適用した場合と比較して削減度合いは10%ほど低くなっているが、演算量の削減度合いはそれでも大きいと言える。

また図5-11から図5-14に示した符号化した画像の比較と、5-11から5-14に示した利用モード割合の変化からも、提案手法により特定のモードに利用が偏ったり主観的な劣化が生じていないことが見て取れる。利用割合ではPピクチャで 16×16 画素単位の利用割合が増加しているが、これは提案手法における処理の打ち切りに寄るものと考えられる。

5.4 むすび

本章では第3章と第4章で提案したIntra予測の演算量削減手法とInter予測の演算量削減手法を組み合わせ用いた際の性能評価を行った。評価実験より2つの提案手法は同時に利用することを確認した。また符号化演算量削減手法のFastMEとも提案手法が同時に利用出来ることを示した。性能評価を符号化演算量の削減度合いと符号量、PSNRの観点から行った。また利用モード割合の変化と符号化した画像を示し、提案手法により利用モードの割合が極端に変化することなく、主観的な劣化を生じない事を確認した。

表 5-11: Baseline プロファイル使用時の QCIF サイズ画像における利用モード割合の変化

		Container Ship			Foreman		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	643	0	724	952	0	955
	Intra16x16	347	990	266	38	990	35
P Picture	Copy	10911	11151	10912	3341	3554	3387
	16x16	1464	2668	1475	4442	10257	4465
	16x8	566	0	561	1532	0	1602
	8x16	452	0	441	2267	0	2462
	8x8	1197	0	1229	5491	0	4800
	8x4	169	0	157	1310	0	1091
	4x8	307	0	286	1640	0	1317
	4x4	67	0	68	423	0	328
	8x8 以下の総数	435	0	435	2216	0	1884
	Intra4x4	0	0	0	33	0	29
	Intra16x16	32	41	36	29	49	31

		News			Silent		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	859	0	874	955	0	975
	Intra16x16	131	990	116	35	990	15
P Picture	Copy	10181	10482	10154	8794	9114	8834
	16x16	1098	3346	1187	1811	4583	1859
	16x8	544	0	600	721	0	796
	8x16	616	0	666	890	0	1011
	8x8	3159	0	2731	3742	0	3029
	8x4	862	0	709	932	0	712
	4x8	1212	0	1049	1200	0	969
	4x4	315	0	315	230	0	198
	8x8 以下の総数	1387	0	1201	1526	0	1227
	Intra4x4	28	0	37	106	0	116
	Intra16x16	6	32	15	12	163	17

表 5-12: Baseline プロファイル使用時の CIF サイズ画像における利用モード割合の変化

		Mobile & Calendar			Paris		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	3851	0	3861	3584	0	3680
	Intra16x16	109	3960	99	376	3960	280
P Picture	Copy	2947	3901	3057	34250	35041	34190
	16x16	16175	51478	17167	6697	20024	6940
	16x8	8964	0	10396	2796	0	3261
	8x16	8769	0	10079	2930	0	3498
	8x8	42561	0	33534	18390	0	15929
	8x4	12314	0	10037	5636	0	4610
	4x8	14164	0	11077	7250	0	6045
	4x4	5057	0	4044	2864	0	2492
	8x8 以下の総数	18524	0	14673	8535	0	7269
	Intra4x4	23	0	16	156	0	188
	Intra16x16	38	61	52	76	375	94

		Tempete		
		Original	16x16	Proposed
I Picture	Intra4x4	3698	0	3728
	Intra16x16	262	3960	232
P Picture	Copy	7156	8643	7087
	16x16	17013	45996	17267
	16x8	8758	0	9605
	8x16	8039	0	8806
	8x8	30972	0	27383
	8x4	10297	0	8925
	4x8	9261	0	7698
	4x4	3470	0	2710
	8x8 以下の総数	13500	0	11679
	Intra4x4	675	0	665
	Intra16x16	299	801	331

表 5-13: High プロファイル使用時の QCIF サイズ画像における利用モード割合の変化

		Container Ship			Foreman		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	581	0	612	735	0	773
	Intra8x8	154	0	37	242	0	188
	Intra16x16	255	990	341	13	990	29
P Picture	Copy	2235	2449	2266	487	466	493
	16x16	608	1474	611	805	3455	929
	16x8	320	0	337	479	0	579
	8x16	221	0	241	824	0	889
	8x8	1547	0	1367	3022	0	2509
	8x4	197	0	186	846	0	566
	4x8	271	0	263	1037	0	758
	4x4	109	0	104	311	0	223
	8x8 以下の総数	531	0	480	1304	0	1014
	Intra4x4	3	0	0	33	0	33
	Intra8x8	10	0	5	21	0	12
	Intra16x16	32	37	20	7	39	11

		News			Silent		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	678	0	731	564	0	677
	Intra8x8	220	0	128	392	0	283
	Intra16x16	92	990	131	34	990	30
P Picture	Copy	2078	2350	2061	1707	1952	1714
	16x16	521	1564	550	697	1924	760
	16x8	241	0	257	282	0	339
	8x16	264	0	317	355	0	381
	8x8	1687	0	1585	1968	0	1671
	8x4	453	0	405	406	0	320
	4x8	736	0	610	573	0	470
	4x4	232	0	192	137	0	95
	8x8 以下の総数	777	0	698	771	0	639
	Intra4x4	64	0	66	56	0	68
	Intra8x8	12	0	7	88	0	55
	Intra16x16	3	46	4	4	84	4

表 5-14: High プロファイル使用時の CIF サイズ画像における利用モード割合の変化

		Mobile & Calendar			Paris		
		Original	16x16	Proposed	Original	16x16	Proposed
I Picture	Intra4x4	3211	0	3512	2893	0	3015
	Intra8x8	674	0	353	843	0	696
	Intra16x16	75	3960	95	224	3960	249
P Picture	Copy	375	402	382	7341	7982	7302
	16x16	2664	15322	2994	2162	7447	2291
	16x8	1699	0	2455	1055	0	1262
	8x16	1773	0	2488	1041	0	1402
	8x8	19893	0	16391	7944	0	6682
	8x4	6014	0	4760	2495	0	2095
	4x8	7328	0	5442	3443	0	2574
	4x4	3733	0	3083	1598	0	1425
	8x8 以下の総数	9242	0	7419	3870	0	3194
	Intra4x4	40	0	63	247	0	290
	Intra8x8	11	0	1	76	0	47
	Intra16x16	36	116	38	48	411	52

		Tempete		
		Original	16x16	Proposed
I Picture	Intra4x4	2722	0	3243
	Intra8x8	1061	0	498
	Intra16x16	177	3960	219
P Picture	Copy	1053	1172	1004
	16x16	2910	14337	3196
	16x8	2209	0	2778
	8x16	1901	0	2530
	8x8	14891	0	12505
	8x4	5871	0	4738
	4x8	5406	0	4068
	4x4	3320	0	2509
	8x8 以下の総数	7372	0	5955
	Intra4x4	132	0	161
	Intra8x8	166	0	75
	Intra16x16	97	331	141

第6章

結論

6.1 総括

本研究は動画像の符号化処理において効率的な符号化処理を行い、演算量を削減する手法について検討を行った。2003年にMPEGとITU-Tの合同標準化部会であるJVT(Joint Video Team)で標準化されたH.264では符号化効率を高めるためにこれまでの標準化では採用が見送られてきた多くの演算量を必要とする符号化機能が多数搭載され、演算量が増加した。

H.264での急激な符号化演算量の増加によりムーアの法則によるハードウェアの性能向上を期待してはいつまでも利用することができず、不要な符号化処理を削減し、画像品質を落とすことなく効率的に処理する手法が必要とされている。そこで特に多くの符号化演算量を占めるIntra予測とInter予測に着目し、両処理の演算量削減手法を提案した。提案手法は符号化効率に大きく貢献しているブロックサイズの処理単位に着目し、これを利用して不要な符号化処理を省略することで演算量の削減を行った。

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章では動画像符号化技術について述べた。画像符号化技術が登場した背景と圧縮技術の必要性を述べ、符号化技術の標準化の歴史と各標準符号化方式で利用されている符号化機能を紹介した。本論文で扱うH.264を取り上げ、新たに追加された符号化機能や利用用途を中心に紹介した。最後にH.264において符号化機能が符号化効率にどの程度寄与していて、どの程度の演算量を占めているか基礎検討を行った。

第3章ではIntra予測の演算量削減手法について述べた。これまでに提案されているIntra予測の演算量削減手法を述べ、本章で提案する複数の処理サイズの違いを活用したIntra予測の符号化演算量削減手法を述べた。提案手法では 4×4 画素単位の予測モード結果を利用して 8×8 と 16×16 画素単位の予測処理を制限する。提案手法の性能評価を符号量、PSNRの観点から評価を行い、提案手法を用いることで符号化効率を大きく低下させることなく、 8×8 画素単位の演算量を約40%、 16×16 画素単位の演算量を35%から70%へ削減出来ることを示した。

第4章ではInter予測の演算量削減手法について述べた。これまでに提案されているInter予測の演算量削減手法を述べ、本章で提案する複数の処理サイズの違いを活用したInter予

測の符号化演算量削減手法を述べた．提案手法では符号化対象のマクロブロックと探索中に参照するマクロブロックで画素値の差分をとり，用いることで探索する動き補償ブロックサイズを効率よく削減する．提案手法の性能評価を符号量，PSNR の観点から評価を行い，提案手法を用いることで符号化効率を大きく低下させることなく，Inter 予測で探索するブロックサイズの種類を全てのブロックサイズを利用出来る条件下で8%から32%に， 8×8 画素単位までに制限した条件下で40%から60%に削減することを示した．

第5章では第3章と第4章で提案した Intra 予測と Inter 予測の演算量削減手法を組み合わせ用いた際の性能評価を行った．2つの提案手法は同時に利用することが出来る．また符号化演算量削減手法の FastME とも提案手法が同時に利用出来ることを示した．性能評価を符号化演算量の削減度合いと符号量，PSNR の観点から行った．また利用モード割合の変化と符号化した画像を示し，提案手法により利用モードの割合が極端に変化することなく，主観的にも問題が無い事を示した．

第6章は結論であり，本論文の総括および今後の課題について述べている．

6.2 今後の課題

今後検討すべき課題を以下に書く．

6.2.1 最適な符号化処理単位の検討

文献 [36] ではマクロブロックサイズの最適なサイズについて検討を行っている．符号化処理に最適な符号化処理単位は画像サイズや画角などによって決定する要素であると考えられる．H.264 では最大サイズが 16×16 画素単位，最小サイズが 4×4 画素単位となっているが，さらに最適な符号化処理単位が存在することが考えられる．符号化処理単位が変化すれば本論文で提案した手法はさらに有効に演算量削減に対して貢献すると考えられる．

6.2.2 動き補償の改善

H.264 で導入された可変ブロックサイズ動き補償は符号化効率の向上に大きく貢献しているが，動物体のエッジ方向に沿う分割が出来ない場合もあり，改善の余地があると言える．利用するブロックサイズの形状や数を変更することで符号化効率が向上することが報告されているが符号化演算量の問題から手法をそのまま適用出来ない問題がある．

符号量は動きベクトルなどが記述されるヘッダ部と予測後の画素値を DCT(整数変換)して得られる DCT 係数の和により与えられる．よって符号量の削減は分割ブロック数の制限による動きベクトル数の削減や，効率的な動き補償による DCT 係数の削減によって実現出来る．マクロブロックの分割を禁止し，動きベクトルの数を1つに制限すればヘッダは小さくなるが発生する DCT 係数値が増加する．またブロックサイズ分割単位を細かくし，分割ブロック数を増加させると DCT 係数を抑えることが出来るが，動きベクトル数が増加しヘッダの符号量が増加することによって符号量は全体で見ると増加することも考えられる．

また任意分割サイズを実現するには直交変換との相性も考慮しなければならない。H.264では最小分割単位を整数変換の処理単位と同じ 4×4 画素単位に制限している。よって全ての処理単位は単体で最適な Intra 予測モードや動きベクトルを直交変換後の符号量まで考慮して決定することが出来る。しかし最小分割単位を1画素とするとRD最適化において符号量の予測が困難になり、符号化演算量の増加につながると考えられる。

文献 [37] で提案されている手法では分割ブロックサイズを2つまでに制限することで動きベクトルの数を制限し、分割形状を工夫することでH.264と同等の符号化性能を実現する手法が提案されている。また文献 [38] により提案されている1画素単位の任意分割を用いる手法では1画素単位の分割を可能にし、H.264に任意の線分による分割パターンを追加したところ符号化効率が4%から5%改善したことが報告されている。

H.264ではヘッダの占める割合が増加しており、予測処理を細かく行ってもヘッダの符号量が増加してしまう問題点が指摘されている。複数の動きベクトルをまとめて扱うなど動き補償処理の改善が必要であると考えられる。

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。

研究の方向性について有意義な御意見を頂き、研究活動を支援していただい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、御提案を頂いた渡辺研究室の皆様に御礼申し上げます。また渡辺研究室、富永研究室の良好で活発な関係において2年間行ってきたH.264ゼミでは貴重な議論の場を与えて頂きました。中でも折に触れて研究に対する助言を頂いた花村 剛氏、永吉 功氏に深く感謝いたします。

配属以来、研究の進め方から文章の書き方などを懇切丁寧に指導して頂いてきた多くの先輩方、互いがよい目標となり切磋琢磨し勉学に励んできた同期の友人、様々な活動を通して貴重な経験をさせて頂いた渡辺、富永両研究室の皆様的心遣いに深く感謝致します。

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1月31日

常松 祐一

参考文献

- (1) Digital Cinema Initiatives, LLC, "Digital Cinema System Specification v4.3," December 2004.
- (2) 竹間 忠夫, "夢のスーパーハイビジョンに挑む," 日本放送出版協会, 2005.
- (3) ISO/IEC 14496-10, "Advanced Video Coding," (ITU-T Rec. H.264) 2003.
- (4) 日経エレクトロニクス, "マンモス・コーデックと呼ばれて," 日経 BP 社, January, 2005.
- (5) 映像情報メディア学会, "統合マルチメディア選書 MPEG," オーム社, 1996.
- (6) 木本, "MPEG におけるスケーラブルビデオ符号化の標準動向," 情報処理学会 AVM 研究会, March, 2005.
- (7) J.Reichel et al, "Scalable Video Coding - Working Draft 4," JVT-Q201, October, 2005.
- (8) Z.Chen, P.Zhou, Y.He, "Fast Motion Estimation for JVT," JVT-G016, 7th Meeting, Pattaya, Thailand, March, 2003.
- (9) ISO/IEC JTC1/SC29WG1, "ISO/IEC 15444-3 JPEG Part3: 2000 Motion JPEG2000," September, 2001.
- (10) 角野, 菊池, 鈴木, 大久保, "H.264/AVC 教科書," インプレス, 2004.
- (11) 矢ヶ崎, 鈴木, 佐藤, 村山, "次世代動画像符号化方式 MPEG4 AVC/H.264 ," トリケッ
プス, 2004.
- (12) 半谷, 杉山, "JPEG・MPEG 完全理解," コロナ社, 2005.
- (13) S.Srinivasan et al, "ペールを脱ぐ WMV9 H.264 とはここが違う," 日経エレクトロニ
クス, March, 2004.
- (14) "アップル - Mac OS X - iChat AV," [http://www.apple.com/jp/macosex/features/
ichat/](http://www.apple.com/jp/macosex/features/ichat/), April, 2005.
- (15) "アップル - QuickTime," <http://www.apple.com/jp/quicktime/>, April, 2005.
- (16) "アップル - iPod," <http://www.apple.com/jp/ipod/>, October, 2005.
- (17) "[D-pa] 社団法人 地上デジタル放送推進協会," <http://www.d-pa.org/>
- (18) "Data Interchange on 60 mm Read-Only ODC - Capacity: 1,8 Gbytes (UMD),"
Standard ECMA-365, June, 2005.
- (19) "JVT Reference Software," <http://iphome.hhi.de/suehring/tml/>, November, 2005.

- (20) C. Kim, Q.Li, C.-C.J. Kuo, "Fast Intra-Prediction Model Selection for H.264 Codec," Proceeding of of the SPIE, Volume 5241, pp.99-110, 2003.
- (21) 高木, 滝嶋, 中島, "H.264—AVC におけるイントラ予測高速化の一検討," 情報処理学会 AVM 研究会, December, 2003.
- (22) 常松, 渡辺, "H.264/AVC における Intra 予測モードの決定手法に関する検討," 電子情報通信学会論文誌 D-II, Vol.J.88-D-II, No.9, September, 2005.
- (23) J.Y.Tham, S.Ranganath, M.Ranganath, and A.A.Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 369-377, Aug. 1998.
- (24) K.K.Ma, G.Qiu, "An Improved Adaptive Rood Pattern Search For Fast Block-matching Motion Estimation in JVT/H.26L," Proceeding of IEEE International Symposium on Circuits and Systems, Bangkok, Thailand, May, 2003.
- (25) D.Zhu, Q.Dai, D.Ding, "Fast Inter Prediction Mode Decision for H.264," Proceeding of of IEEE International Conference on Image Processing, October, 2004.
- (26) X.Li, E.Q.Li, Y.Chen, "Fast Multi-Frmae Motion Estimation Algorithm With Adaptive Search Strategies in H.264," Proceeding of of IEEE International Conference on Acoustics, Speech, & Signal Processing, May, 2004.
- (27) Y.Huang, B.Hsieh, T.Wang, S.Chien, S.Ma, C.Shen, L.Chen, "Analysys And Reduction Of Reference Frame For Motion Estimation In MPEG-4 AVC/JVT/H.264," Proceeding of of IEEE International Conference on Acoustics, Speech & Signal Processing, May, 2004.
- (28) M.Chen, Y.Chiang, H.Li, M.Chi, "Efficient Multi-Frame Motion Estimation Algorithm For MPEG-4 AVC/JVT/H.264," Proceeding of of IEEE International Symposium on Circuits and Systems, May, 2004.
- (29) Zhi Zhou, Ming-Ting Sun, Yuh-Feng Hsu, "Fast Variable Block-size Motion Estimation Algorithms Based On Merge And Split Procedures For H.264/mpeg-4 Avc," Proceeding of of IEEE International Symposium on Circuits and Systems, May, 2004.
- (30) Y.K.Tu, J.F.Yang, Y.N.Shen, M.T.Sun, "Fast Variable-Size Block Motion Estimation Using Merging Procedure With an Adaptive Threshold," Proceeding of IEEE International Conference on Multimedia & Expo, Baltimore, MD, USA, July, 2003.
- (31) 清水, 米山, 柳原, 中島, "H.264 符号化処理における動き予測の高速化に関する一検討," 情報処理学会 AVM 研究会, October, 2003.

- (32) C.Cho, S.Huang, J.Wang, "An Embedded Merging Scheme For H.264/AVC Motion Estimation," Proceeding of IEEE International Conference on Image Processing, September, 2003.
- (33) X.Li, G.Wu, "Fast Integer Pixel Motion Estimation," JVT-F011, 6th Meeting, Awaji Island, December, 2002.
- (34) A.Chang, P.H.W.Wong, Y.M.Yeung, O.C.Au, "Fast Multi-Block Selection For H.264 Video Coding," Proceeding of of IEEE International Symposium on Circuits and Systems, May, 2004.
- (35) D.Wu, S.Wu, K.P.Lim, F.Pan, Z.G.Li, X.Lin, "Block Inter Mode Decision For Fast Encoding Of H.264," Proceeding of of IEEE International Conference on Acoustics, Speech, and Signal Processing, May, 2004.
- (36) 松村, 内藤整, 川田, 小池, "超高詳細映像を対象とした高圧縮符号化における最大マクロブロックサイズの決定方式に関する一検討," 情報処理学会 AVM 研究会, October, 2005.
- (37) S.SEKIGUCHI, S.KATO, S.ADACHI, M.ETOH, "Low-overhead INTER Prediction Modes," VCEG 14th meeting, Santa Barbara, CA, VCEG-N45, September, 2001.
- (38) 近藤, 笹井, "線分分割ブロック動き補償を用いた動画像符号化," PCSJ, November, 2004.

図一覧

2-1	動画像圧縮技術の基本処理構成	6
2-2	動き補償の原理	7
2-3	プロファイルの関係	12
2-4	H.264 エンコーダの構成	13
2-5	Intra 予測の予測方向とモード番号	14
2-6	H.264 で Inter 予測に追加された機能	15
2-7	H.264 符号化処理時間の割合	20
2-8	Intra 予測のモード利用割合	21
2-9	Inter 予測のブロックサイズ利用割合	21
2-10	各機能の符号化効率への寄与 (Baseline プロファイル)	23
2-11	各機能の符号化効率への寄与 (High プロファイル)	24
2-12	Baseline プロファイルと High プロファイルの符号化効率比較	25
3-1	Intra 予測の提案手法	27
3-2	Baseline プロファイル使用時のレート対歪み特性	33
3-3	High プロファイル使用時のレート対歪み特性	34
4-1	Inter 予測の提案手法	37
4-2	Mobile&Calendar におけるブロックサイズ分割形状	38
4-3	Baseline プロファイル使用時のレート対歪み特性	43
4-4	High プロファイル使用時のレート対歪み特性	44
5-1	Baseline プロファイル使用時のレート対歪み特性	49
5-2	High プロファイル使用時のレート対歪み特性	50
5-3	Foreman Frame No.1(I ピクチャ)	51
5-4	Foreman Frame No.12(B ピクチャ)	52
5-5	Foreman Frame No.13(P ピクチャ)	53
5-6	Mobile&Calendar Frame No.1(I ピクチャ)	54
5-7	Mobile&Calendar Frame No.12(B ピクチャ)	55
5-8	Mobile&Calendar Frame No.13(P ピクチャ)	56
5-9	Baseline プロファイル使用時のレート対歪み特性	66
5-10	High プロファイル使用時のレート対歪み特性	67
5-11	Foreman Frame No.12(B ピクチャ)	68

5-12 Foreman Frame No.13(P ピクチャ)	69
5-13 Mobile&Calendar Frame No.12(B ピクチャ)	70
5-14 Mobile&Calendar Frame No.13(P ピクチャ)	71

表一覧

2-1	国際標準の動画像符号化方式	8
2-2	H.264 で規定されているプロファイル	18
2-3	実験条件	19
2-4	符号化機能を制限したときの処理時間	22
3-1	8×8 探索モード決定に用いるコスト値	28
3-2	実験条件	29
3-3	Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化	31
3-4	High プロファイル使用時の PSNR, 符号量, 演算量の変化	32
4-1	実験条件	39
4-2	Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化	41
4-3	High プロファイル使用時の PSNR, 符号量, 演算量の変化	42
5-1	実験条件	46
5-2	Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化	47
5-3	High プロファイル使用時の PSNR, 符号量, 演算量の変化	48
5-4	Baseline プロファイル使用時の QCIF サイズ画像における利用モード割合の 変化	58
5-5	Baseline プロファイル使用時の CIF サイズ画像における利用モード割合の変化	59
5-6	High プロファイル使用時の QCIF サイズ画像における利用モード割合の変化	60
5-7	High プロファイル使用時の CIF サイズ画像における利用モード割合の変化	61
5-8	実験条件	63
5-9	Baseline プロファイル使用時の PSNR, 符号量, 演算量の変化	64
5-10	High プロファイル使用時の PSNR, 符号量, 演算量の変化	65
5-11	Baseline プロファイル使用時の QCIF サイズ画像における利用モード割合の 変化	73
5-12	Baseline プロファイル使用時の CIF サイズ画像における利用モード割合の変化	74
5-13	High プロファイル使用時の QCIF サイズ画像における利用モード割合の変化	75
5-14	High プロファイル使用時の CIF サイズ画像における利用モード割合の変化	76

研究業績

	題 名	発表年月	発表掲載誌	連名者
国内				
(1)	コンテンツ制作のための ID 付加方式の提案	2004 年 3 月	信学春季全大, D-11-111	常松 祐一 高木 真一 富永 英義
(2)	H.264/AVC における Intra 予測モードの決定手法に関する一検討	2004 年 11 月	PCSJ2004, P-5.16	常松 祐一 渡辺 裕
(3)	H.264/AVC FRExt における Intra 予測モードの決定手法に関する一検討	2004 年 12 月	情報処理学会 AVM 研究会, Vol.2004 No.126	常松 祐一 渡辺 裕
(4)	モードの偏りを用いた H.264/AVC FRExt Intra モード決定手法	2005 年 3 月	信学春季全大, D-11-55	常松 祐一 渡辺 裕
(5)	フィルムグレインを考慮した H.264/AVC モード決定手法に関する一検討	2005 年 3 月	信学春季全大, D-11-56,	後藤 崇行 常松 祐一 渡辺 裕
(6)	H.264 における動き補償ブロックサイズ決定手法に関する一検討	2005 年 7 月	情報処理学会 AVM 研究会, Vol.2005 No.66, pp19-24	常松 祐一 後藤 崇行 渡辺 裕
(7)	フィルムグレインを考慮した H.264/AVC 符号化方式に関する一検討	2005 年 7 月	情報処理学会 AVM 研究会, Vol.2005 No.66, pp31-36	後藤 崇行 常松 祐一 渡辺 裕
(8)	H.264/AVC における Intra 予測モードの決定手法に関する検討	2005 年 9 月	電子情報通信学会論文誌 D-II, Vol.J88-D-II, No.9, pp1979-1981	常松 祐一 渡辺 裕

	題 名	発表年月	発表掲載誌	連名者
国内 (9)	H.264 Inter 予測における動き補償ブロックサイズ高速決定手法	2005年10月	情報処理学会 AVM 研究会, Vol.2005 No.98, pp11-16	常松 祐一 渡辺 裕