

2011年度 修士論文

次世代動画像符号化に適した並列補間手法の  
ハードウェア設計に関する研究

指導教授 戸川 望 教授

早稲田大学大学院 基幹理工学研究科  
情報理工学専攻

5110B044-3

栗岡 大生

2012年2月6日

# 目次

<b>1</b>	<b>序論</b>	<b>1</b>
1.1	本論文の背景と意義	2
1.2	本論文の概要	4
<b>2</b>	<b>H.264/AVC</b>	<b>5</b>
2.1	本章の概要	6
2.2	H.264/AVC の概要	7
2.3	ブロック分割	9
2.4	intra 予測	10
2.5	inter 予測	13
2.5.1	動き検出	13
2.5.2	小数精度動き補償	15
2.6	符号化	16
2.6.1	指数ゴロム符号	16
2.6.2	CAVLC	16
2.6.3	CABAC	17
2.7	本章のまとめ	20
<b>3</b>	<b>補間</b>	<b>21</b>
3.1	本章の概要	22
3.2	補間とは	23
3.3	H.264/AVC の補間方法	24
3.3.1	1/2 精度の補間方法	24
3.3.2	1/4 精度の補間方法	24
3.4	徳永らの手法	27
3.5	問題点	28
3.6	本章のまとめ	29
<b>4</b>	<b>次世代動画画像符号化に適した並列補間手法</b>	<b>30</b>
4.1	本章の概要	31

4.2	本提案手法と H.264/AVC との比較	32
4.3	仮定	34
4.4	可変窓	36
4.5	本章のまとめ	38
<b>5</b>	<b>実験・評価</b>	<b>39</b>
5.1	本章の概要	40
5.2	ソフトウェアシミュレーション結果	41
5.2.1	目的	41
5.2.2	方法	43
5.2.3	結果	44
5.3	Verilog-HDL による評価結果	49
5.3.1	目的	49
5.3.2	方法	50
5.3.3	結果	50
5.4	考察	51
5.5	本章のまとめ	52
<b>6</b>	<b>結論</b>	<b>53</b>
	謝辞	56
	参考文献	57
	本論文に関する発表業績	59

# 第1章

## 序論

## 1.1 本論文の背景と意義

情報化社会と呼ばれるこの時代では，個人が通信端末を持ち，毎日，インターネットで通信を行っている．これは通信媒体のめざましい進歩によりもたらされた．通信端末とはPCのみならず，携帯電話，ゲーム機，家電など多様である．これら様々な通信端末が情報を送受信するのに伴い，情報の形も多様化した．文字データから始まり，画像，音楽など多くの種類のデータが存在する．LSIの集積度が向上し，チップに載せられる回路も増えたため，そのデータを処理する速度も向上した．これにより，画像，音楽を別個とするのではなく，映像として扱うことが非常に多くなっている．したがって，今後，動画像を目にする機会が増えるため，画質の向上が必要となると考えられる．さらに，携帯など通信を行う端末で動画像を扱う場合，通信速度が限られるためにビット・レートの低下が必要であると予想される．

最近では，ストリーミング配信を用いた動画配信サイトの発展が目覚ましい．さらに，ワンセグによるテレビの視聴なども始まっており，2011年にはテレビ放送がアナログからデジタルへと切り替わり，映像デジタル技術が社会に与える影響は大きくなっている．したがって，動画圧縮技術を用いたサービスは身近となった．

しかし，文字や画像データと比較して，動画データは大きく，無圧縮で蓄積，送信するには実用的でない．このため，動画像圧縮技術は発展をしてきた．その中でもH.264/AVCは映像デジタル圧縮技術の最先端技術と言える．H.264/AVCはITU-T(International Telecommunication Standardization Sector)という標準化機関とMPEGを規定したISO(International Standardization Organization)/IEC(International Electrotechnical Commission)により定められている国際標準規格である．従来規格であるMPEG-2の二倍以上の圧縮率を実現している．これにより，携帯電話に代表されるような低速なモバイル機器でもテレビ電話が可能となっている．低速な通信だけでなく，ハイビジョンテレビ放送など，大容量を必要とする通信でも利用されており，幅広い用途で使われている．したがって，H.264/AVCは現在，最も注目されている技術の一つである．

現在，H.264/AVCの後継規格としてH.265/HEVCが議論されている．H.265/HEVCではH.264/AVCの二倍の圧縮率を目標としている．H.264/AVCでは計算が複雑にならないよう，できるだけ，単純であるが圧縮に効果的な演算方法を採用してきた．しかし，H.265/HEVCでは多少の複雑度は受け入れることにして，圧縮率の向上を目指している[8]．

中でも小数精度動き補償は計算の多くを占めている．その中の処理である補間は未だ，単純計算のみで計算している[3-6]．多少の複雑度を受け入れるとして，この補間の精度を上げることが出来れば，全体の圧縮率の向上に貢献できると考えられる．この補間の既存研究として，係数を動的に変える補間[10-12]や，その補間窓を静的に変えた補間[1,13]，動きベクトルを考慮して動的に補間窓を変える補間[14]などが挙げられる．係数を動的に変える補間

では、遅延オーバーヘッドとスループット低下が発生するという問題がある。これは一度、補間した結果を用いて係数を計算するためである。動きベクトルを考慮して動的に補間窓を変える補間では、圧縮する動画の特徴によってはあまり有効ではない場合がある。これらの問題を解決する必要がある。また、ハードウェア化を考えた場合、補間はループ処理で単純計算が多いのでハードウェア化に向いていると言える。

本論文では、画面間予測の小数精度動き補償での動き予測の精度向上を目的とし、次世代動画像符号化に適した並列補間手法を提案する。提案する次世代動画像符号化に適した並列補間手法とは、現行手法である 6-tap Wiener Filter による補間に加えて、ものの動き情報を考慮した補間を行い、動き検出の際に、誤差の少ない方の補間結果を採用する手法である。小数精度動き補償をする場合、整数画素間を埋める補間という作業をする。H.264/AVC では、 $1/2$  精度補間において、6-tap Wiener Filter による補間のみをする。そのため、ものの動きやフレームの特性を考慮できていないという問題がある。しかし、動き情報を加味する補間をすることによって、斜めに移動するものや激しく動くものに対して補間誤差が減少すると考えられる。さらに、現行手法を含む二種類の補間を行い、誤差の小さい方を採用するので、H.264/AVC と比較して、必ず、誤差が減少するのも特徴の一つである。また、ハードウェア化を考えた場合、完全並列可能なので、遅延オーバーヘッドが発生しないのも特徴である。だが、補償部にスループット低下と面積増加のペナルティがある。

## 1.2 本論文の概要

本論文は6章で構成される．以下に概要を示す．

第2章「H.264/AVC」では，現在のテレビ放送などで用いられている H.264/AVC 規格を説明する．フレームを画素の塊に分割するブロック分割，動画のフレーム予測として，intra 予測，inter 予測，それらの予測結果とオリジナルフレームとの誤差の符号化方法を示す．

第3章「補間」では，H.264/AVC の inter 予測で行われる小数精度動き補償の作業の一つである補間と徳永らの手法を説明する．補間には輝度補間と色差補間があるが，輝度補間のみを説明する．そして，その問題点を示す．

第4章「次世代動画像符号化に適した適応的な補間方法」では，H.264/AVC の補間を含む2種類の補間方法を用いて補間を行い，予測誤差の少ない方を採用する．もう1種類の補間では，動き情報を加味することにより，予測精度を上げることができるという仮定を用いて，可変窓を用いた補間方法を行う．このようなフレームに対して適応的な補間を行うことで予測誤差を減少させる手法を提案する．

第5章「実験・評価」では，4章で述べた手法を Reference Software JM[2] と Verilog-HDL に実装し，PSNR，ビット・レート，面積，遅延を評価する．JM では，ITU-T SG16 で定められた H.26L に関する推奨シミュレーション条件に基づいて実験を行い，PSNR とビット・レートの関係を示した RD 曲線を用いて評価をする．Verilog-HDL では，Design Compiler で論理合成をして，面積・遅延評価をする．

第6章「結論」では，本論文を総括し，今後の研究課題を示す．

## 第2章

# H.264/AVC



## 2.1 本章の概要

本章では最新動画画像圧縮技術である H.264/AVC の概要を述べる。

2.2 節「H.264/AVC の概要」では、H.264/AVC の歴史とエンコード・デコード処理の流れを説明する。

2.3 節「ブロック分割」では、フレームをマクロブロックとサブマクロブロックに分割する手順、適切な分割方法を説明する。

2.4 節「intra 予測」では、対象フレームの隣接するマクロブロックから画面を予測する画面内予測を説明する。

2.5 節「inter 予測」では、一枚または二枚のフレームを参照し、画面を予測する画面間予測を説明する。参照フレームと対象フレームを比較し、動きベクトルを算出する動き検出と小数精度で画面を予測する小数精度動き補償を述べる。

2.6 節「符号化」では、可変長符号化を説明する。最初に、指数ゴロム符号化、次に、CAVLC(Context-Adaptive Variable Length Coding)、最後に、CABAC(Context-Adaptive Binary Arithmetic Coding) を述べる。

## 2.2 H.264/AVC の概要

H.264/AVC が主要な動画像圧縮技術となった背景と圧縮処理の流れを述べる。

背景を説明する．動画像を圧縮し，情報量を削減することをエンコード，元に戻すことをデコードという．このような動画圧縮技術が利用されている分野は大別して以下の2つの分野に分類される．

- 放送・通信分野
- 蓄積分野

動画像符号化の標準化を行う機関は ISO/IEC やその下部組織である MPEG，ITU-T である．だが，当初は，分野ごとに標準化を行う機関が異なっていた．放送・通信分野では ITU-T が，蓄積分野では MPEG が主導権をとって標準化を行っていた．近年では，汎用的にどの分野でも利用できる方式が求められており，共同で標準化を行っている．参考に表 2.1 に H.26X

表 2.1: H.26X の動画像符号化方式の動向

	H.261	H.262	H.263	H.264
Standardization Year	1989	1994	1998	2003
Picture Type	P	I/P/B	I/P/B	I/P/B
Interlace	-	Frame/Field	Frame	Frame/Field
Macroblock Size	16x16	16x16	16x16	16x16 ~ 4x4
MC Accuracy	1	0.5	0.5	0.25
Orthogonal Transform	8x8 DCT	8x8 DCT	8x8 DCT	4x4/8x8 Integer Transform
Application	Video Conference on ISDN	Multipurpose	Mobile Contents	Multipurpose

の動画像符号化方式の動向を示す．

H.264/AVC の圧縮処理の流れを説明する．図 2.1 にエンコーダの例を示す． $F_n$  が圧縮対象となっているフレームで， $F_{n-1}$  が画面間予測をするときに参照されるフレームで， $F'_n$  が再構成フレームを示す．対象フレームに対して画面内，画面間の2モードの内から予測方法が1つ選択される．その予測結果と対象フレームとの差分を圧縮伸長し，符号化して伝送する．差分は逆圧縮伸長され，予測結果に足しあわせられ，再構成フレームとして格納される．格納されたフレームは参照フレームとして使われる．デブロッキングフィルタとはマクロブロック (画素の塊) の境界にできる歪を軽減するフィルタである．

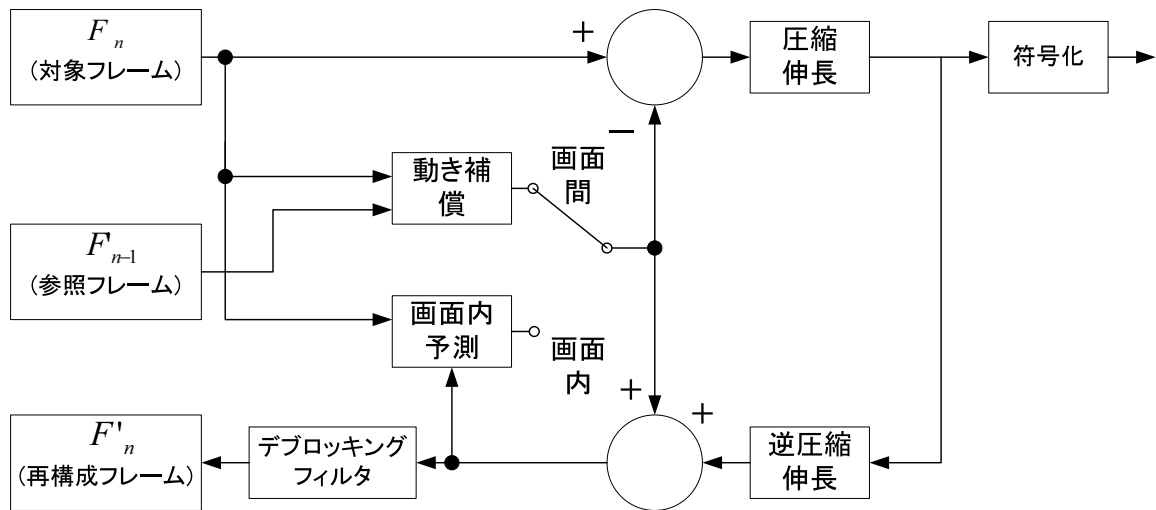


図 2.1: 動画圧縮のエンコーダの例 .

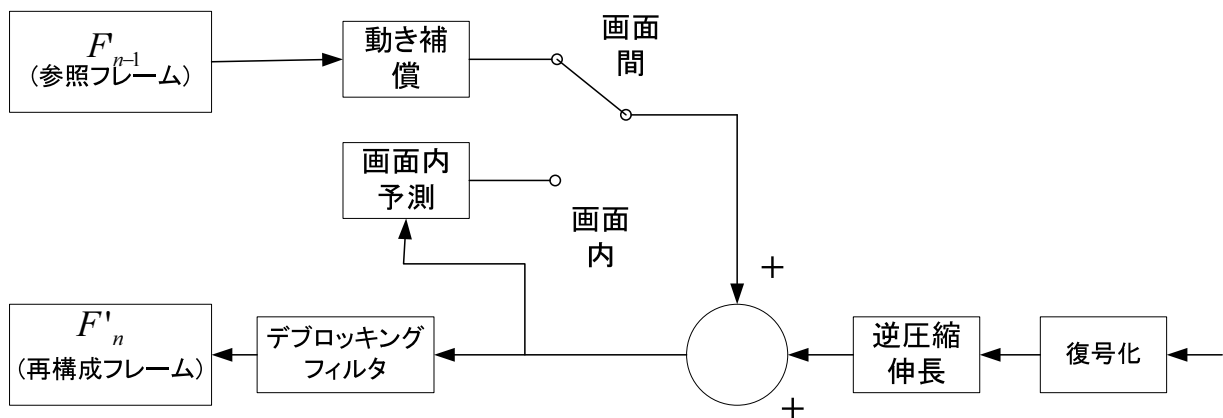


図 2.2: 動画圧縮のデコーダの例 .

図 2.2 にデコーダの例を示す．伝送されてきた情報を復号，逆圧縮伸長し，参照フレームを参考にエンコーダで行った予測モードと同じ方法で予測する．伝送されてきた情報は予測誤差であるので，予測結果にこれを足し合わせることでフレームを再現している．

伝送されるのは，フレームの画素値そのものではなく，オリジナルフレームと予測したフレームの誤差である．誤差は値が 0 に偏るため，0 に近い値に短い符号を割り当てることによって，伝送量を小さくできる．したがって，予測誤差が小さいとより小さく圧縮することができる．これにより，動画像を圧縮している．

## 2.3 ブロック分割

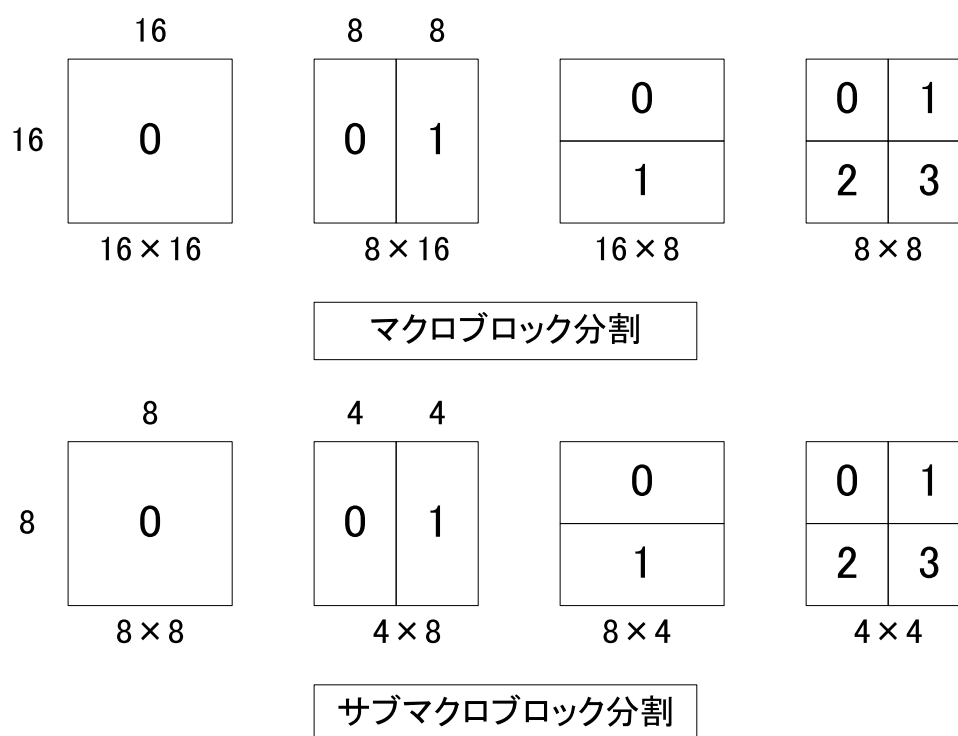


図 2.3: マクロブロックとサブマクロブロックの例。

ブロック分割とは、フレームをマクロブロック及びサブマクロブロックに分割する作業である。マクロブロックとは画素の固まりである。図 2.3 に示すように、 $16 \times 16$ 、 $8 \times 16$ 、 $16 \times 8$ 、 $8 \times 8$  のマクロブロックがある。分割の際、 $8 \times 8$  のマクロブロックが選択された場合、そのマクロブロックをさらに小さく分割することがある。これをサブマクロブロック分割という。 $8 \times 8$ 、 $4 \times 8$ 、 $8 \times 4$ 、 $4 \times 4$  のサブマクロブロックがある。画面予測は画素単位ではなくブロック単位で行う。

フレームをどのようにブロック分割するかを説明する。 $16 \times 16$  のような大きなマクロブロックに分割する場合は、各ブロックの分割情報が小さくなるが、精度の粗い予測となる。 $4 \times 4$  のような小さなサブマクロブロックに分割する場合は、精度は良いが、各ブロックの分割情報が大きくなる。したがって、色の変化が少なく、平坦な部分には大きなマクロブロックに、色の変化が大きく、細かい部分には小さなマクロブロックに分割する。ブロック分割の具体例を図 2.4 に示す。

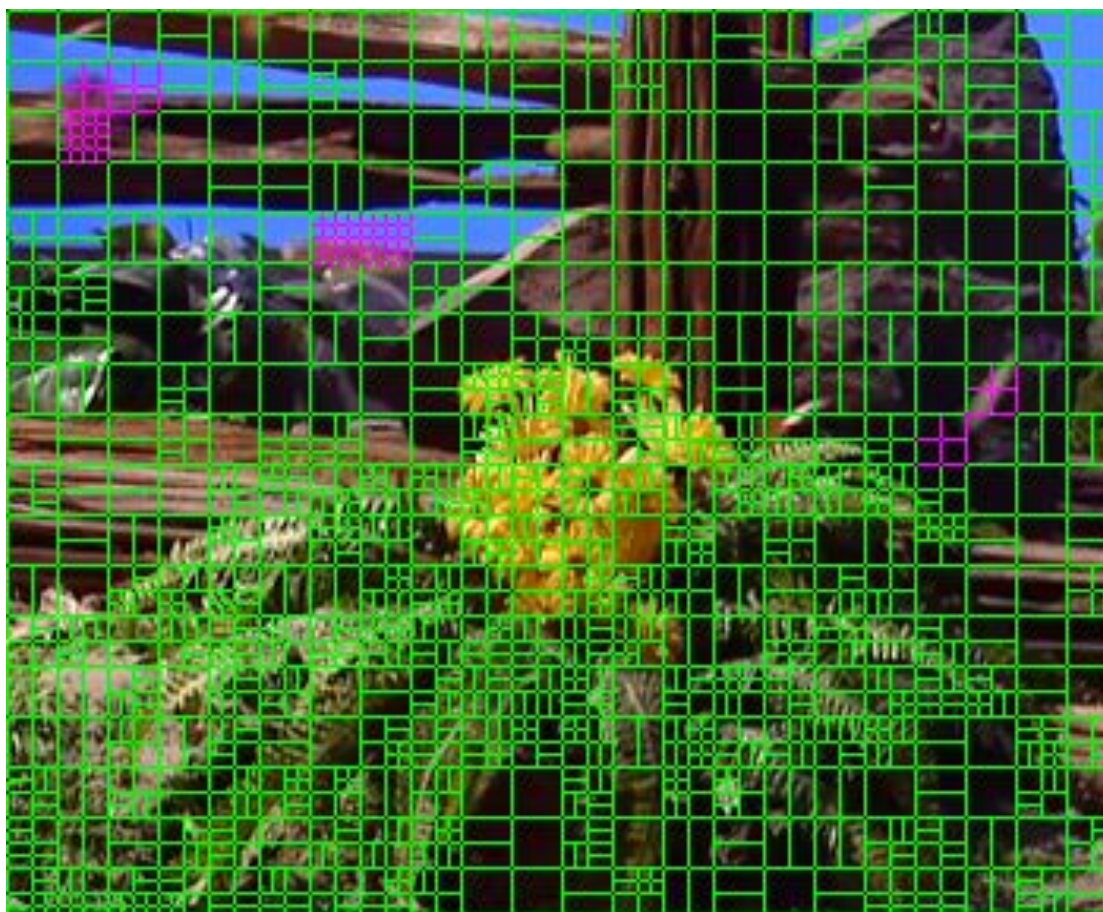


図 2.4: ブロック分割の例 .

## 2.4 intra 予測

intra 予測とは対象フレーム内のエンコード済みのマクロブロックを参考に、エンコード対象となっているマクロブロックを予測する方法である。参考とされるマクロブロックは、対象となっているマクロブロックの左上、上、右上、左に位置するマクロブロックである。intra 予測には 4x4intra 予測と 16x16intra 予測がある。マクロブロックの大きさによって予測方法は異なる。

まず、4x4intra 予測を説明する。図 2.5 に全ての予測モードを示す。対象マクロブロックに対し、この中から最適な予測モードを選択し、参考とされるマクロブロックの画素値をコピーする。モード 2 に示す平均値とは、左のブロックの 4 画素と上のブロックの 4 画素の計 8 画素で平均を取り、対象ブロックのすべての画素値を予測するモードである。ただし、対象ブロックが最左端や最上端にある場合、左側の画素や上側の画素を参照とする予測モードの選択を禁止する。

予測方法を詳しく示す。通常、対象ブロックの予測は隣接ブロックの予測との相関が高く



図 2.5: 4x4intra 予測モード .

なる．したがって，予測モードの選択は，隣接するブロックを参考とする．具体的には符号化済み隣接ブロックの予測方向から，対象ブロックの予測方向を予測する．左ブロックの予測方向に対応する予測モードと，上ブロックの予測方向に対応する予測モードを比較し，小さい番号の予測モードを選択する．これは，予測モード番号が小さい方が短い符号が割り当てられていることによる．

次に，16x16intra 予測を説明する．図 2.6 に全ての予測モードを示す．予測方法は 4x4intra 予測と同様で，対象マクロブロックに対し，この中から最適な予測モードを選択し，参考とされるマクロブロックの画素値をコピーする．しかし，対象ブロックの画素の数が 4x4intra 予測の場合は 16 個であったが，16x16intra 予測の場合は 256 個となる．したがって，モード 2 の平均値も左のブロックの 16 画素と上のブロックの 16 画素の計 32 画素で平均を取り，対象ブロックのすべての画素値を予測する．

4x4intra 予測と比較して 16x16intra 予測は，予測方向の符号化に必要なビット数が少ないため，平坦な画像部分で用いると，大幅な圧縮が可能である．

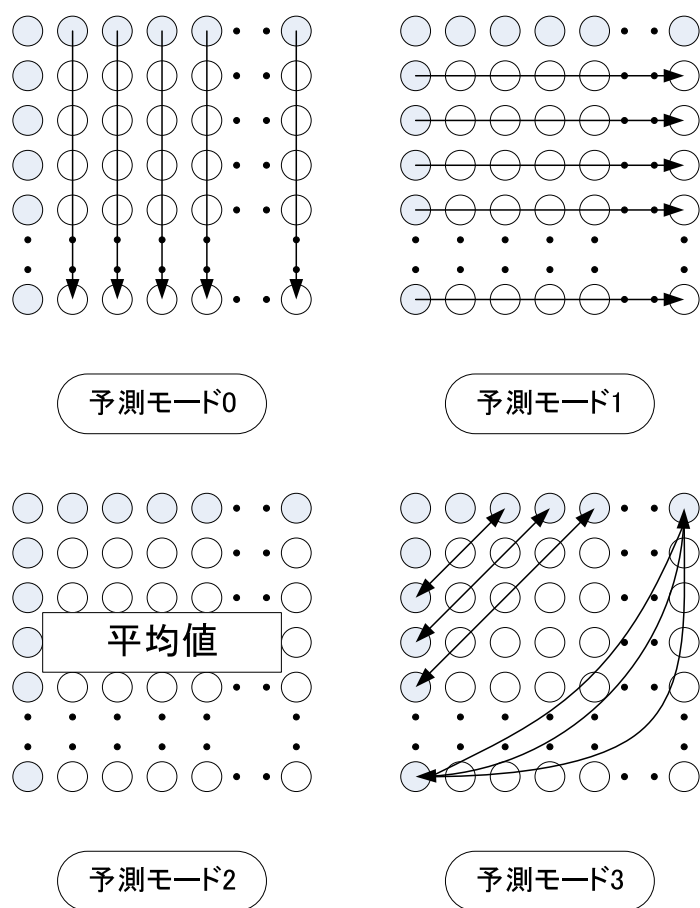


図 2.6: 16x16intra 予測モード .

## 2.5 inter 予測

intre 予測とは，圧縮対象となっているフレームに対して，他のフレームを参照することにより，動きを検出し，ブロック単位で予測をすることである．場合によっては，画素間を補間し，小数精度で予測を行うことがある．これを，小数精度動き補償という．

動き検出を行う際に，参照するフレームによって P ピクチャと B ピクチャに分別される．P ピクチャとは，順方向予測のことであり，I ピクチャ (intra 予測により得られるフレーム)，P ピクチャにより得られる予測画像である．ただし，参照できるのは時間軸上で，当該フレームより過去に存在するマクロブロックを用いた順方向 (前方向) の予測のみである，B ピクチャとは，過去と未来の双方向からの予測符号化により得られる予測画像である．つまり，I ピクチャ，P ピクチャを参照とする．時間軸上で，当該フレームより過去，もしくは未来，もしくはその両方に存在するマクロブロックを用いたそう予測ができる．

表 2.2: I, P, B ピクチャの特徴

	画面内予測	前方向予測	後向予測	双方向
I ピクチャ		×	×	×
P ピクチャ	×		×	×
B ピクチャ	×			

表 2.2 に各ピクチャの特徴を示した．I ピクチャは画面内予測のみ可能であり，P ピクチャは時間軸上で前に存在するフレームを参照とすることができ，B ピクチャは時間軸上で前後どちらに存在するフレームも参照とすることができる．

以降，参照ピクチャを用いた動き検出方法と小数精度動き補償を説明する．

### 2.5.1 動き検出

動き検出とは動き探索とも呼ばれ，周辺ブロックを参考に対象ブロックの動きベクトルを算出する方法である．本節で整数精度の動き検出を説明し，次節で小数精度を説明する．

整数画素精度の動き検出では，周辺ブロックに対する動きベクトル値から，メディアン予測 (中央値による予測) により，対象ブロックに対する動きベクトルの予測値を生成する．動きベクトル予測値を  $pmv$ ，探索の中心を Search Center，設定された探索範囲を Search Range とすると，絶対座標は  $(0,0)$  ではなく  $pmv$  とし，探索範囲は  $pmv \pm \text{Search Range}$  にして，絶対座標  $(0,0)$  が含まれるように Search Center を調整する．

このように求められた Search Center を起点として，図 2.7 に示したようにらせん状の順番で動きベクトルの探索をする．これを，スパイラル探索という．スパイラル探索は，コスト関数を用い，最適な箇所を探索する．探索において，0 番目と 15 番目のコスト関数の値が



15	9	11	13	16
17	3	1	4	18
19	5	0	6	20
21	7	2	8	22
23	10	12	14	24

Search Center  
探索の中心点

図 2.7: スパイラル探索の例 .

同じであった場合 , より  $pmv$  に近い前者を選択する . これは ,  $pmv$  に近い方が符号量が少ないことによる .

## 2.5.2 小数精度動き補償

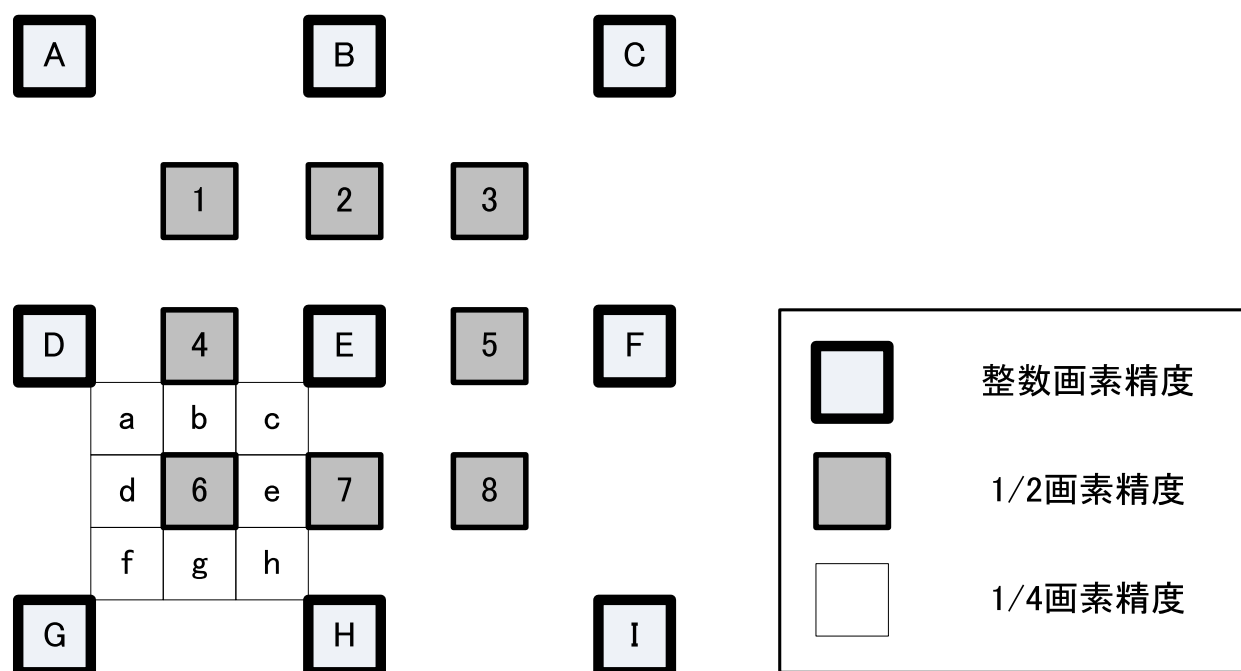


図 2.8: 小数精度動き探索の例 .

小数精度動き探索の例を図 2.8 に示す．太枠で示された A ~ J は整数画素精度で，中太枠で示された 1 ~ 8 は 1/2 画素精度で，細枠で示された a ~ h は 1/4 画素精度である．整数精度の動き検出で E が最小コストであると判断されたとする．このとき，E およびその周辺の 1/2 画素精度の 1 ~ 8 に関して，最小のコストを与える点を求め，これを 1/2 精度での最適な動きベクトル値とする．その結果，6 が 1/2 画素精度の探索による最適点であるとする．このとき，6 およびその周辺の 1/4 画素精度の a ~ h に関して，最小のコストを与える点を求め，最適動きベクトル値とする．

この最適動きベクトル値だけシフトした参照フレームの箇所と，対象フレームの該当箇所の誤差を取り，出力する．これを小数精度動き補償と呼ぶ．直接，画素値を送信するよりも誤差を送信した方が符号量が小さくて済む．これを利用した圧縮をしている．

## 2.6 符号化

H.264/AVC では、シンタックス要素のエントロピー符号化方式として、単純な表による変換の他に高効率な符号化方式が用意されており、高い符号化効率が必要とされる DCT 係数の符号化時などに適用される。規格では、指数ゴロム符号と呼ばれる基本的な符号の他に、CALVC(Context-Adaptive Variable Length Coding) というコンテキスト適応型可変長符号化方式と CABAC(Context-Adaptive Binary Arithmetic Coding) というコンテキスト適応型 2 値算術符号化方式の 2 種類の高効率符号化方式が用意されている。ここで、コンテキスト適応型とは周囲の状況に応じて効率の良い符号化方式を適応的に選択することである。

以降、指数ゴロム符号、CALVC、CABAC を順に整理する。

### 2.6.1 指数ゴロム符号

指数ゴロム符号とは、基本的なシンタックス要素の符号化に適応される符号である。

表 2.3: 指数ゴロム符号の構成

指数ゴロム
1
0 1 x
0 0 1 x x
0 0 0 1 x x x
0 0 0 0 1 x x x x
0 0 0 0 0 1 x x x x x
⋮
⋮

表 2.3 に示すように、'0' の連続 (プリフィックス) のあとにセパレータとして '1' を挟み、'x' で書かれた '0' と '1' の組み合わせ (サフィックス) を挿入する。指数ゴロム符号はこのような構成を持ち、いったん表 2.4 のように非負整数として復号された後、それぞれのシンタックス要素の値との対応づけが行われる。

### 2.6.2 CAVLC

CAVLC は、量子化された DCT 係数値の符号化にのみ用いられる。図 2.9 に示すように、DCT 係数をジグザグスキャンによって一次元のベクトルとした後、係数値や連続する '0' の長さ (ラン) など、復号に必要な情報を符号化している。具体例を図 2.10 に示す。H.264/AVC では、MPEG-2 などのようにランとレベルを組み合わせたものに対する符号化をするのでは

表 2.4: 符号番号の復号

指数ゴロム	符号番号
1	0
0 1 0	1
0 1 1	2
0 0 1 0 0	3
0 0 1 0 1	4
0 0 1 1 0	5
0 0 1 1 1	6
0 0 0 1 0 0 0	7
0 0 0 1 0 0 1	8
0 0 0 1 0 1 0	9
...	...

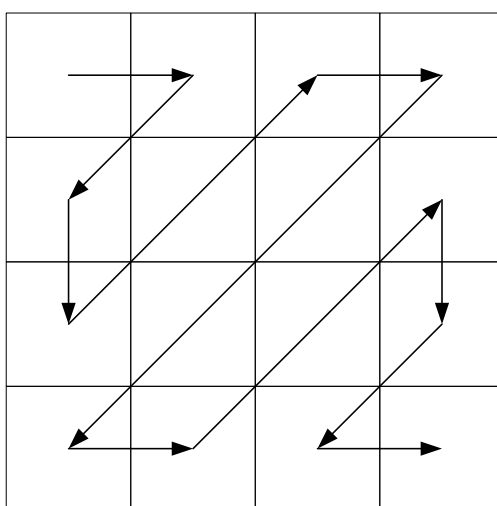


図 2.9: ジグザグスキャンの例 .

なく、ランとレベルそれぞれの VLC(可変長符号化) テーブルを用いてスキャンの逆順方向から別々に符号化し、復号側ではそれらを組み合わせることで係数値の復号をする。

ランとレベル以外の情報である，非 0 係数の個数と最後に連続する絶対値 1 の係数の個数の符号化の際に，周囲のマクロブロックの状態に応じて，複数の VLC テーブルのうちの 1 つを適応的に選択して符号化をする．これが，CALVC の符号化効率が高い理由の一つである．

### 2.6.3 CABAC

CABACは図 2.11 に示すように、通常 2 値算術符号化器に加えて、多値信号を 2 値信号に変換するための 2 値化部と符号化すべき 2 値信号の発生確率を周囲の状況に応じて計算・更

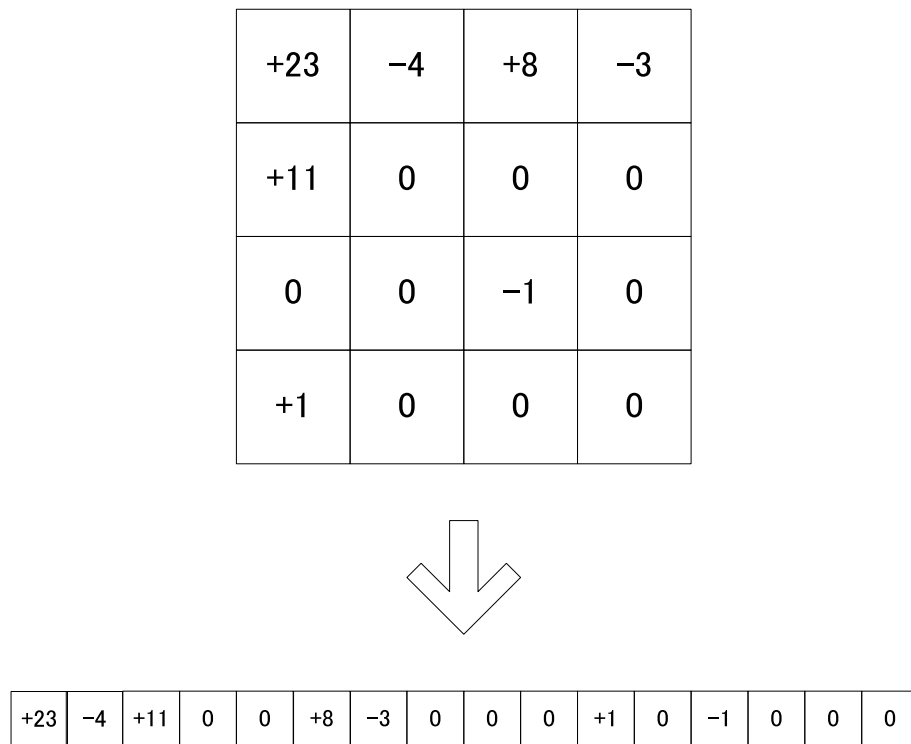


図 2.10: ジグザグスキャン順に量子化された DCT 係数の例 .

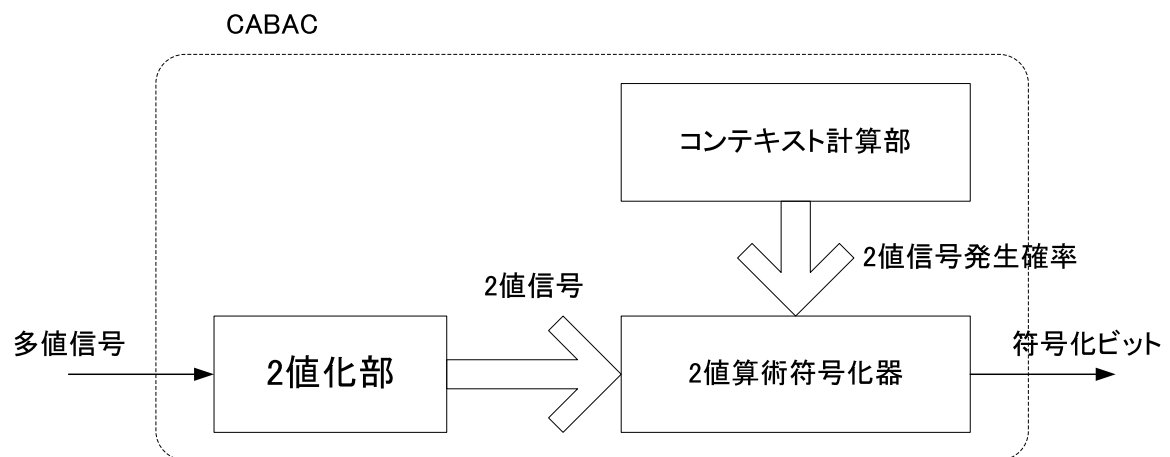


図 2.11: CABAC の構成の例 .

新するコンテキスト計算部から成る .

算術符号化とは、定常的な信号を理論限界まで圧縮することが可能な符号化方式である . CABAC 内部の 2 値算術符号化器も、これと同じ動作をする . 実際には、区間幅を保持する記憶素子の桁数が有限であるため、出力すべき区間上位ビットが確定した段階で適時確定ビットを出力し、区間幅を広げる再正規化と呼ばれる処理をする . また、CABAC はバイパスモードという、まったく圧縮を行わない特殊なモードもある .

コンテキスト計算部では、2 値信号の発生確率を複数保持しており、現在の符号化対象や周囲の状況に応じて切り替えて 2 値算術符号化部に与える。2 値信号の発生確率は、発生確率の高い 0, 1 いずれかのシンボルとその発生確率テーブルの番号で保持されている。

2 値化部では、多値信号から 2 値信号への変換をする。それぞれのシンタックス要素に応じて適用される 2 値化の方式は異なる。

## 2.7 本章のまとめ

本章ではH.264/AVCの処理の流れを説明した。まず、規格化の歴史とエンコーダ・デコーダの処理を示した。エンコーダ・デコーダにおいて、同じ予測をして、予測誤差を足しあわせることにより、フレームを再現している。誤差を伝送することにより、圧縮を行っていることがわかった。次に、フレームを画素の塊であるマクロブロックに適切に分割する方法を整理した。色の変化が少ない平坦な箇所には小さなマクロブロックを、色の変化が大きく細かい箇所には大きなマクロブロックを割り当てるのが適当であった。intra 予測では、対象フレーム内のエンコード済みマクロブロックを用いて予測する方法を説明した。予測相関度が高いことから、周辺ブロックを用いて対象ブロックを予測する方法であった。inter 予測では、ブロック分割で分けたマクロブロック単位で、複数のフレームを用い、動きベクトルを使った予測方法を示した。動き探索において整数画素間を埋めて探索、誤差の出力をする小数精度動き補償を述べた。符号化では、H.264/AVC で用いられている指数ゴロム、CAVLC、CABAC の3つの符号化方式を整理した。以上のような予測、符号化をすることによって動画像を圧縮している。だが、次世代動画像符号化に向けてさらなる予測精度向上、適切な符号化による圧縮率向上をする必要がある。

次章では、小数精度動き補償の処理の一部である補間を詳しく説明する。H.264/AVC で用いられている 6-tap Wiener フィルタによる補間を示し、その問題点を考える。

## 第3章

### 補間



### 3.1 本章の概要

本章では小数精度動き補償の作業の一つである補間を説明する．

3.2 節「補間とは」では，補間とはどのような処理であるかを述べる．

3.3 節「H.264/AVC の補間方法」では，H.264/AVC で用いられている補間方法を整理する．補間は  $1/2$  精度の補間と  $1/4$  精度の補間があり， $1/2$  精度の補間では 6-tap Wiener フィルタによる補間が， $1/4$  精度の補間では Bi-linear フィルタによる補間が行われる．この二つの補間方法を述べる．

3.4 節「徳永らの手法では，動きベクトルに合わせて適応的に入力輝度値を変化させる手法を説明する．

3.5 節「問題点」では，H.264/AVC の補間方法と徳永らの手法の問題点を考察する．

## 3.2 補間とは

補間とは，整数精度の画素値を入力とし，整数精度の画素間を埋めた小数精度の画素値を出力とする処理である．画素値とは輝度値，色差値の両方を指す．だが，色差値の変化は輝度値の変化と比較して，人の目に対して鈍感である．色差値の PSNR と輝度値の PSNR を同程度改善したとしても，輝度値を改善した方が人の目で見ると良く感じるということである．したがって，本章では，輝度値の補間のみに着目する．

補間処理を定式化すると以下のように表すことができる．

$$p^{sp} = \sum_i \sum_j h_{i,j} p \quad (3.1)$$

$p^{sp}$  は出力となる小数精度の輝度値， $h$  はフィルタ係数， $p$  は入力となる整数精度の輝度値である．出力となる小数精度の輝度値の周辺の整数精度の輝度値を入力する．入力された輝度値と係数を掛けて足し合わせることにより，小数精度の輝度値を計算し，出力する．

### 3.3 H.264/AVC の補間方法

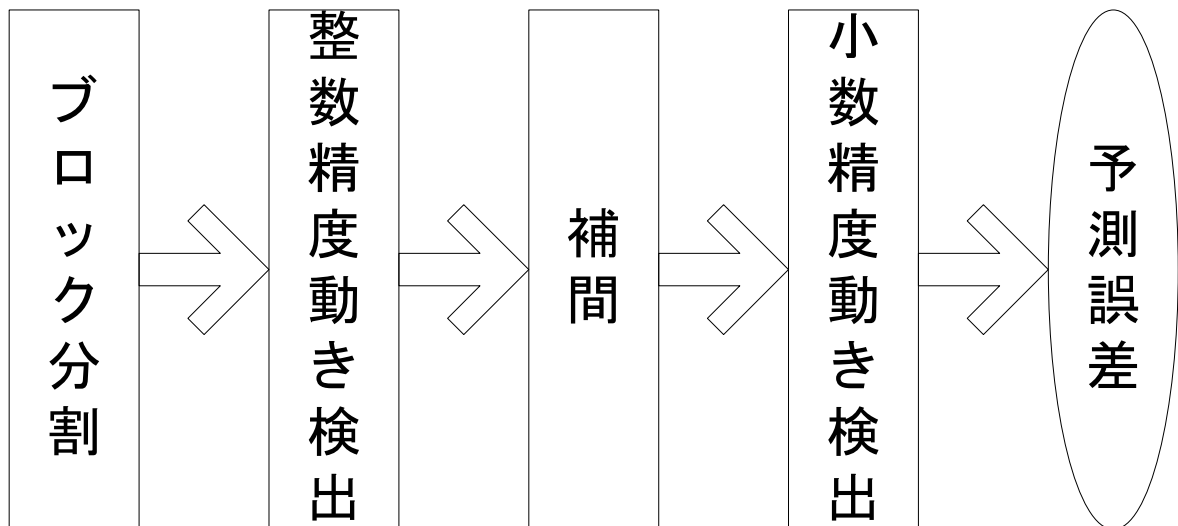


図 3.1: 小数精度動き補償の流れの例．

小数精度動き補償の流れを図 3.1 に示す．補間とは，小数精度の動き補償を行う場合に，整数精度から小数精度に落とし込むために整数画素間を埋める作業である．補間の精度は  $1/2$  精度と  $1/4$  精度があり，それぞれ補間方法が異なる． $1/2$  精度の補間は，整数画素， $1/2$  精度の画素を用いて， $1/4$  精度の補間は，整数画素， $1/2$  精度の画素， $1/4$  精度の画素を用いて行う．これより順に，2 つの補間方法を整理する．

#### 3.3.1 $1/2$ 精度の補間方法

$1/2$  精度の補間方法とは図 3.2 の太枠で示された整数画素と破線で示された  $1/2$  精度の画素を用いて， $1/2$  精度の画素を埋める作業のことである．式で表すと以下ようになる．

$$b = \min(255, \max(0, (E - 5F + 20G + 20H - 5I + J) \div 32)) \quad (3.2)$$

アルファベットで表したのは図に示した場所の輝度値である．フィルタ係数は  $\{1/32, -5/32, 20/32, -5/32, 1/32\}$  で固定係数であり，横方向と縦方向の画素を用いた補間方法がある．このフィルタを 6-tap Wiener フィルタという．どのフレームに対しても  $1/2$  精度の補間は同様の補間を行う．

#### 3.3.2 $1/4$ 精度の補間方法

$1/4$  精度の補間方法とは図 3.2 の太枠で示された整数画素と破線で示された  $1/2$  精度の画素と実線で示された  $1/4$  精度の画素を用いて， $1/4$  精度の画素を埋める作業のことである．式

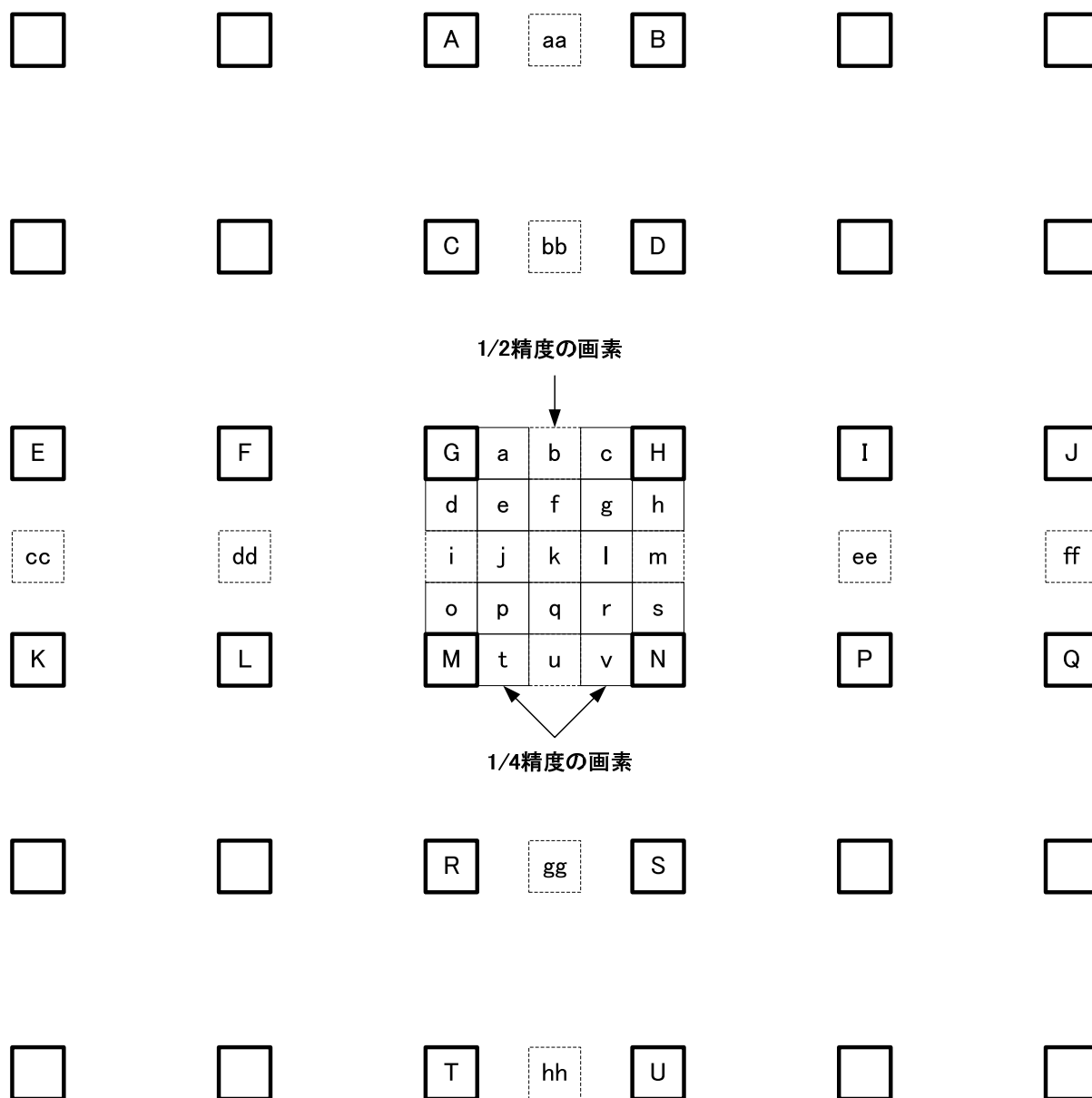


図 3.2: 補間の例 .

で表すと以下のようになる．

$$t = (M + u) \div 2 \quad (3.3)$$

アルファベットで表したのは図に示した場所の輝度値である．フィルタは 2-tap の平均値フィルタであり，このフィルタを 2-tap Bi-linear フィルタという．どのフレームに対しても 1/4 精度の補間は同様の補間を行う．

### 3.4 徳永らの手法

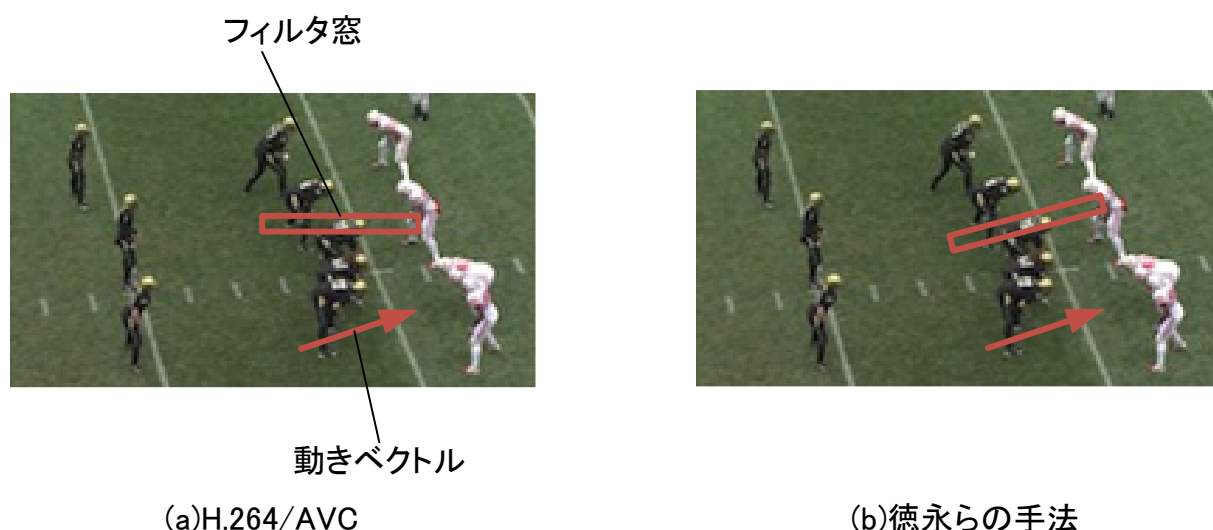


図 3.3: 徳永らの手法の例 .

徳永らの手法を説明する．一般的に， $1/4$  精度の補間を改善するよりも， $1/2$  精度の補間を改善した方が，PSNR の向上が大きいと言われている． $1/4$  精度の補間に処理を加えると，演算量も膨大になってしまう可能性もある．したがって，徳永らの手法は， $1/2$  精度の補間のみを対象としている．H.264/AVC の補間では， $1/2$  精度の補間の入力は一列の整数精度の輝度値のみである．この場合，被写体の動きが激しい時や斜めに動く時に予測精度が低下するという問題点がある．入力輝度値を，整数精度動き検出で検出した動きベクトルに基づいて，適応的に変化させることにより，激しい動きや斜めに対する動きにも対応させる手法である．

具体例を図 3.3 に示す．H.264/AVC では，被写体が斜めに動いているのに対して，使用する周辺画素 (フィルタ窓) が横一列の画素となっている．徳永らの手法では，被写体が斜めに動いている場合，動きベクトルにより判断して，適応的にフィルタ窓を変化させる．フィルタ窓は 7 種類から選択される．これにより，PSNR が向上すると徳永らは主張している．

### 3.5 問題点

H.264/AVC の補間と徳永らの手法においての問題点を考察する．H.264/AVC ではどんなフレームに対しても同じフィルタ補間を行っており，フレームの特性を活かした補間が行えていない．

例えば，上下左右一列に対するフィルタ補間はしているが，斜めに対するフィルタ補間はされていない．また，動きの激しいものに対しては適切な補間ができていない．したがって，ものが斜めに動いている場合や激しく動いている場合において，精度が下がるという問題がある．さらに，平坦な箇所はよいのだが，変化の激しいエッジに対しても同様に周辺画素から補間をしている．したがって，フレーム特性を活かした補間はされていないという問題がある．

徳永らの手法では，斜めに動く場合や動きが激しい場合は適しているが，少しの動きにも反応してしまい，フィルタ窓を変化させてしまう．激し過ぎる動きに対しても適切な補間ができないと考えられる．したがって，不安定な補間である．

このことから，ものの動きやエッジ情報を考慮しつつ，安定的な補間をすることにより，予測精度の向上が見込めると考えられる．

### 3.6 本章のまとめ

本章では、H.264/AVC の補間方法と徳永らの手法を説明した。

補間とは画素値に係数を掛けて足し合わせることにより、間の画素値を埋める作業であった。1/2 精度の補間では 6-tap Wiener フィルタによる補間、1/4 精度の補間では 2-tap の Bilinear フィルタによる補間がされていた。これらのフィルタは固定係数であり、どのフレームに対しても同じ補間を行うため、斜めの動きやエッジの情報を考慮していないという問題点を述べた。

一方、徳永らの手法は動きに対して適応的である一方、安定的ではないという問題点を述べた。

次章では、この問題点を考慮し、仮定を立て、画面間予測の小数精度動き補償での動き予測の精度向上を目的とし、次世代動画像符号化に適した適応的な補間方法を提案する。



## 第4章

# 次世代動画像符号化に適した並列補間手法

## 4.1 本章の概要

本章では H.264/AVC の補間方法の問題点を考慮し，画面間予測の小数精度動き補償での動き予測の精度向上を目的とし，次世代符号化に適した適応的な補間方法を提案する．

4.2 節「本提案手法と H.264/AVC との比較」では，本提案手法と H.264/AVC の補間手法との比較をして，本提案手法の特徴を述べる．提案手法とは2種類の補間をして，誤差の少ない方の結果を採用する方法である．2種類の補間とは，6-tap Wiener フィルタによる補間と，可変窓による補間である．この2種類の補間をすることによって，お互いの弱点を補強しあう．

4.3 節「仮定」では，前章で挙げた補間方法の問題点を基に仮定をする．誤差の生じやすい場合を想定する．この仮定より，可変窓による適応的な補間方法を提案する．

4.4 節「可変窓」では，ブロックの動きベクトルを加味し，適応的に補間窓を変化させる手法を説明する．補間窓は全部で8種類あり，動きベクトルの方向によって変化する．

## 4.2 本提案手法とH.264/AVCとの比較

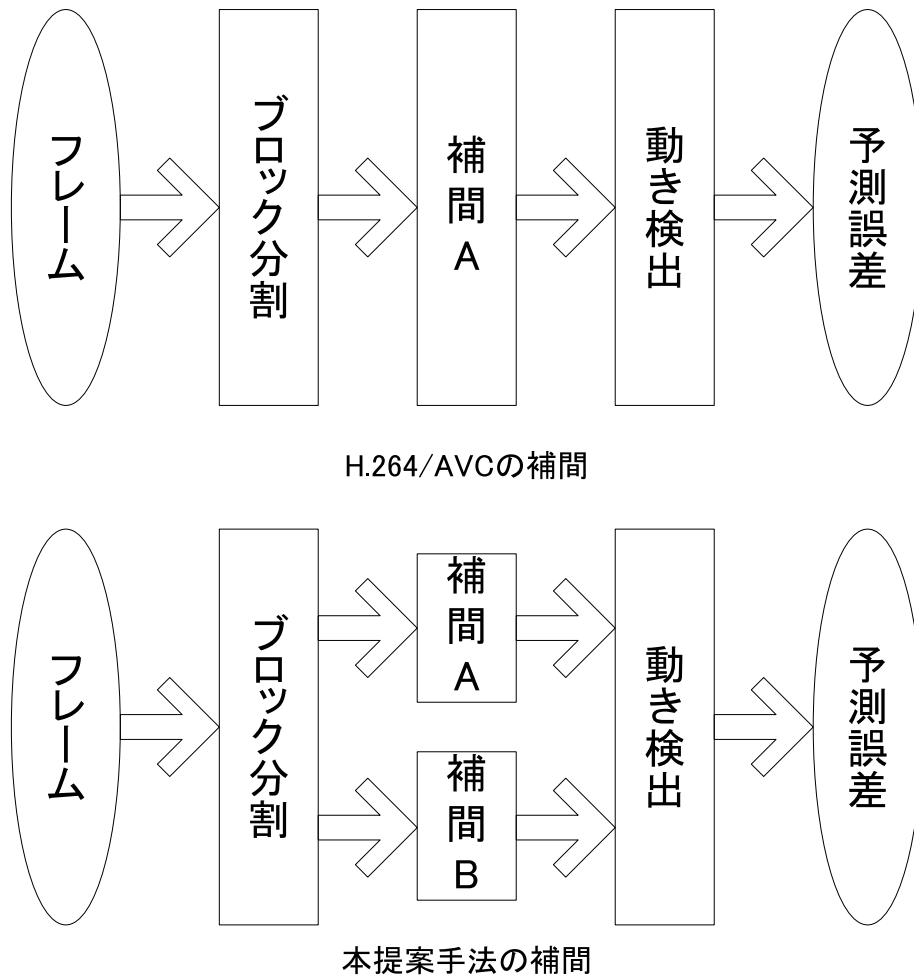


図 4.1: H.264/AVC の補間と本提案手法との比較の例 .

図 4.1 に H.264/AVC による補間と本提案手法との比較の図を示す。H.264/AVC では、6-tap Wiener フィルタによる補間のみをしていた。しかし、本提案手法では補間 A、補間 B の 2 種類による補間を行い、予測誤差が少なかった方の結果を採用する。したがって、H.264/AVC による補間より画質が下がることはない。

次に、補間 A と補間 B でどのような補間をするかを説明する。補間 A では、従来通りの 6-tap Wiener フィルタによる補間をする。これは、H.264/AVC でも採用されている通り、安定した補間を行えるからである。だが、斜めの動きやエッジを考慮していないという問題がある。補間 B では、徳永らの手法を参考とし、ベクトル情報を考慮した可変窓による補間をする。だが、動き検出に誤差がなければ不安定であるという問題がある。これより、お互いの弱点を補強し合うことができると考えられる。補間 A と補間 B の特徴を比較したものを

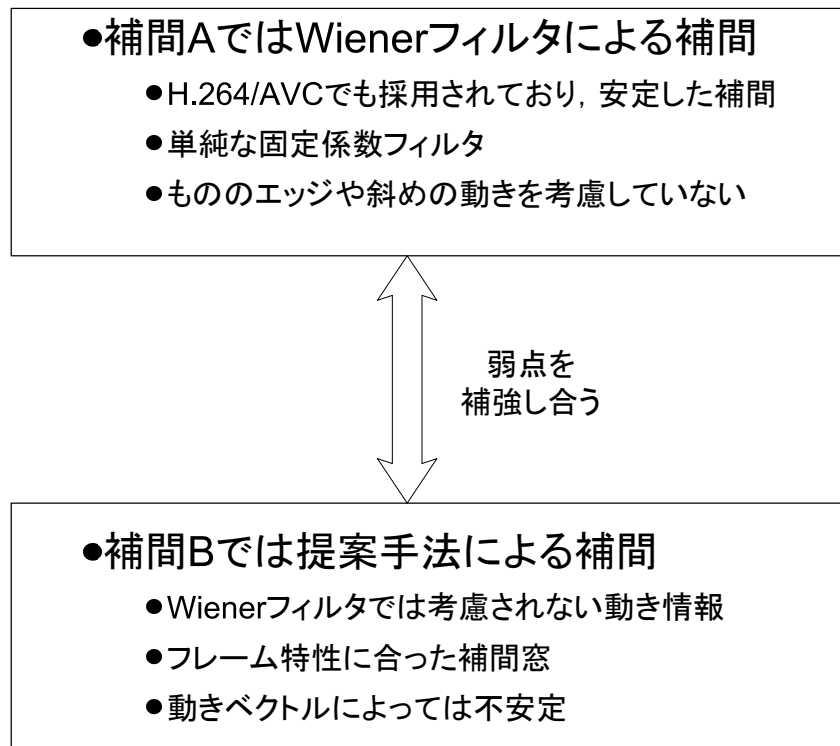


図 4.2: 補間 A と補間 B の特徴の比較の例 .

図 4.2 に示す .

### 4.3 仮定

H.264/AVC の補間による問題点より以下の仮定をする.

- 動きの大きい箇所では誤差が生じやすい

動きが激しい箇所，特に，斜めに動く箇所では誤差が生じやすいと考えられる．さらに，エッジでは誤差が生じやすいと考えられる．この仮定を説明する．

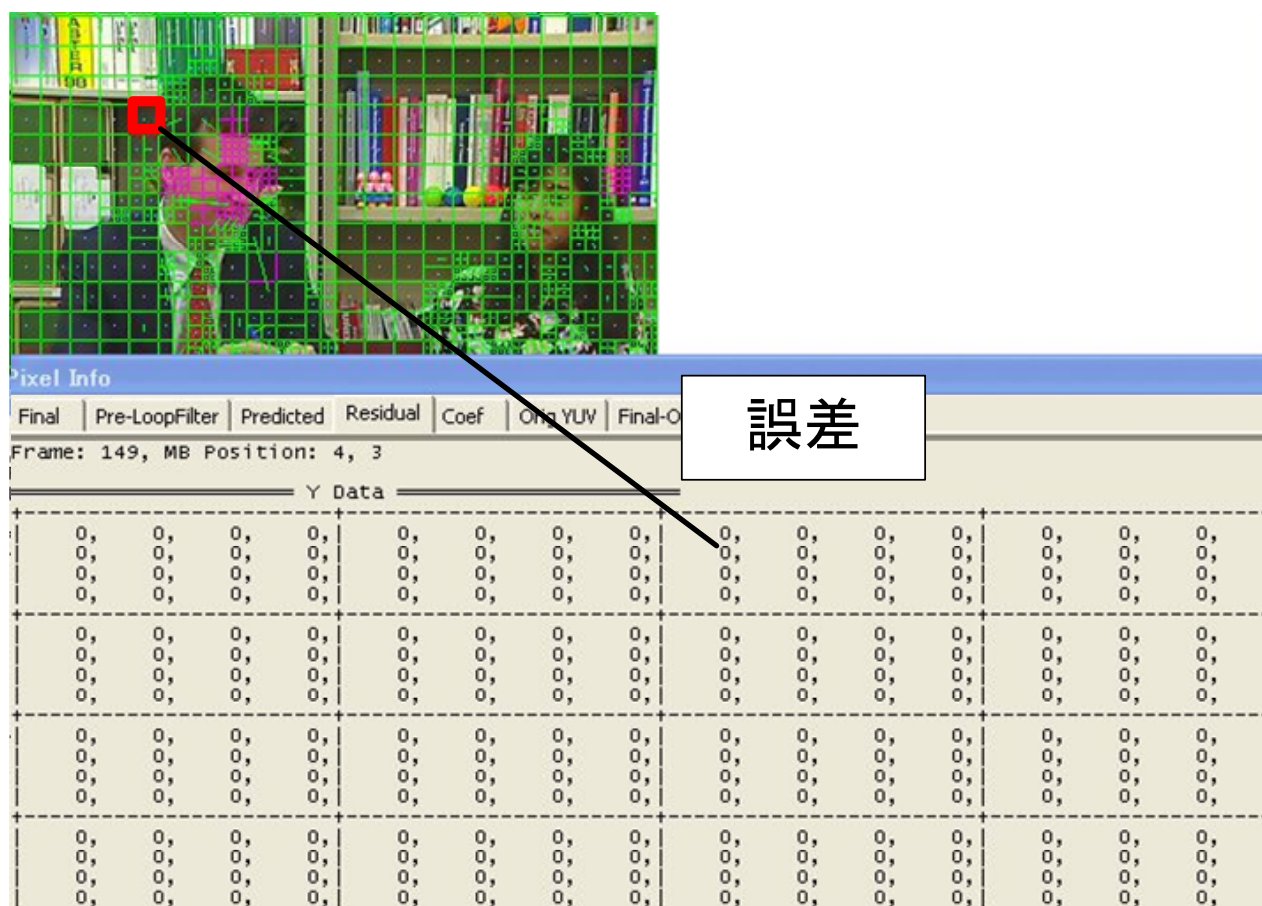


図 4.3: ベクトルが小さい場合の誤差の例.

図 4.3, 4.4 にベクトルが大きい場合と小さい場合の誤差の状況を示す．これは paris という動画の 149 フレーム目である．ブロック分割がされており，ブロックの中心から線でベクトルが表されている．赤の太枠で示した箇所の誤差を示している．ベクトルの小さいブロックでは誤差は 0 である．だが，ベクトルの大きいブロックでは誤差が発生していることが分かる．これより，動きの大きい箇所では誤差が生じやすいという仮定を立てる．

以上，動きの大きい箇所では誤差が生じやすいという仮定を立てた．この仮定より，徳永らの手法を参考とし，動きベクトルを考慮した可変窓による並列補間方法を提案する．

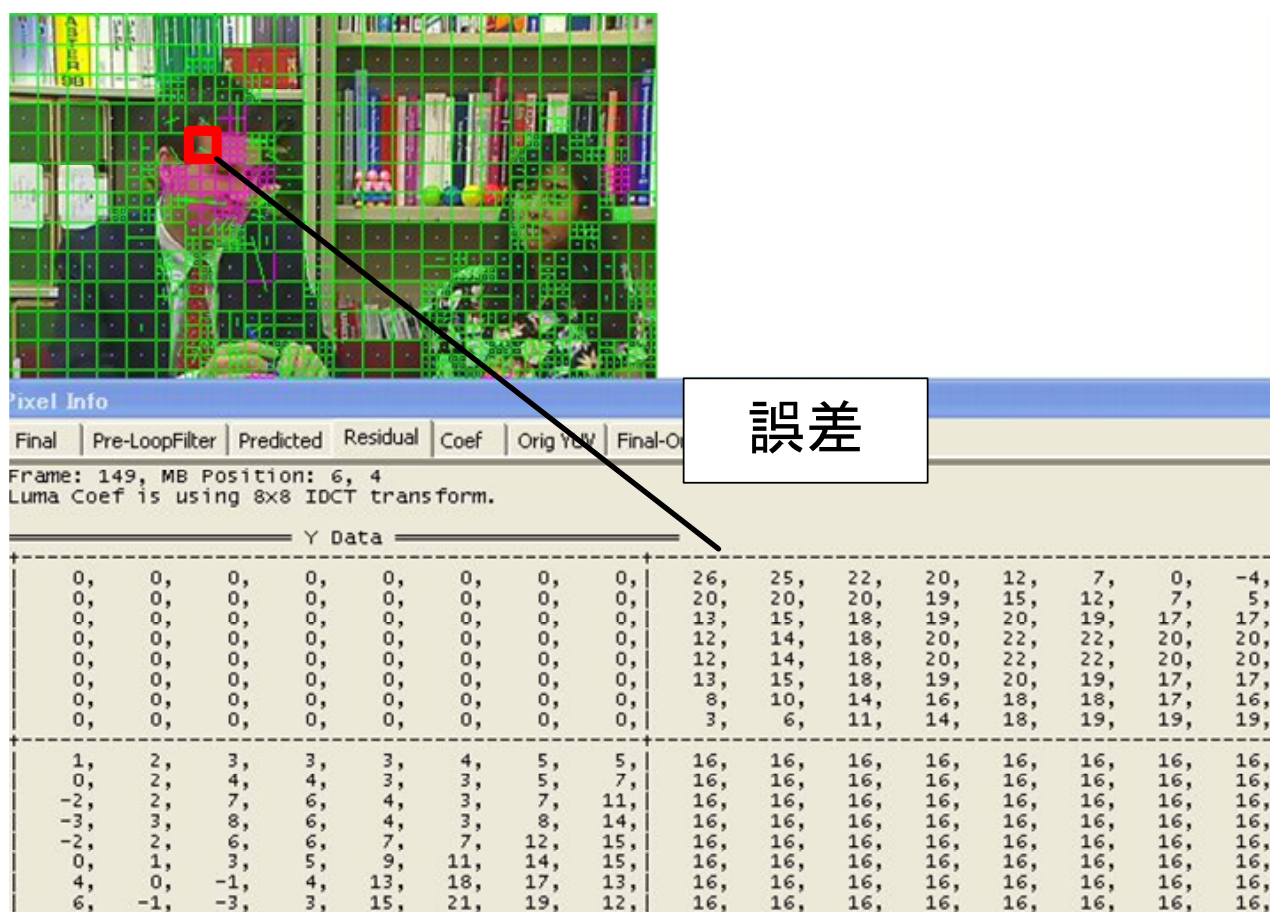


図 4.4: ベクトルが大きい場合の誤差の例.

H.264/AVC の補間にこの可変窓による補間を並列にすることにより，互いの欠点を相殺し合い，利点を活かすことができると考えられる．詳細は次節より順に説明する．

## 4.4 可変窓

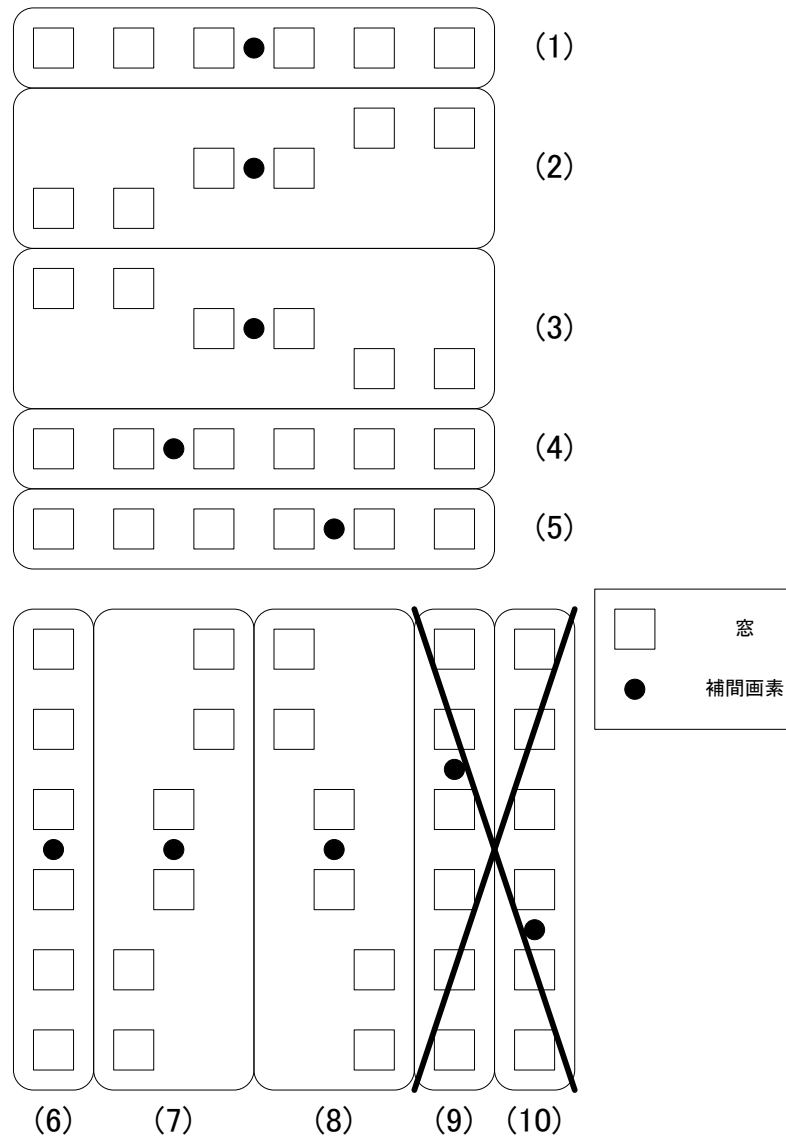


図 4.5: 可変窓による窓の例 .

仮定より，動きの大きい箇所では誤差が生じやすいということから，補間精度の向上を目的とし，ものの動きに応じて補間の窓を変化させる可変窓を提案する．提案する可変窓による補間を説明する．H.264/AVC による補間では図 4.5 の (1)，(6) による窓の補間しか行われない．徳永らの手法は (1)，(2)，(3) などを含む 7 種類の窓の補間をする．本提案手法では，横ずれや縦ずれを加え，ベクトルを 10 種類に分類し，それに応じて異なる窓を選択し，補間する．窓の選択はブロック毎にする．これは，動きベクトルの計算がブロック単位で行われることによる．すべての窓を図 4.5 の (1) ~ (10) に示した．この可変窓を用いた補間することにより，従来手法よりも誤差の少ない補間ができると考えられる．

## 各可変窓の選択率(mobile)

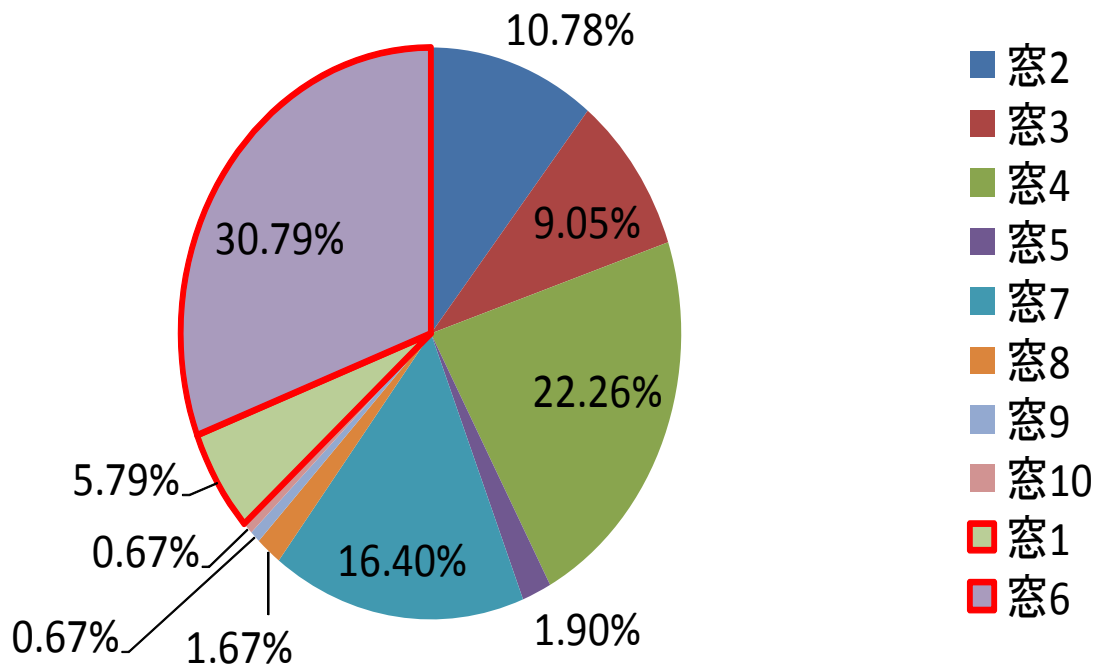


図 4.6: 各窓の選択率の例 .

だが、窓を増やしたことにより、計算量が増大し、遅延が発生する。この遅延を抑えるために、不必要な窓を削除する方法を考察する。不必要な窓を判断するために、各窓の選択率を調べた。その結果を図 4.6 に示す。これは mobile という動画でどの可変窓がどの程度、選択されたかを表したチャートである。窓 9、10 は他の窓と比較して、選択率が極端に低いことが分かる。したがって、この 2 つの窓を削除し、可変窓とする。



## 4.5 本章のまとめ

本章では，動きの大きい箇所では誤差が生じやすいという仮定を立て，予測精度向上を目的として，次世代動画像符号化に適した適応的な補間方法を提案した．動き情報を考慮した可変窓による補間をすることにより，精度の高い補間をすることができると考えられる．可変窓では，ものが斜めに動く場合や激しく動く場合に予測精度が上がると思われる．さらに，Wiener フィルタと併用することにより，安定した補間ができる．これにより，予測誤差の減少が期待できる．

次章では，提案した手法を実装し，実際に評価を行う．

# 第5章

## 実験・評価

## 5.1 本章の概要

本章では前章で提案した手法を Reference Software JM と Verilog-HDL に実装し，提案手法の評価をする．

5.2 節「ソフトウェアシミュレーション結果」では，ITU-T SG16 で決められた H.26L に関する推奨シミュレーション条件に則った実験結果を評価する．提案手法を Reference Software JM に実装し，提案手法と H.264/AVC と徳永らの手法と比較する．評価は RD 曲線を用いる．PSNR 向上とビット・レート削減が実現しているか確認する．

5.3 節「Verilog-HDL による評価結果」では，提案手法と H.264/AVC を Verilog-HDL に実装し，面積，遅延を評価する．5.2 節の利得結果を得て，ハードウェアにおいて，どの程度の面積，遅延オーバーヘッドが発生するかを確認する．

5.4 節「考察」では，実験から得られた RD 曲線や面積，遅延から考察をする．本提案手法がどのような場合に有効で，なぜ有効であるかを考える．

## 5.2 ソフトウェアシミュレーション結果

本章では，提案手法を H.264/AVC のエンコーダが C 言語で書かれた Reference Software JM に実装し，提案手法と H.264/AVC と徳永らの手法を比較する．比較するのは動画像でよく用いられている客観評価の指標である PSNR[dB] とビット・レート [kbps] である．一般的に PSNR が上がれば画質が上がると言われている．評価方法は縦軸を PSNR，横軸をビット・レートとした RD 曲線である．グラフが左上に行けばいくほど高画質，高圧縮が実現できていることになる．以下，目的，方法，結果の順に述べる．

### 5.2.1 目的

表 5.1: 3つのプログラムの特徴

	誤差の小さい方を選択	可変窓
anchor	×	×
tokunaga		×
proposed		

まず，実装内容を説明する．本実験では，3つのプログラムを比較する．3つのプログラムとは，anchor，tokunaga，proposed である．各プログラムの特徴を表 5.1 に示す．プログラムを順に説明する．

1つ目は JM 本体 (anchor) である．これは，HHI(ハインリヒ・ヘルツ研究所) が提供する H.264/AVC の標準ソフトウェアである．内容は，H.264/AVC のエンコーダ・デコーダが C 言語で実装されたものである．提案者はこの JM に提案する手法を実装し，その有用性を示す．バージョンは JM16.1 を使用している．

2つ目は tokunaga である．これは，3.4 節で説明した徳永らの手法である．動き情報に合わせて適応的に補間窓を変化させる．

3つ目は proposed である．これは，4 章で説明した提案手法である．動き情報に合わせて適応的に窓をさせ，不要と思われる窓を削除した．また，H.264/AVC の補間方法を並列にし，補間による誤差のムラを軽減させた．

3つのプログラムを比較したものを図 5.1 に示す．以降，これらのプログラムを用いて実験結果を比較をする．

次に，実験目的を示す．目的は RD 曲線による評価である．RD 曲線とは Rate-Distortion 曲線のことであり，レート-歪を示した曲線である．縦軸は PSNR[dB] で，横軸はビット・レートである．グラフの曲線が左上になればなるほど，高画質，高圧縮が実現できているという

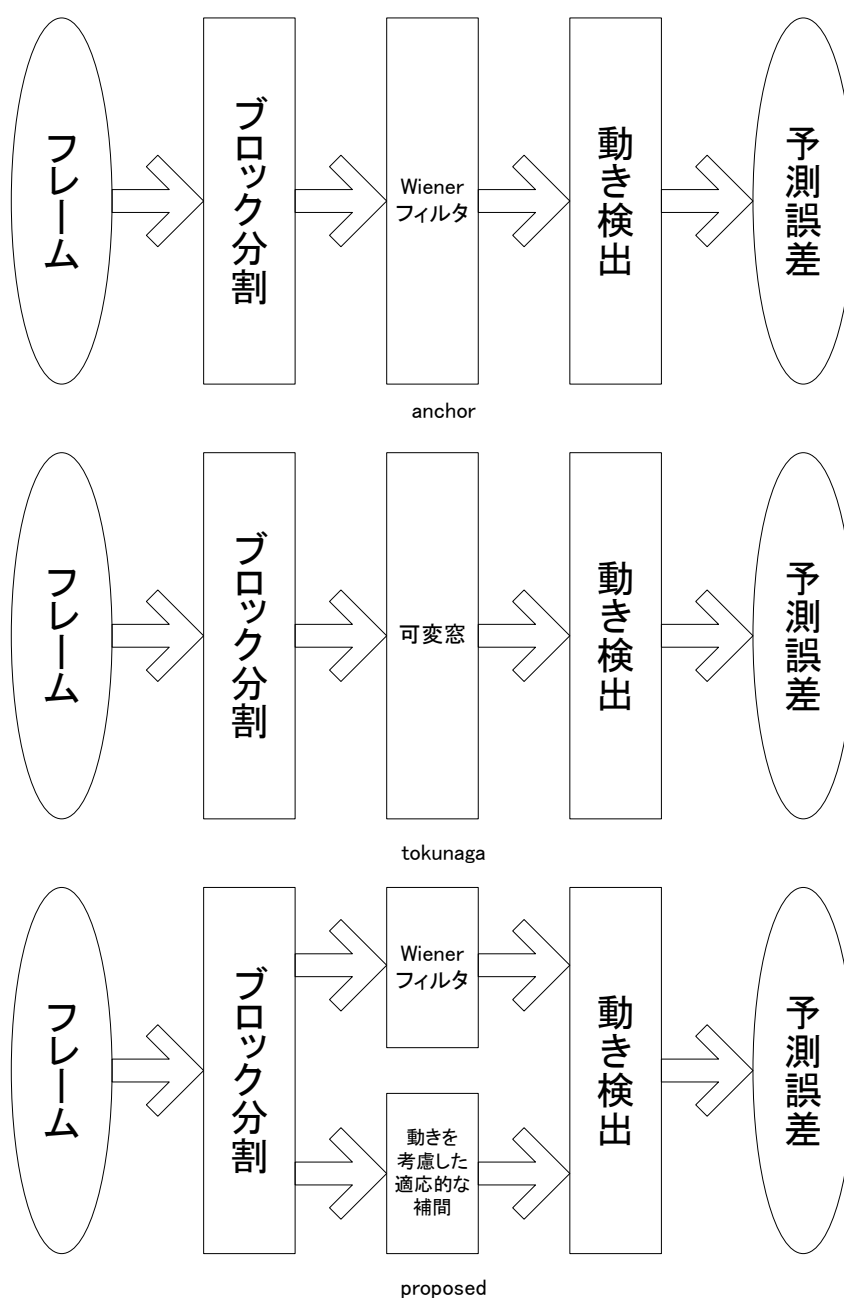


図 5.1: anchor と tokunaga と proposed の比較の例 .

ことになる．説明した3つのプログラムで特徴の異なる7種類の動画を圧縮し，結果から得られた各 RD 曲線において，すべての動画で anchor と tokunaga よりも PSNR-Y の向上およびビット・レートの低下していることを確認することが目的である．

次節より，目的通りの結果が得られているかを確認する．

### 5.2.2 方法

実験方法を示す．ITU-T SG16 で決められた H.26L に関する推奨シミュレーション条件に則り実験する．

表 5.2: 共通実験で定められているテスト・シーケンス

フレーム・スキップ	2	1	0
シーケンス	Container (QCIF:300 フレーム)	Paris (CIF:300 フレーム)	Mobile (CIF:300 フレーム)
	News (QCIF:300 フレーム)	Silent Voice (QCIF:300 フレーム)	Tempete (CIF:300 フレーム)
	Foreman (QCIF:300 フレーム)		

次に，ITU-T SG16 で決められた H.26L に関する推奨シミュレーション条件を示す．表 5.2 に示されたフレームスキップですべてのテスト・シーケンスに対し，それぞれ以下の 4 種類のパラメータセットを適用し，エンコードする．

$$QPI = QPP = 28; QPB = 30$$

$$QPI = QPP = 32; QPB = 34$$

$$QPI = QPP = 36; QPB = 38$$

$$QPI = QPP = 40; QPB = 42$$

これらのパラメータは I ピクチャ，P ピクチャ，B ピクチャそれぞれのピクチャに対する量子化パラメータである．この実験をそれぞれ anchor，tokunaga，proposed に対して行う．エンコードしたシーケンス情報から，PSNR-Y とビット・レートの RD 曲線を作成する．この実験をそれぞれ anchor，tokunaga，proposed に対して行う．評価はこれらの RD 曲線を比較して行う．

### 5.2.3 結果

前節で示した実験方法で得られた結果を示す．結果は RD 曲線で表す．すべての動画において，H.264/AVC と徳永らの手法より，グラフが左上にきていることが確認できる．これは，H.264/AVC と徳永らの手法より高画質，高圧縮が実現していることを意味する．

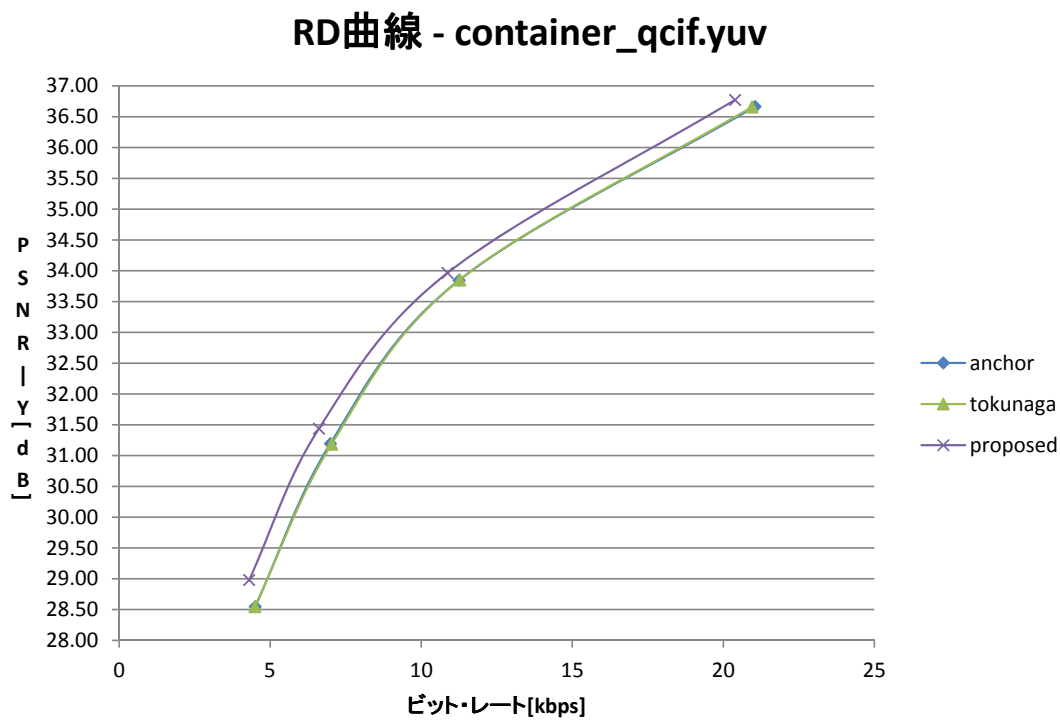


図 5.2: container の RD 曲線 .

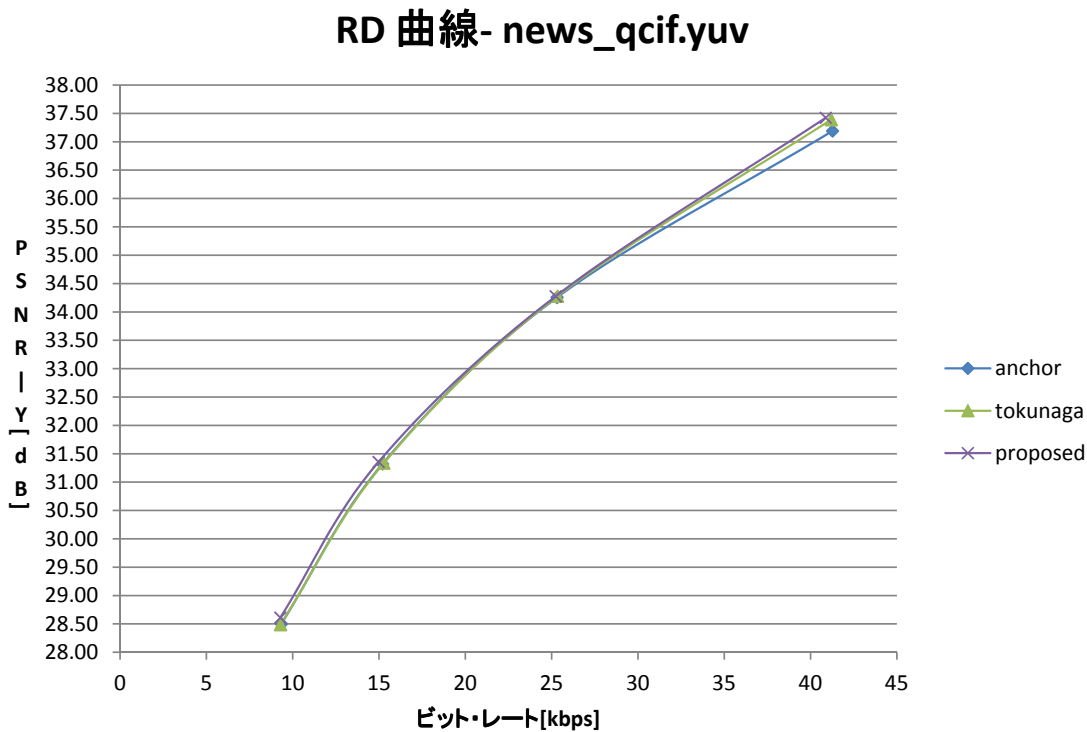


図 5.3: news の RD 曲線 .



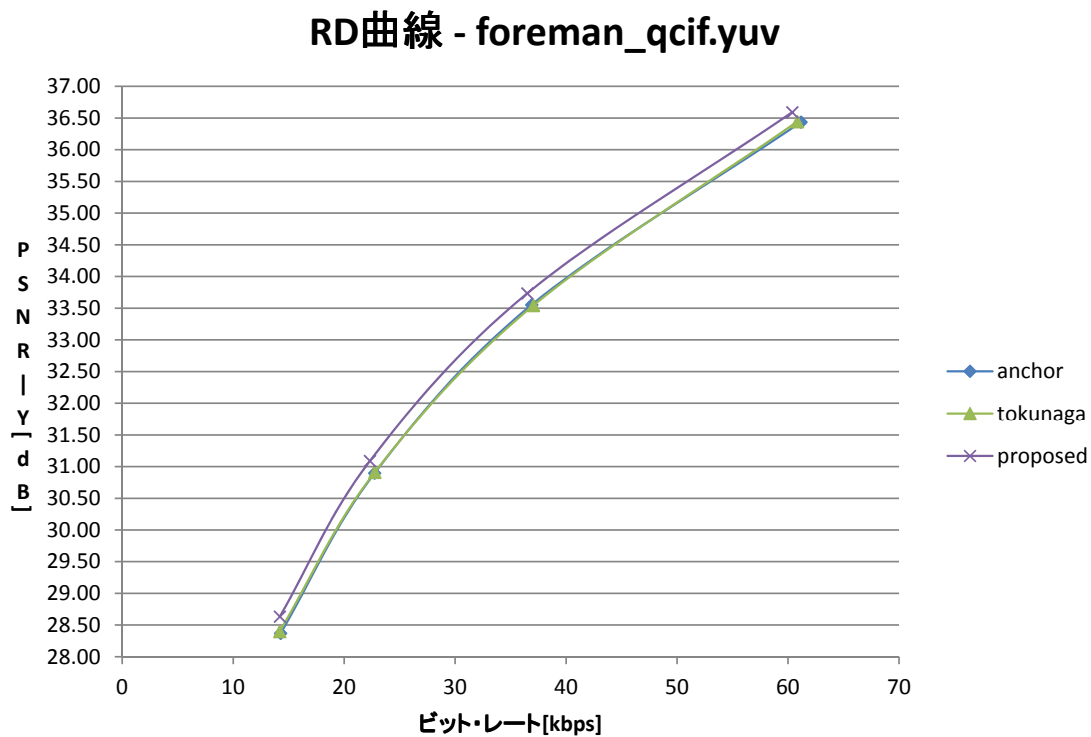


図 5.4: foreman の RD 曲線 .

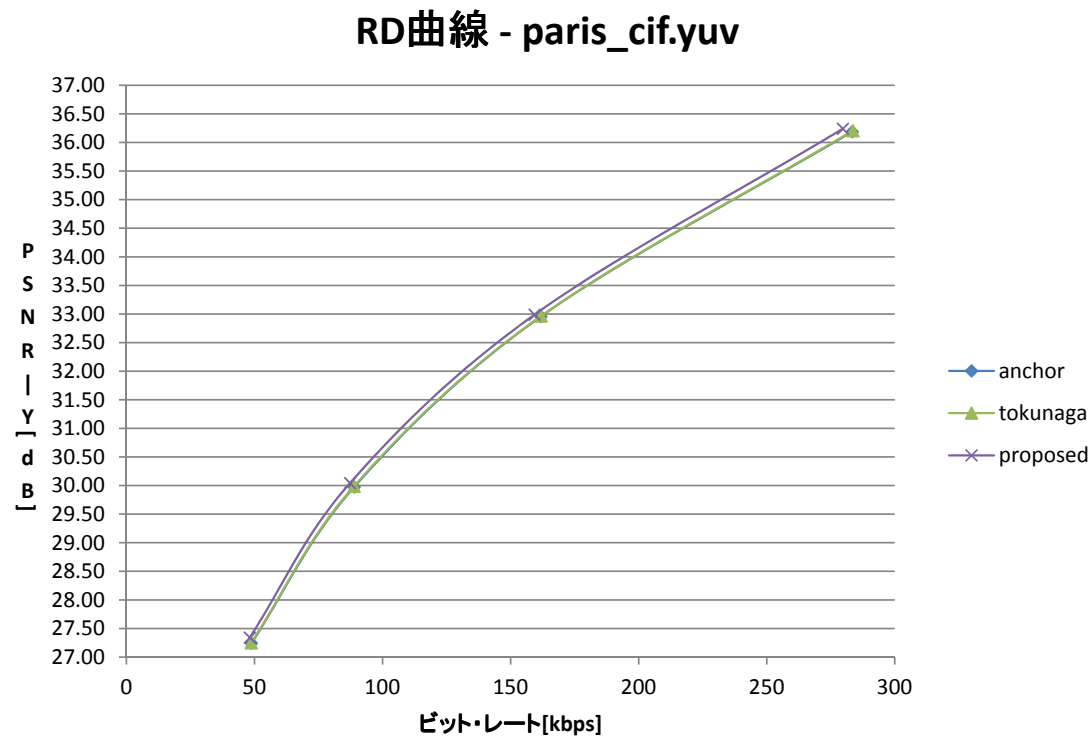


図 5.5: paris の RD 曲線 .

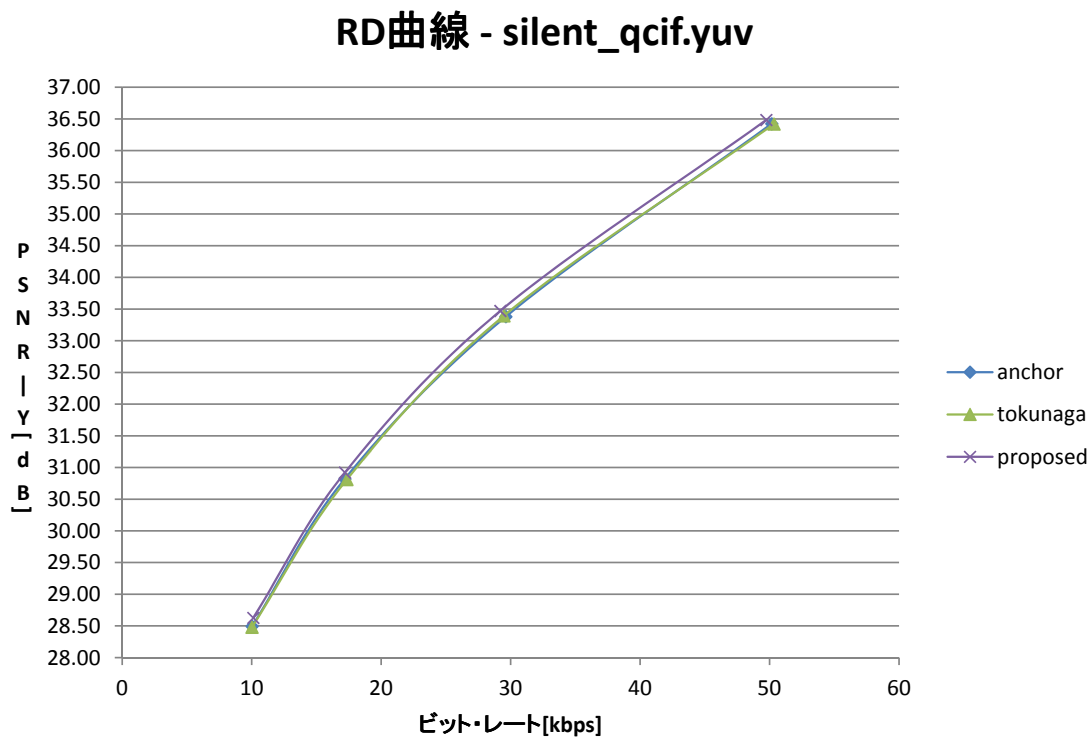


図 5.6: silent の RD 曲線 .

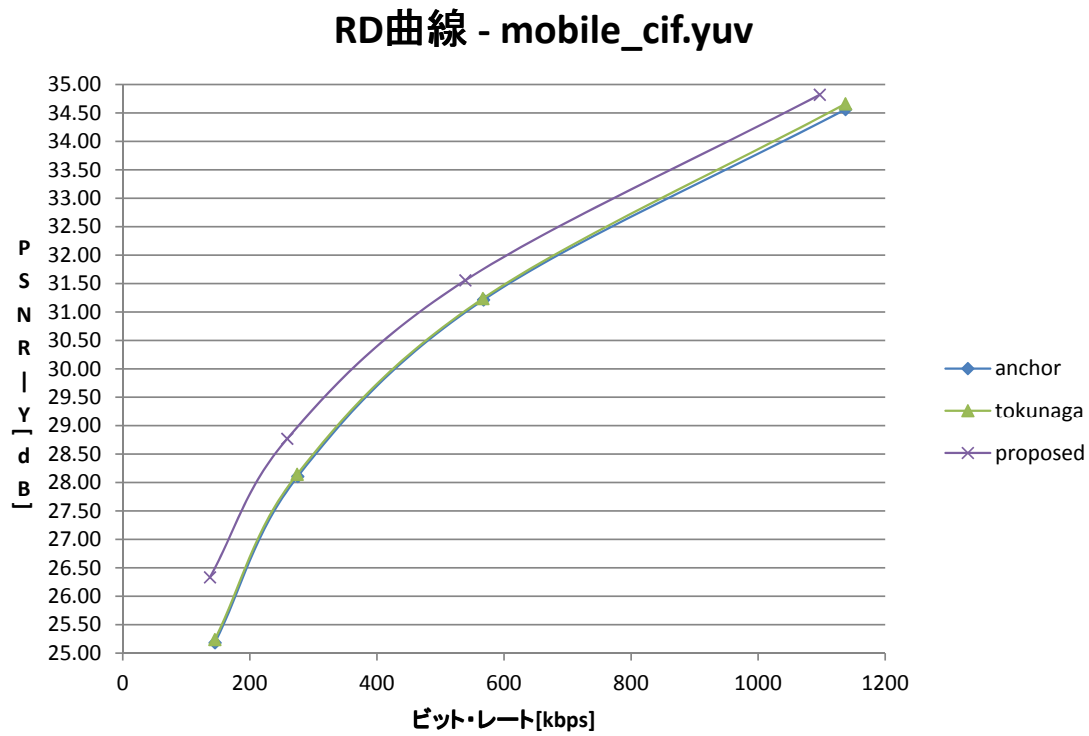


図 5.7: mobile の RD 曲線 .

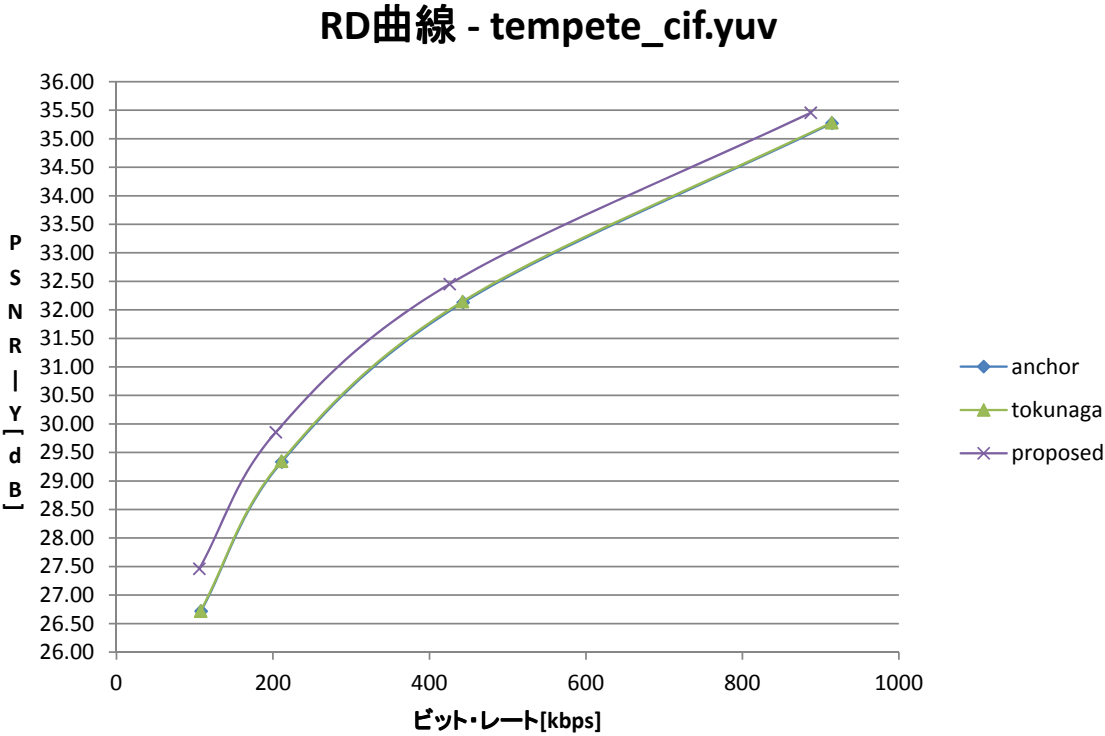


図 5.8: tempete の RD 曲線 .

## 5.3 Verilog-HDL による評価結果

本章では，H.264/AVC と提案手法を Verilog-HDL で実装し，比較する．比較するのは面積，遅延である．提案手法の並列補間による面積，遅延オーバーヘッドがどの程度になるのかを確かめる．以下，目的，方法，結果の順に述べる．

### 5.3.1 目的

まず，実装内容を説明する．本実験では，2 つのプログラムを比較する．2 つのプログラムとは，`anchor`，`proposed` である．内容は前節で説明した同名プログラムの補間部のみを Verilog-HDL で実装したものである．ここでの補間部とは，輝度補間のみでなく，色差補間も含むこととする．提案手法の補間部は，単純に並列させただけでなく，計算機やレジスタを共有可能な箇所は共有し，遅延がないよう設計している．

次に，実験目的を示す．実験目的は，提案手法による面積，遅延オーバーヘッドはどの程度かを確かめることである．面積は単純に 2 倍になっていないことと，遅延が発生していないことを確かめる．

次節より，目的通りの結果が得られているかを確認する．

表 5.3: 輝度補間器のみの論理合成結果 .

	面積 [ $\mu m^2$ ]	遅延 [ns]
anchor	7357.8	1.13
proposed	11687.382	1.13

表 5.4: 補間器の論理合成結果 .

	面積 [ $\mu m^2$ ]	遅延 [ns]
anchor	27596.3	1.6
proposed	31925.9	1.6

### 5.3.2 方法

実験方法を示す．実験方法は，実装した Verilog-HDL のファイルに対し，Synopsys 社の Design Compiler Z-2007.03-SP4 のトポグラフィカルモードを用いて論理合成をする．ここから得られた面積，遅延を比較，評価する．

### 5.3.3 結果

前節で示した実験方法で得られた結果を示す．表 5.3 に輝度補間器のみ，表 5.4 に補間器の論理合成結果を表す．面積は輝度補間器のみで 58.8%，補間器で 15.7%増加するものの 2 倍より小さく，遅延オーバーヘッドもなかった．

## 5.4 考察

得られた結果を整理し，考察する．

ソフトウェアシミュレーションの実験の結果は，すべてのシーケンスにおいて，anchor と tokunaga よりも PSNR-Y の向上，ビット・レートの低下が見られた．

Verilog-HDL による実験の結果は，anchor と比較して，面積は2倍より小さく，遅延オーバーヘッドはなかった．

以上，目的通りの結果が得られた．したがって，提案手法による補間で予測精度が向上したと考えられる．

次に，実験結果から考察をする．ソフトウェアシミュレーションによる実験結果では，シーケンス mobile，tempete では特に PSNR-Y の向上，ビット・レートの削減が見られた．これらのシーケンスに共通していることは，画面全体が動くことである．他のシーケンスでは背景が固定されており，画面の一部が動く．したがって，動きが激しい場合において，proposed が有効であることが分かった．

tokunaga は anchor と比較して，大きく精度向上が見られる場合もあれば，あまり効果が期待できない場合もある．やはり，シーケンスによって精度が不安定であると言える．

結論として，proposed では動きの少ないフレームでは主に 6-tap Wiener フィルタによる安定した補間が採用され，動きの激しいフレームでは主に可変窓フィルタによる動きを考慮した補間が採用されていると思われる．このことから，お互いの弱点を補強し合う補間をしていると考えられる．

Verilog-HDL による実験結果では，輝度補間器のみで考えると面積増加が 58.8%，遅延オーバーヘッドはなし，色差補間を含む補間器で考えると面積増加が 15.7%，遅延オーバーヘッドはなしという結果であった．輝度補間器において，輝度値の内部バッファと  $1/4$  精度の補間器を共有し， $1/2$  精度の補間器は独立させて設計した．したがって，面積は2倍にならずに済んだと考えられる．また，並列させたことにより，ソフトウェアではエンコード時間が増加していたが，ハードウェアではペナルティは少なくて済む．

ペナルティについてもう少し考察する．前述した補間器は，補償部を考慮していない．したがって，補償部によるスループット低下と面積増加のペナルティが発生する．補償部において，このスループット低下と面積増加が補償部のトレードオフとなる．補償は 8bit 減算器を並列して設計すると考えられる．8bit 減算器 1 つにかかる面積は，5.3.2 節と同様の方法で実験した場合， $50.1 \mu m^2$  掛かると見積もれる．これを並列すればするほど，スループットの低下を軽減することができる．だが，面積は増加する．

以上より，本提案手法で現行の画質を保ちつつ，動きの激しい動画の画質向上が可能であることがわかった．

## 5.5 本章のまとめ

本章では4章で提案した手法を実装し、実験した。ソフトウェアシミュレーションによる実験では、大きな動きが発生するシーケンスに対して、提案手法が有効であることが分かった。Verilog-HDLによる実験では、面積は単純に2倍とはならないことと遅延オーバーヘッドは発生しないことが分かった。目的で示した通りの実験結果が得られた。考察から、仮定を立てた通り、斜めに動くシーケンスや動きの激しいシーケンスに対して本提案手法が有効であることもわかった。ただし、補償部においてスループットと面積のペナルティが発生すると考えられる。これらのトレードオフを考慮し、設計することが重要であると考えられる。

## 第6章

### 結論



本論文では、次世代動画像符号化に適した並列補間手法の提案を行った。

第2章「H.264/AVC」では、H.264/AVCでの画面予測及び符号化を説明した。ブロック分割では、マクロブロック及びマクロブロックの分割方法を述べた。色の変化の平坦な箇所には大きなマクロブロックを、色の変化の激しい箇所には小さなマクロブロックを割り当てるのが適当であることを述べた。intra 予測では、符号化対象ブロックの左上、上、右上、左のブロックの画素を用いて、9つのモードから画素を予測する方法を示した。inter 予測では、1枚または2枚のフレームを用いて画面を予測する方法を示した。符号化では、指数ゴロム符号、CAVLC、CABACの3種類の符号化を説明した。H.265に向けてさらなる予測精度向上、それに適した符号化が必要であることを述べた。

第3章「補間」では、H.264/AVCで用いられている輝度信号の補間方法と徳永らの手法を説明した。H.264/AVCでは、整数画素及び1/2精度の画素を用いて6-tap Wiener フィルタで1/2精度の補間、整数画素、1/2精度の画素、1/4精度の画素を用いてBi-linear フィルタで1/4精度の補間を行っている。徳永らの手法では、1/2精度の補間に着目し、動きに合わせて補間窓を変化させる。だが、これらの方法では、どんな場合でも、同様の方法で補間をしたり、フレームによっては誤差が大きくなってしまうという問題点があった。

第4章「次世代動画像符号化に適した並列補間手法」では、3章で述べた問題点を基に動き情報を加味した適応的な並列補間方法を提案した。H.264/AVCの6-tap Wiener フィルタの補間に加え、動き検出によって得られるものの動き情報を加味した補間を行い、誤差の少ない方を採用する。動きベクトルが大きいものは誤差が発生しやすいという仮定より、可変窓を用いた並列補間方法を提案した。

第5章「実験・評価」では、4章で述べた提案手法をReference Software JMとVerilog-HDLに実装し、評価した。JMでは、6-tap Wiener フィルタと提案手法のフィルタの採用率と、どの程度のPSNR-Yとビット・レートの利得が得られるかを比較し、確かめた。プログラムは、JM本体(anchor)、徳永らの手法(tokunaga)、提案手法(parallel+vector)の2種類を用いた。PSNR-YはJM本体と比較して、最大で1.15[dB]、徳永らの手法と比較して、1.17[dB]の向上が見られた。ビット・レートはJM本体と比較して、最大で5.5%、徳永らの手法と比較して、5.8%の削減が見られた。目的通り、PSNR-Yの向上とビット・レートの低下が確認できた。Verilog-HDLでは、H.264/AVCの補間器と提案手法の遅延・面積評価をした。面積は15.7%増加、スループットは微小に低下するものの、遅延オーバーヘッドは変化しないことが分かった。

以上、次世代符号化に適した適応的な補間方法により、予測誤差を小さくすることが可能であることが分かった。PSNR-Yの向上、ビット・レートの低下、面積、遅延が確認できた。課題点はメモリを考慮して、動作させるように設計することである。Verilog-HDL実装では、輝度値が順番に入力されるという仮定をしているので、フレームメモリに対してロード命令

ができない．輝度，色差信号などを保持するメモリを仮定して，必要なアドレスを計算し，ロードできると良い．

# 謝辞

本論文全般にわたり，御指導ならびに御助言を授かった戸川望教授，大附辰夫教授，柳澤政生教授に深く感謝いたします．

また，終始，適切なご指導および御助言を頂きました戸川研究室助教の史又華氏，柳澤研究室助教の廣津良氏に深く感謝いたします．

最後に，本論文に関する研究活動全般にわたり支援していただいた戸川研究室，大附研究室および柳澤研究室の皆様に感謝いたします．

## 参考文献

- [1] D. Rusanovskyy, K. Ugur, J. Lainema, “Adaptive Interpolation with Directional Filters,” ITU-T SGI 6/Q.6 Doc. VCEG-AG21, Shenzhen, China, October 2007.
- [2] 榎本忠儀, 画像 *LSI* システム設計技術, コロナ社, 2005.
- [3] I. E. G. Richardson, *The H.264 Advanced Video Compression Standard*, John Wiley & Sons Ltd, 2010.
- [4] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Wiley & Sons Ltd, 2003.
- [5] ITU-T, “Advanced video coding for generic audiovisual services,” H.264 Standard, March 2009.
- [6] Joint Video Team, *Reference Software JM16.1*, <http://iphome.hhi.de/suehring/tml/download/>, 2009.
- [7] K. M. Cann, W. J. Han, I. K. Kim, “Samsung’s response to the Call for Proposals on video compression technology,” ITU-T SG16 Doc. JCTVC-A124, Dresden, DE, April 2010.
- [8] 角野眞也, 菊池義浩, 鈴木輝彦, *H.264/AVC 教科書*, 株式会社インプレス R&D, 2009.
- [9] R. Wang, M. L. J. Li, Y. Zhang, “High throughput and low memory access sub-pixel interpolation architecture for H.264/AVC HDTV decoder,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 3, pp. 1006–1013, Aug. 2005.
- [10] S. Wittmann, T. Wedi, “Separable Adaptive Interpolation Filter,” ITU-T SGI 16/Q.6 Doc. T05-SG16-C-0219, Geneva, Switzerland, June 2007.
- [11] T. Wedi, “Adaptive interpolation filters and high-resolution displacements for video coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 4, pp. 484–491, Apr. 2006.
- [12] T. Wedi, “Adaptive interpolation filters for motion compensated prediction,” in *Proceedings of 2002 IEEE International Conference Image Processing*, vol. 2, pp. 509–512, Sept. 2002.

- [13] T. Wedi, “Adaptive interpolation filters for motion compensated hybrid video coding,” in *Proceedings of Institut fuer Theoretische Nachrichtentechnik und Informationsverarbeitung Picture Coding Symposium*, pp. 49–52, Apr. 2001.
- [14] 徳永佳晃, 長谷山美紀, 北島秀夫, “ブロックの動きと画像内容を考慮した付加ビットを必要としない適応内挿フィルタ選択法,” 電子情報通信学会技術研究報告, vol.J88-D2, no.7, pp. 1313–1317, July 2005.
- [15] Y. L. S. Lin, C. Y. Kao, H. C. Kuo, J. W. Chen, *VLSI Design for Video Coding*, Springer Verlag, 2010.
- [16] Y. Vatis, B. Edler, D. T. Nguyuen, J. Ostermann, “Two-dimensional non-separable Adaptive Wiener Interpolation Filter for H.264/AVC,” ITU-T SGI 6/Q.6 Doc. VCEG-Z17, Busan, South Korea, April 2005.

# 本論文に関する発表業績

国内学会（査読なし）

1. 栗岡大生，柳澤政生，戸川望，“動きベクトルを考慮した遅延オーバーヘッドのないハードウェア向き適応的並列補間手法,” 電子情報通信学会大会講演論文集 (G0508A), pp. 75, 2011.