

修士論文

題目

H.265/HEVCにおけるインター  
予測モード選択処理低減法の研究

指導教員

近藤 利夫 教授

2016年度

三重大学大学院 工学研究科 情報工学専攻  
コンピュータアーキテクチャ研究室

西村 隆太郎 (415M513)

## 内容梗概

2018年にスーパーハイビジョンの本放送開始が予定されるなど、動画の高精細化が進んでいる。この高精細化に伴うデータ量増大により、更なる圧縮率の向上が求められている。その要求に応えるために、JCT-VC(Joint Collaborative Team on Video Coding)により、従来規格であるH.264/AVCと比較して画質を維持しつつ圧縮率を2倍にまで高めたH.265/HEVCが2013年に標準化された。このH.265/HEVCにおける圧縮率向上の中核技術の一つに、CTU(Coding Tree Unit)による四分木構造及びCU(Coding Unit), PU(Prediction Unit), TU(Transform Unit)の導入によって実現した適応的なブロック分割があげられる。この分割方法では、画像の様子が一様な領域は大きなブロックが、画像の様子が複雑な領域には小さなブロックが、それぞれ割り当てられるようにして、圧縮率の向上が図られている。しかしこの分割方法を決定するためには、膨大な演算が必要な動き探索処理を含むインター予測モードの適用可否判定処理を繰り返さなければならないため、H.265/HEVCの符号化処理高速化のボトルネックとなっている。そのため、エンコード時間削減には不必要なインター予測候補PUの削減が効果的であると考えられる。

そこで本論文では、CUを構成する $2N \times 2N$ 画素の分散や、CUを4分割した $N \times N$ ブロックそれぞれの分散値の分散、各PUを構成する画素の分散や分散値の分散が閾値を超えたか否かの輝度値分散特徴判定により、 $2N \times 2N$ サイズ以外の候補PUを大幅に削減する手法を提案した。この提案手法をHM-10.0に実装し、エンコード時間を50%以上低減できることを示した。またフレーム別のエンコード時間削減率についても評価した結果、全てのフレームで高いエンコード時間削減率を維持できていることから、提案手法は低遅延用途にも有効であることを示した。

# Abstract

In recent years, video image definition has been greatly enhanced such as 8K Ultra HDTV. Since such enhancement causes explosive data increase, high efficiency video coding (HEVC) is required. In this situation, H.265/HEVC which can double the data compression rate compared with H.264/AVC was developed by JCT-VC(Joint Collaborative Team on Video Coding) in 2013. To achieve high compression efficiency, H.265/HEVC adopts a flexible block structure which includes CTU(Coding Tree Unit), CU(Coding Unit), PU(Prediction Unit) and TU(Transform Unit). This technique provides adaptive area partitioning corresponding to the movement and the texture of subjects. Large blocks are selected in flat texture region and small blocks are selected in complex texture region. However this partition mode selection causes huge amount of computational complexity, because this process repeats inter prediction which contains motion estimation and motion compensation to determine the best partition from many candidates. Therefore, unnecessary PU candidate exclusion reducing the motion estimation process is effective for the computational complexity reduction and encoding time reduction.

In this paper, we proposed a fast PU mode selection method utilizing variance characteristic of luminance samples. This proposed method utilizes the  $2N \times 2N$  variance of luminance samples in a CU, the variance of four variances of luminance samples in each different  $N \times N$  blocks in the CU and the variance and the variance of variances of luminance samples in a PU. We implemented the proposed method to HM-10.0, and evaluated its effectiveness. The evaluation result showed that the proposed method was able to reduce encoding time by more than 50%. In addition, we showed that the proposed method is effective for low delay encoding.

# 目次

1	まえがき	1
2	符号化処理と関連研究	2
2.1	符号化処理の概要	2
2.2	H.265/HEVCにおける符号化処理	2
2.3	関連研究	4
3	符号化特性調査	7
4	提案手法	13
4.1	対象 CU 内輝度値分散特徴を利用した PU 削減法	13
4.1.1	正方形 PU への限定手法	13
4.1.2	対称分割 PU の候補削減手法	14
4.1.3	非対称 PU の候補削減手法	15
4.1.4	提案手法の流れ	15
4.2	CU 内輝度値に特徴がない場合の PU 削減法	16
5	評価	17
5.1	実装方法	17
5.2	評価条件	17
5.3	評価結果	19
5.4	分散の計算時間	26
5.5	低遅延および蓄積用途における有効性	29
5.6	考察	36
6	あとがき	37
	謝辞	37
	参考文献	37

## 目 次

2.1	インター予測処理 . . . . .	2
2.2	インター予測 PU の種類 . . . . .	3
2.3	ブロック分割 . . . . .	4
2.4	ダイヤモンド探索 . . . . .	5
3.5	$2N \times 2N$ 画素の分散値別の $2N \times 2N$ 選択率 1 . . . . .	8
3.6	$2N \times 2N$ 画素の分散値別の $2N \times 2N$ 選択率 2 . . . . .	9
3.7	$N \times N$ ブロック各分散値の分散に対する $2N \times 2N$ 選択率 1 . . . . .	10
3.8	$N \times N$ ブロック各分散値の分散に対する $2N \times 2N$ 選択率 2 . . . . .	11
4.9	$N \times N$ 分散値の分散 . . . . .	13
4.10	$N \times 2N$ における計算方法 . . . . .	14
4.11	$nL \times 2N$ における計算方法 . . . . .	15
5.12	実装方法 . . . . .	17
5.13	フレーム別エンコード時間 (Random Access) . . . . .	32
5.14	フレーム別エンコード時間削減率 (Random Access) . . . . .	33
5.15	フレーム別エンコード時間 (Low Delay) . . . . .	33
5.16	フレーム別エンコード時間削減率 (Low Delay) . . . . .	34

## 表 目 次

5.1	テストシーケンス . . . . .	18
5.2	符号化条件 . . . . .	18
5.3	実行環境 . . . . .	19
5.4	Random Access における評価結果 1 . . . . .	20
5.5	Random Access における評価結果 2 . . . . .	21
5.6	Low Delay における評価結果 1 . . . . .	22
5.7	Low Delay における評価結果 2 . . . . .	23
5.8	関連研究との比較 (Random Access) . . . . .	25
5.9	関連研究との比較 (Low Delay) . . . . .	25
5.10	Random Access における分散の計算時間 . . . . .	27
5.11	Low Delay における分散の計算時間 . . . . .	28
5.12	ECU の性能 (Random Access) . . . . .	30
5.13	ECU の性能 (Low Delay) . . . . .	31

# 1 まえがき

2018年にスーパーハイビジョンの本放送開始が予定されるなど、動画像の高精細化が進んでいる。この高精細化に伴い、動画像のデータ量が大幅に増大し、更なる圧縮率の向上が求められている。その要求に応えるために、ITU-T VCEGとISO/IEC MPEGによって組織されたJCT-VC(Joint Collaborative Team on Video Coding)により、画質を維持しつつ圧縮率を現在広く利用されているMPEG-2の4倍、H.264/AVCの2倍にまで高めたH.265/HEVC[1]が2013年に標準化された。このH.265/HEVCに新たに組み込まれた圧縮手法の中で、CTUの適応的なブロック分割は、圧縮率向上の要の技術である。CTUはH.265/HEVCにおける各ピクチャの正方形の分割単位であり、標準で64x64画素ブロックから8x8画素ブロックまで再帰的に四分木分割可能な四分木構造を採る。この木構造の各節であるCU(Coding Unit)は更にインター予測やイントラ予測を行う際に、PU(Prediction Unit)に分割される。このPUは、 $2N \times 2N$ 、 $N \times 2N$ 、 $2N \times N$ 、 $2N \times nU$ 、 $2N \times nD$ 、 $nL \times 2N$ 、 $nR \times 2N$ と、非対称分割(AMP: Asymmetric Motion Partition)も入り、H.264/AVCの可変ブロックサイズに比べ大幅に種類を増やしている。このCTUの適応的な分割方法では、画像の模様や動きが一樣な領域は大きなブロックが、画像の模様や動きが複雑な領域には小さなブロックが、それぞれ割り当てられるようにすることで、圧縮率の向上が図られている。しかし、この圧縮率の向上のための最適なCUやPUの決定に膨大な処理が必要となり、H.265/HEVCの符号化処理高速化のボトルネックとなっている。そこで本論文では、この適応的なCTU分割の処理量増大の主因が、各CUにおける最適なPU分割モードとその予測方式を見つけ出すために、インター予測の適用可否判定処理を全ての候補PUに対して繰り返すことにある点に着目し、候補PUの種類を大幅に制限する手法を提案する。具体的には対象CU内輝度値分散特徴を利用し $2N \times 2N$ 以外の候補PUの低減手法を提案し、従来手法と比較してエンコード時間を50%以上低減できることを示す。またフレーム別のエンコード時間削減率についても評価し、低遅延用途における有効性も示す。

## 2 符号化処理と関連研究

### 2.1 符号化処理の概要

動画画像符号化処理は大きく分けてインター予測(動き補償), DCT(Discrete Cosine Transform) あるいは DST(Discrete Sine Transform), 量子化, エントロピー符号化で構成されており, 図 2.1 の対象フレームのように分割されたブロック単位で処理を行う. インター予測処理では符号化対象フレーム内の対象ブロックと最も類似度が高い領域を, 参照フレームの探索範囲内から画素の差分絶対値和である SAD(Sum of Absolute Differences) を用いて探索する. この探索の結果見つかった最類似ブロックと対象ブロック間での画素の差分である残差信号と, 最類似ブロックへの相対位置を表す動きベクトルを, DCT, 量子化プロセスを経て符号化する.

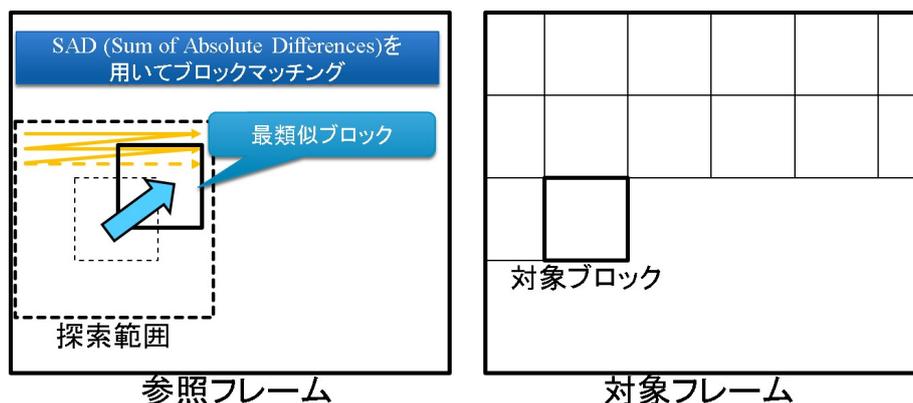


図 2.1: インター予測処理

インター予測処理は高い圧縮率を実現可能であるものの, 膨大な SAD 演算を必要とする動き探索処理が含まれているため, 処理時間が増大してしまう. H.265/HEVC のテストモデルである HM においては, インター予測が全体の約 7 割を占めていることが報告されている [2].

### 2.2 H.265/HEVC における符号化処理

H.265/HEVC では, 従来規格ではマクロブロックと呼ばれた符号化ブロックのサイズを, 標準で 64x64 画素から 8x8 画素まで分割可能とし,

新たに CU として定義している．またこの CU を節とする四分木構造が CTU であり，CTU を分割することで可変のブロックサイズを実現している．CTU は標準で深さ 0 から 3 まで分割可能であり，各節の CU サイズは  $2N \times 2N$  と表記される．つまり  $2N \times 2N$  画素ブロックの CU は  $N=4, 8, 16, 32$  の 4 サイズを取りうる．更に各 CU 内で取りうる PU の種類も，図 2.2 で示すように正方形と非対称を含む様々な長方形の分割が設けられており，画像に応じて柔軟な分割方法を選択できる．しかしながら H.265/HEVC では最適な CU，PU を選択するために，図 2.3 に示すように四分木構造の全ての枝分かれに属する全ての節の CU で，インター予測だけでも図 2.2 に示す多様な PU 候補を用いて 2.1 章で述べたように膨大な SAD 演算を必要とする動き探索処理と動き補償処理を含むインター予測を試行している．

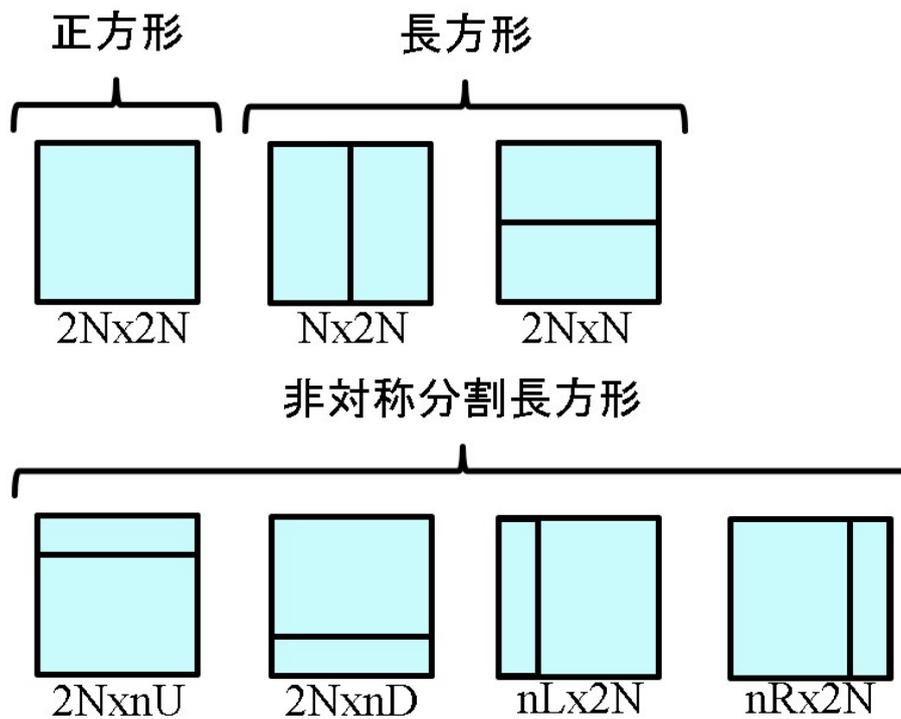


図 2.2: インター予測 PU の種類

また，多様なインター予測モードに加えてイントラ予測も予測モードとして設けられており，それぞれの予測モードや PU での残差信号を用いて

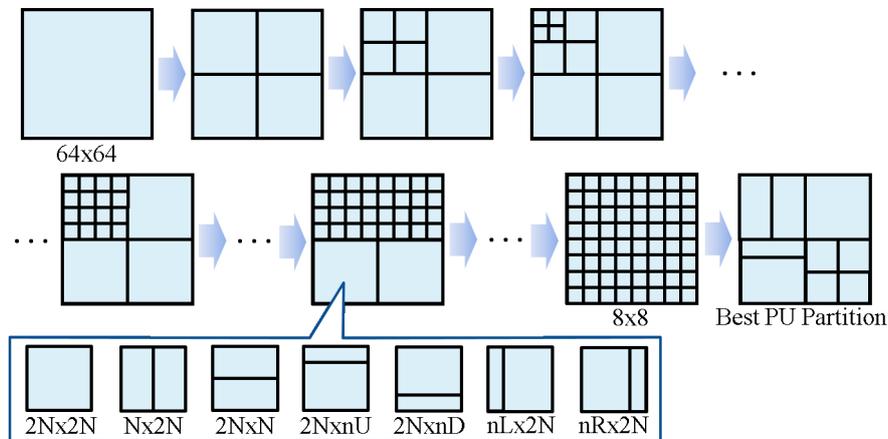


図 2.3: ブロック分割

式(1)により RD(Rate Distortion) コストを計算し, RD コスト最小となる予測モードと PU を選択する. なお式(1)の  $D$  は SAD あるいは SATD (Sum of the Absolute Transformed Differences),  $\lambda$  はラグランジュ乗数,  $B$  は予測モード情報などの符号量を表す.

$$J = D + \lambda \times B \quad (1)$$

このような全ての候補を試し, その結果に基づいて最適な分割方法の一つを選択する手法は, 演算量が膨大な動き検出処理を繰り返すこととなるため, 高速化の障害となっている.

## 2.3 関連研究

著者らはインター予測に含まれる動き探索処理についての改良を行ってきた. 具体的には整数画素精度の動き探索について, 図 2.4 のように探索中心から 64 画素離れた点まで 2 のべき乗の距離の点を探索する拡大ダイヤモンド探索と, 近傍 4 点を探索するスモールダイヤモンド探索 (図 2.4 の距離 1 の点) を周期的に切りかえる探索アルゴリズムを主軸に, 輝度分散値を用いた拡大制御やブロックサイズによる周期の設定などを組み込んだ手法を発表した [3].

この結果 BD-rate の悪化を最小限に抑えつつ, HM に組み込まれている探索アルゴリズムである TZSearch と比較して, AD (Absolute Difference)

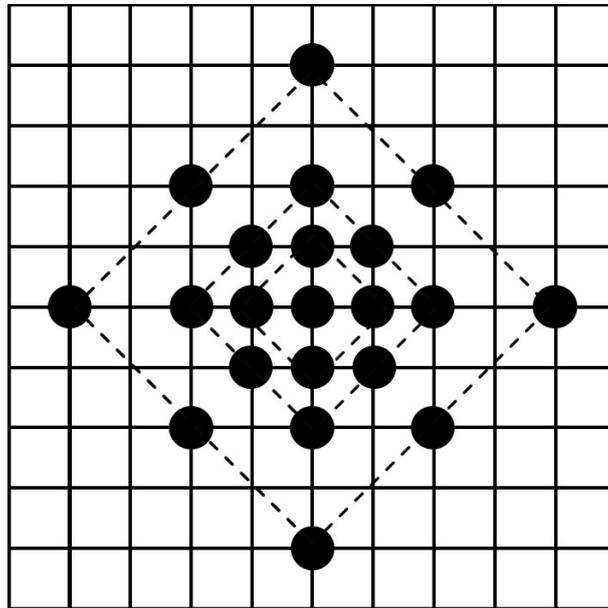


図 2.4: ダイヤモンド探索

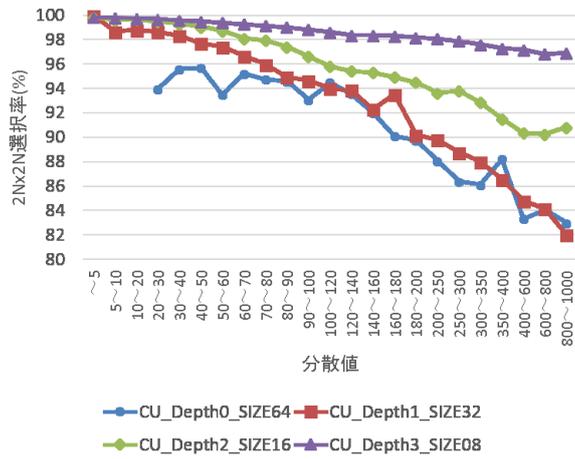
演算を約 80%削減した。しかしながら、最も計算時間が必要な処理の一つであるこの整数画素精度動き探索処理の演算量を大幅に削減しても、符号化全体の処理時間削減への効果は大きくない。よって更なる高速化のためには、インター予測そのものの試行を省略して、整数画素精度の動き探索に加えて小数画素精度の動き探索、RD コスト計算などの処理を削減することが効果的であると考えられる。

このインター予測、あるいはイントラ予測の試行省略可能である手法も数多く提案されている [4][5][6][7][8][9]。実行時間削減の効果が大きいものとして、CTU 枝切り手法や、各 CU での候補 PU の削減手法が挙げられる。その中で単純な方法ながら大幅な実行時間削減を実現しているものとして、CU 内輝度値の分散を用いた CTU 枝切り手法 [4][5][6] がある。この手法では処理中の CU の情報のみを参照するため、近傍の符号化済みブロックの情報を参照する手法と比べて、高並列化が実現しやすい。しかしインター予測有効の符号化条件においては、BD-rate の悪化が約 7%と大きい。また画像の輝度値の分散をみるのではなく、処理中 CU の最良モードに応じて枝切り可否の判定を行う手法が ECU オプションとして標準の HM に組み込まれている。これを拡張し、インター予測に用いられる候補 PU を  $2N \times 2N$  のみに限定する手法 [7] も提案されており、BD-rate の

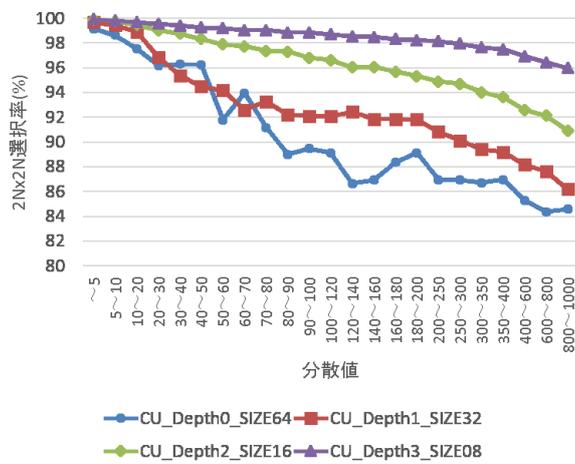
悪化を最小限に抑えつつ 20%以上の実行時間削減を可能としている。しかし予測モードを参照する手法は、フレーム毎に実行時間削減率に大きなばらつきが生じると考えることができ、リアルタイム符号化などの低遅延用途には向かず蓄積用途のみに有効な手法である。この他にも符号化対象 CTU の空間的あるいは時間的近傍の CTU の深さ情報を用いた手法 [8] や、処理中 CU の SATD などの指標を用いた手法 [9] などが報告されている。しかしながら近傍のブロックの情報を用いるため高並列化に向かない、数多くのパラメータの調整が必要など、それぞれ問題点がある。これらの関連研究の調査から、処理中 CU 内の情報のみを用いた手法を主軸とした高い符号化時間削減率を実現した手法の提案が望まれることが分かる。

### 3 符号化特性調査

対象 CU のテクスチャが平坦または一様な場合には，CU 内の動きがばらついていないため， $N \times 2N$  や  $2N \times N$  などの長方形分割の PU ではなく，正方形の  $2N \times 2N$  サイズの PU が選択される可能性が高くなると推測される．この仮説に基づき，PU の種類を制限する手法を明らかにするために，輝度値の分散特徴に基づく符号化特性の調査を行ってきた [10][11]．具体的には CU 内の  $2N \times 2N$  画素全ての輝度値の分散値に対する  $2N \times 2N$  選択率と，CU を 4 分割した  $N \times N$  ブロックの各分散値の分散に対する  $2N \times 2N$  選択率を調査し，その特性を示した．図 3.5，3.6 に  $2N \times 2N$  画素の分散値別の  $2N \times 2N$  選択率，図 3.7，3.8 に  $N \times N$  ブロック各分散値の分散に対する  $2N \times 2N$  選択率を示す．なお Random Access 符号化条件を選択し，QP=32 で BasketballDrill, BQMall, Kimono, ParkScene を全フレーム符号化した結果である．またこのグラフは特定の区間の値域にある CU での  $2N \times 2N$  選択率が示してある．例えば分散値が 400 より大きく 600 以下の CU が  $2N \times 2N$  を選択した割合が示してある．

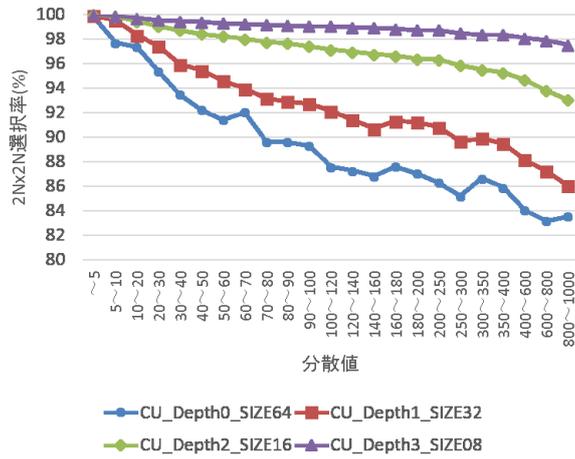


(a)BasketballDrill

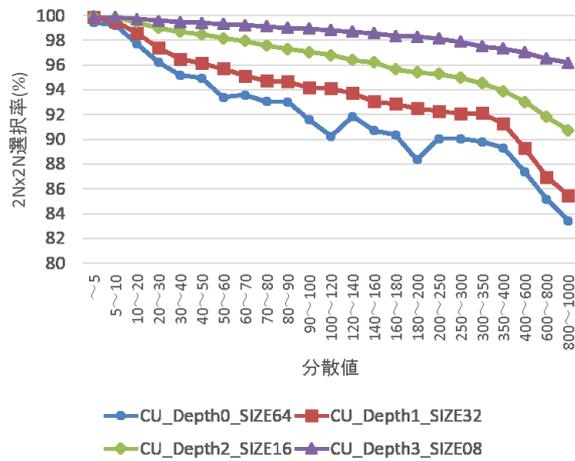


(b)BQMall

図 3.5: 2Nx2N 画素の分散値別の 2Nx2N 選択率 1

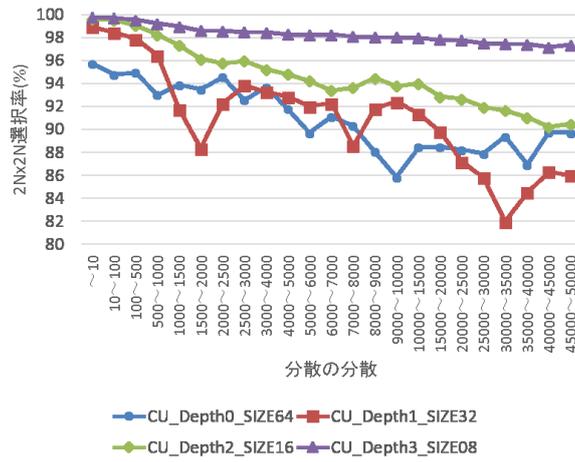


(c)Kimono

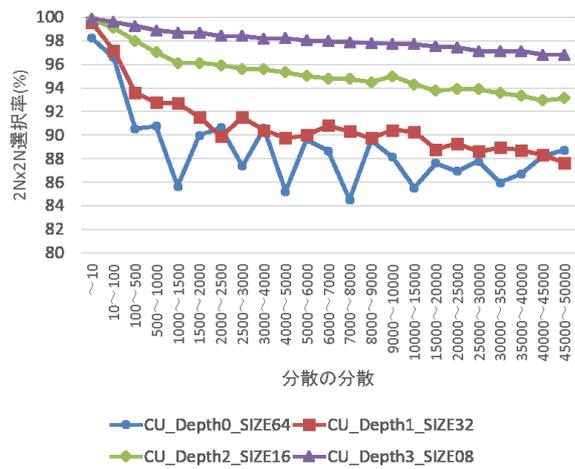


(d)ParkScene

図 3.6: 2Nx2N 画素の分散値別の 2Nx2N 選択率 2

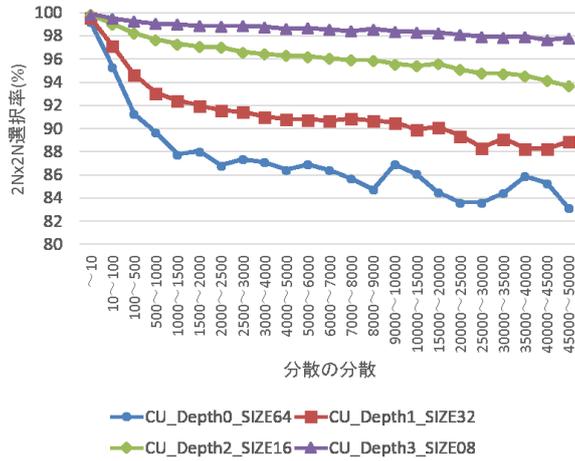


(a)BasketballDrill

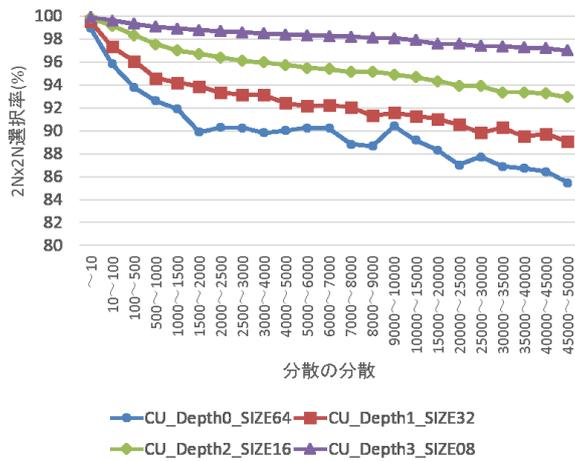


(b)BQMall

図 3.7:  $N \times N$  ブロック各分散値の分散に対する  $2N \times 2N$  選択率 1



(c)Kimono



(d)ParkScene

図 3.8:  $N \times N$  ブロック各分散値の分散に対する  $2N \times 2N$  選択率

図 3.5, 3.6, 3.7, 3.8 より予想通り対象 CU のテクスチャが平坦または対称の場合は,  $2N \times 2N$  サイズの PU が選択される割合が高いことが明らかになった。更には階層別に選択率の傾向が異なることも確認できた。CU サイズが  $64 \times 64$  のように大きな場合は, 分散値が大きくなると急激に選択率が減少するのに対し, CU サイズが  $8 \times 8$  のように小さな場合は分散値が大きくなっても, ある程度高い選択率を維持することが分かる。

これらの結果から分散値が小さい場合は  $2N \times 2N$  が選択される割合が高いため, その他の PU を用いたインター予測の試行が不必要であることが分かる。よって候補 PU を分散値に応じて制限することで, 大幅なエンコード時間削減が期待できる。

## 4 提案手法

### 4.1 対象 CU 内輝度値分散特徴を利用した PU 削減法

#### 4.1.1 正方形 PU への限定手法

3章では対象 CU のテクスチャが平坦または対称の場合に、 $2N \times 2N$  サイズの PU が選択される割合が高いことを明らかにした。これらの符号化特性の調査結果に基づき、インター予測に用いる候補 PU を大幅に削減する手法を提案する。3章と同様に輝度値の分散特徴として、対象 CU の画素の平坦さを表す  $2N \times 2N$  画素全ての輝度値に対する分散値 ( $2N \times 2N$  分散)、CU 内のテクスチャの対称性を表す CU を 4 分割した  $N \times N$  ブロックの各分散値の分散 ( $N \times N$  分散値の分散) を、提案手法でも用いる。輝度値の分散は (2) の式により計算する。ただし  $n$  は構成画素数、 $ave$  は輝度値の平均値である。

$$V = \frac{1}{n} \sum_{i=1}^n (Luminance_i - ave)^2 \quad (2)$$

$2N \times 2N$  分散値は式 (2) により導出できる (構成画素数  $n = 2N \times 2N$ )。  $N \times N$  分散値の分散は、まず図 4.9 に示すように  $2N \times 2N$  の CU を四分割し、四分割したブロックそれぞれで、式 (2) を用いて分散値  $V_{LU}, V_{RU}, V_{LB}, V_{RB}$  を計算する (構成画素数はそれぞれ  $n = N \times N$ )。計算した 4 つの分散値 ( $V_{LU}, V_{RU}, V_{LB}, V_{RB}$ ) の間で、更に分散を計算し、この値を  $N \times N$  分散値の分散とする。計算された  $2N \times 2N$  分散または  $N \times N$  分散値の分散の値が小さい場合は  $2N \times 2N$  の PU の選択率が高いことから、これらの値どちらかが閾値以下の場合には、インター予測に用いる PU を  $2N \times 2N$  に限定する。

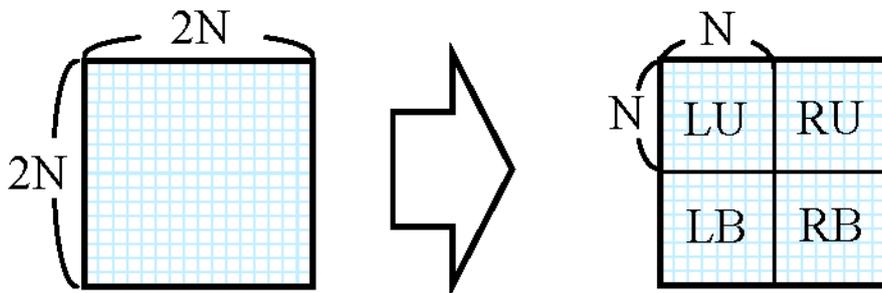


図 4.9:  $N \times N$  分散値の分散

#### 4.1.2 対称分割 PU の候補削減手法

4.1.1 章により，対象 CU 全体のテクスチャが平坦または対称の場合に，大幅に PU の種類を限定することができる．しかしながら，CU 内で局所的な特徴がある場合は，CU 内全ての輝度を用いた分散では値が大きくなり，選択される PU も正方形の  $2N \times 2N$  ではなく長方形の  $N \times 2N$  や  $2N \times N$  が選択されると考えられる．また CU 内の局所的特徴が存在する領域によって，縦方向分割の  $N \times 2N$  または横方向分割  $2N \times N$  の適否がある程度推定できるとも考えられる．そこで対象 CU を縦方向分割した各ブロック ( $N \times 2N$ ) と横方向分割した各ブロック ( $2N \times N$ ) でも同様に，輝度値の分散や，分散値の分散を計算する．

図 4.10 に示す縦方向分割 ( $N \times 2N$ ) を例に計算方法を説明する．通常の分散は式 (2) により同様に求められる (構成画素数  $n = N \times 2N$ )．なおこの分散は 2 つの  $N \times 2N$  サイズのブロックそれぞれで計算する．分散値の分散も同様に 2 つの  $N \times 2N$  サイズのブロック別々に 2 つ求める．図 4.10 の左側の  $N \times 2N$  ブロック (LU と LB で構成されるブロック) の分散値の分散を求めるには，まず式 (2) により  $V_{LU}, V_{LB}$  を求める (構成画素数はそれぞれ  $n = N \times N$ )．この  $V_{LU}, V_{LB}$  間で更に分散を計算した値が左側の  $N \times 2N$  ブロックの分散値の分散になる．横方向分割の  $2N \times N$  の場合も同様に計算することができる．

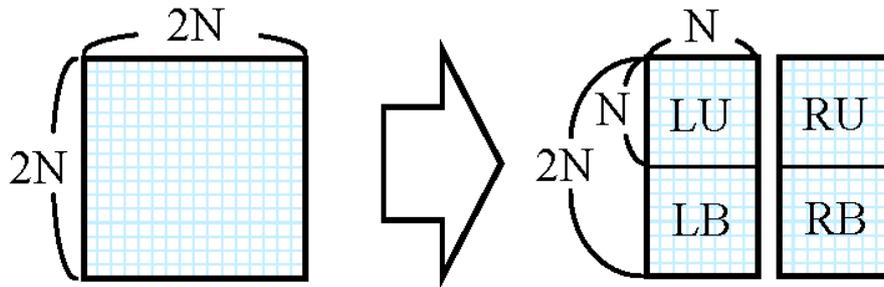


図 4.10:  $N \times 2N$  における計算方法

$N \times 2N$  の分散 2 つともに関値以下，または  $N \times 2N$  の分散値の分散 2 つともに関値以下の場合， $2N \times N$  と非対称分割 PU をインター予測の候補 PU から除外する．また， $2N \times N$  の分散 2 つともに関値以下，または  $2N \times N$  の分散値の分散 2 つともに関値以下の場合， $N \times 2N$  と非対称分割 PU をインター予測の候補 PU から除外する． $N \times 2N$ ， $2N \times N$  とともに分散値が関

値以下の場合、4.1.1 章の手法での条件には当てはまらないが、ある程度 CU 全体のテクスチャが平坦または一様であると推測し、インター予測の候補 PU を  $2N \times 2N$  のみに限定する。

#### 4.1.3 非対称 PU の候補削減手法

4.1.2 章の手法を拡張し非対称分割 PU についても同様に分散と分散値の分散を計算する。計算方法は上述の  $N \times 2N$  と同様である。ただし  $V_{LU}, V_{RU}, V_{LB}, V_{RB}$  は、図 4.11 のようにそれぞれ構成画素数  $n = N \times (N/2)$  または  $N \times (3N/2)$  のブロックで求める。 $2N \times nU, 2N \times nD, nL \times 2N, nR \times 2N$  それぞれで分散と分散値の分散を計算し、これらの値が閾値以下となる分割方法がある場合、他の非対称 PU を候補 PU から除外する。条件を満たす PU が複数個ある場合は、再び候補 PU を  $2N \times 2N$  のみに限定する。

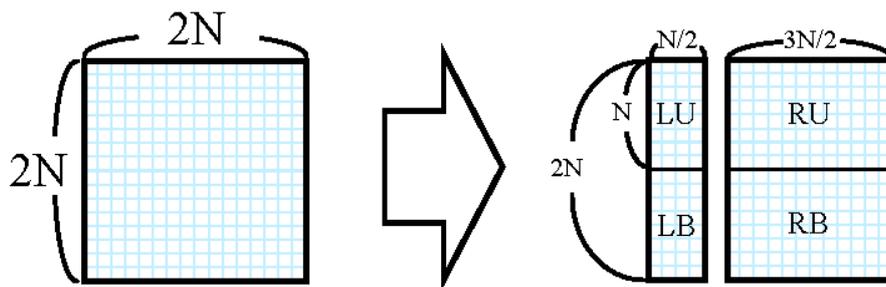


図 4.11:  $nL \times 2N$  における計算方法

#### 4.1.4 提案手法の流れ

4.1.1, 4.1.2, 4.1.3 章を併せた手法の流れを説明する。

Step1)  $2N \times 2N$  分散と  $N \times N$  分散値の分散を計算する。

Step2)  $2N \times 2N$  分散と  $N \times N$  分散値の分散どちらかが閾値以下ならばインター予測候補 PU を  $2N \times 2N$  のみに限定し、Step6 へ移行する。

Step3) 縦方向分割の  $N \times 2N$ , 横方向分割の  $2N \times N$  それぞれで、分散と分散値の分散を計算する。

Step4)  $N \times 2N$  の分散または分散値の分散が閾値以下の場合は、候補 PU から  $2N \times N$  と AMP を除外する。 $2N \times N$  の分散または分散値の分散が閾値以下の場合は、候補 PU から  $N \times 2N$  と AMP を除外して Step6 へ移行する。ただし、 $N \times 2N$ 、 $2N \times N$  の分散または分散値の分散がともに閾値以下なら候補 PU を  $2N \times 2N$  のみに限定する。

Step5) AMP のいずれかの PU の分散または分散値の分散が閾値以下の場合は、閾値以下となる PU 以外の非対称分割 PU を候補から除外する。ただし、条件を満たす PU が複数個ある場合は、再び候補 PU を  $2N \times 2N$  のみに限定する。

Step6) 候補 PU を用いたインター予測、及びイントラ予測を試行する。

## 4.2 CU 内輝度値に特徴がない場合の PU 削減法

4.1 章では対象 CU 内の輝度値分散特徴を利用し、候補 PU を大幅に低減する手法を提案した。しかしながら対象 CU 内のテクスチャが非常に複雑で、分散値が小さくなる分割方法が存在せず、現階層では適切な PU の形状が無いと推測できる場合がある。その場合は最終的に、より細かく四分木分割された階層で符号化されるとも考えられる。そこで、4.1 章の手法では分散値が閾値以下になる PU が見つからず、なおかつ現階層が CTU の最下層である深さ 3 でなければ、更に小さなサイズの CU で符号化されると推測し、処理中 CU ではインター予測の候補 PU を  $2N \times 2N$  のみに限定する。この手法を 4.1 章の手法に高速化オプションとして追加する。

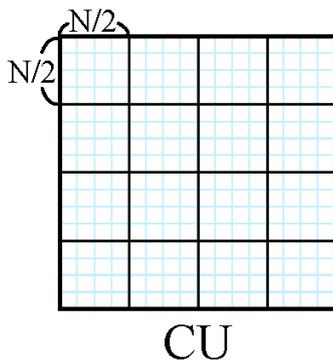
## 5 評価

### 5.1 実装方法

提案手法では様々な形状のブロックで分散や分散値の分散を求める．上述の提案手法の説明では，説明の簡単化のために分散の式を (2) のように各輝度値と平均値の差分の二乗で表していたが，式 (3) のように変形できる．

$$V = E(Luminance^2) - E(Luminance)^2 \quad (3)$$

従って図 5.12 のように CU を 16 分割した  $(N/2) \times (N/2)$  ブロックでそれぞれ輝度値の二乗の平均と輝度値の平均を計算して，これらを組み合わせることにより様々な形状のブロックの分散値や分散値の分散を計算できる．よって画素の走査は各 CU 一回のみで済むことから，分散の計算コストは非常に小さい．



CUを16分割した  
各 $(N/2) \times (N/2)$ ブロックで  
 $E(X^2)$ ,  $E(X)$ を計算

図 5.12: 実装方法

### 5.2 評価条件

H.265/HEVC 参照ソフトウェア HM-10.0 に提案手法を実装し，その性能を確認した．表 5.1 に用いたテストシーケンス，表 5.2 に符号化条件，表 5.3 に実行環境を示す．なお評価項目は Bitrate，Y-PSNR，エン

コード時間  $ET$ (Encoding Time) , インター予測時間  $IPT$ (Inter Prediction Time) を採り ,  $BD$ -rate と式 (4)(5) を用いてエンコード時間削減率を表す  $ETRR$ (Encoding Time Reduction Ratio) とインター予測時間削減率を表す  $IPTRR$ (Inter Prediction Time Reduction Ratio) も算出した . また Random Access では全フレームを , Low Delay では 25 フレームを符号化した .

$$ETRR = \frac{ET_{HM-10.0} - ET_{proposed}}{ET_{HM-10.0}} \times 100 \quad (4)$$

$$IPTRR = \frac{IPT_{HM-10.0} - IPT_{proposed}}{IPT_{HM-10.0}} \times 100 \quad (5)$$

表 5.1: テストシーケンス

	解像度	フレーム数	フレームレート
BasketballDrill	832x480	500	50
BQMall	832x480	600	60
Kimono	1920x1080	240	24
ParkScene	1920x1080	240	24

表 5.2: 符号化条件

Max CU size	64x64
CU depth	0-3
QP	22, 27, 32, 37
GOP structure	Random Access, Low Delay

表 5.3: 実行環境

OS	Windows Server 2012
CPU	Intel Xeon CPU E3-1240 V2 3.40GHz
RAM	16.0 GB

### 5.3 評価結果

提案手法の評価結果を表 5.4, 5.5, 5.6, 5.7 に示す。なお分散の閾値を 100, 分散の分散の閾値を 2000 に設定した手法, 分散の閾値を 250, 分散の分散の閾値を 5000 に設定し, 4.2 章の高速化オプションを適用する場合としない場合の二通りの手法の, 計三種類の手法で評価を行った結果である。

表 5.4: Random Access における評価結果 1

		Random Access							
		HM				提案手法 (100, 2000, 高速化オプション無し)			
Sequence	QP	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)
BasketballDrill	22	3616.78	40.50	6535.24	4727.32	3633.00	40.48	4578.97	2755.01
	27	1755.93	37.40	5496.41	4284.10	1761.39	37.38	3783.22	2552.56
	32	865.62	34.46	4748.39	3920.48	867.55	34.44	3205.50	2377.12
	37	459.20	31.94	4263.01	3647.54	459.53	31.93	2853.28	2236.00
BQMall	22	3831.91	40.23	6980.50	5274.34	3864.17	40.21	5159.08	3433.00
	27	1823.89	37.74	5901.80	4776.20	1837.89	37.72	4294.96	3160.70
	32	931.35	35.02	5247.52	4435.19	936.55	35.00	3770.62	2966.21
	37	498.61	32.28	4808.75	4181.50	500.34	32.26	3456.81	2820.72
Kimono	22	4732.49	41.60	15090.49	11750.75	4754.58	41.59	8165.74	4852.60
	27	2159.37	39.73	12794.86	10556.77	2173.28	39.72	6809.06	4569.88
	32	1053.09	37.42	11389.73	9757.92	1057.95	37.40	6019.05	4369.27
	37	533.29	35.03	10355.29	9071.83	534.67	35.01	5482.24	4183.56
ParkScene	22	7406.56	40.05	13995.26	10757.10	7460.50	40.03	8974.11	5718.91
	27	3179.04	37.52	11714.24	9687.62	3197.78	37.50	7267.59	5267.22
	32	1450.09	34.91	10375.18	8964.68	1455.73	34.89	6395.05	4977.93
	37	670.63	32.39	9595.54	8474.84	672.71	32.38	5904.77	4780.83
		Random Access							
		提案手法 (250, 5000, 高速化オプション無し)				提案手法 (250, 5000, 高速化オプション有り)			
Sequence	QP	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)
BasketballDrill	22	3644.05	40.48	4094.76	2288.66	3653.27	40.47	3471.19	1657.30
	27	1765.87	37.37	3357.60	2130.60	1771.59	37.36	2796.73	1571.81
	32	868.99	34.43	2823.09	1987.75	872.18	34.42	2331.93	1489.57
	37	461.09	31.92	2500.86	1876.52	462.01	31.90	2061.81	1437.09
BQMall	22	3881.86	40.21	4693.19	2958.29	3896.66	40.20	3810.39	2069.43
	27	1846.08	37.71	3864.39	2726.70	1855.75	37.71	3094.99	1945.91
	32	939.90	34.98	3402.26	2588.22	946.50	34.97	2692.87	1870.77
	37	501.53	32.25	3113.64	2474.67	505.38	32.22	2475.39	1826.68
Kimono	22	4762.62	41.58	7275.02	3948.89	4765.22	41.58	6899.93	3557.59
	27	2176.44	39.71	6007.41	3781.17	2178.54	39.71	5663.73	3400.35
	32	1059.71	37.40	5256.70	3617.74	1060.56	37.39	4936.64	3285.87
	37	535.83	35.01	4774.04	3489.92	536.57	35.00	4491.11	3178.32
ParkScene	22	7482.60	40.03	7708.65	4461.80	7493.10	40.03	7008.05	3729.32
	27	3207.54	37.49	6157.91	4160.33	3213.92	37.49	5561.03	3536.95
	32	1458.27	34.88	5393.84	3979.56	1461.20	34.88	4859.26	3427.53
	37	672.82	32.37	4976.21	3842.04	674.58	32.36	4466.64	3335.41

表 5.5: Random Access における評価結果 2

Random Access				
提案手法 (100, 2000, 高速化オプション無し)				
Sequence	QP	BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	0.7	29.93	41.72
	27		31.17	40.42
	32		32.49	39.37
	37		33.07	38.70
BQMall	22	1.2	26.09	34.91
	27		27.23	33.82
	32		28.14	33.12
	37		28.11	32.54
Kimono	22	1.0	45.89	58.70
	27		46.78	56.71
	32		47.15	55.22
	37		47.06	53.88
ParkScene	22	1.1	35.88	46.84
	27		37.96	45.63
	32		38.36	44.47
	37		38.46	43.59
Ave		1.0	35.86	43.73
Random Access				
提案手法 (250, 5000, 高速化オプション無し)				
Sequence	QP	BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	1.1	37.34	51.59
	27		38.91	50.27
	32		40.55	49.30
	37		41.34	48.55
BQMall	22	1.9	32.77	43.91
	27		34.52	42.91
	32		35.16	41.64
	37		35.25	40.82
Kimono	22	1.4	51.79	66.39
	27		53.05	64.18
	32		53.85	62.93
	37		53.90	61.53
ParkScene	22	1.6	44.92	58.52
	27		47.43	57.06
	32		48.01	55.61
	37		48.14	54.67
Ave		1.5	43.56	53.12
Random Access				
提案手法 (250, 5000, 高速化オプション有り)				
Sequence	QP	BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	1.7	46.89	64.94
	27		49.12	63.31
	32		50.89	62.01
	37		51.63	60.60
BQMall	22	2.8	45.41	60.76
	27		47.56	59.26
	32		48.68	57.82
	37		48.52	56.32
Kimono	22	1.6	54.28	69.72
	27		55.73	67.79
	32		56.66	66.33
	37		56.63	64.96
ParkScene	22	1.9	49.93	65.33
	27		52.53	63.49
	32		53.16	61.77
	37		53.45	60.64
Ave		2.0	51.32	62.82

表 5.6: Low Delay における評価結果 1

		Low Delay							
		HM				提案手法 (100, 2000, 高速化オプション無し)			
Sequence	QP	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)
BasketballDrill	22	3776.58	40.66	409.58	305.03	3792.61	40.64	278.14	174.11
	27	1809.62	37.53	347.89	277.42	1821.20	37.52	229.01	160.46
	32	883.22	34.60	301.23	254.61	888.45	34.60	196.79	152.35
	37	460.78	32.08	273.87	239.08	462.75	32.08	176.19	143.11
BQMall	22	5188.20	39.85	369.66	277.01	5224.13	39.84	283.31	192.52
	27	2454.99	36.96	304.88	246.98	2480.01	36.95	234.02	174.33
	32	1263.99	33.94	268.64	229.01	1269.72	33.94	202.24	162.80
	37	663.15	30.93	246.25	214.96	665.18	30.91	185.33	156.28
Kimono	22	7251.20	42.15	2491.36	1930.35	7278.34	42.13	1291.78	741.24
	27	3606.21	40.20	2116.27	1720.90	3631.29	40.19	1086.42	698.87
	32	1786.39	37.51	1827.53	1558.72	1801.74	37.50	941.44	669.01
	37	900.76	34.82	1607.09	1407.50	903.34	34.79	832.76	638.41
ParkScene	22	9613.51	40.00	2048.15	1587.82	9687.16	39.98	1257.21	797.91
	27	4045.56	37.27	1668.83	1401.56	4080.33	37.25	1009.90	739.38
	32	1793.71	34.57	1449.60	1272.21	1804.72	34.56	869.51	692.19
	37	799.42	32.07	1323.53	1192.01	800.76	32.07	786.51	655.36
		Low Delay							
		提案手法 (250, 5000, 高速化オプション無し)				提案手法 (250, 5000, 高速化オプション有り)			
Sequence	QP	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)	Bitrate (kbps)	Y-PSNR (dB)	ET (sec)	IPT (sec)
BasketballDrill	22	3818.38	40.64	252.35	145.45	3830.51	40.64	209.73	105.82
	27	1827.25	37.51	207.62	136.61	1828.48	37.50	171.96	101.84
	32	889.41	34.60	174.37	129.08	889.65	34.57	143.44	97.43
	37	462.90	32.07	156.99	123.44	464.00	32.06	127.72	94.69
BQMall	22	5244.29	39.84	261.03	168.20	5259.44	39.83	200.32	107.79
	27	2490.16	36.95	214.31	155.48	2498.69	36.94	162.54	103.42
	32	1274.71	33.92	185.62	145.96	1283.06	33.91	140.27	99.29
	37	666.01	30.91	170.51	139.86	675.48	30.90	126.16	95.19
Kimono	22	7287.10	42.13	1145.13	595.83	7293.93	42.13	1095.57	544.40
	27	3631.49	40.18	951.11	565.00	3630.87	40.18	911.87	517.16
	32	1805.43	37.50	818.12	546.14	1805.30	37.50	773.06	494.67
	37	904.70	34.78	717.72	520.07	904.57	34.77	674.88	476.84
ParkScene	22	9713.6	39.98	1074.03	616.96	9719.64	39.98	984.75	523.72
	27	4084.85	37.24	845.18	580.34	4087.93	37.24	767.15	493.48
	32	1806.51	34.55	735.48	557.20	1810.11	34.55	653.75	475.85
	37	803.80	32.05	660.51	530.27	802.65	32.05	594.90	463.78

表 5.7: Low Delay における評価結果 2

		Low Delay		
		提案手法 (100, 2000, 高速化オプション無し)		
Sequence	QP	BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	0.7	32.09	42.92
	27		34.17	42.16
	32		34.67	40.16
	37		35.67	40.14
BQMall	22	0.9	23.36	30.50
	27		23.24	29.42
	32		24.72	28.91
	37		24.74	27.30
Kimono	22	1.2	48.15	61.60
	27		48.66	59.39
	32		48.49	57.08
	37		48.18	54.64
ParkScene	22	1.1	38.62	49.75
	27		39.48	47.25
	32		40.02	45.59
	37		40.58	45.02
Ave		1.0	36.55	43.86
		Low Delay		
		提案手法 (250, 5000, 高速化オプション無し)		
Sequence	QP	BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	1.2	38.39	52.32
	27		40.32	50.76
	32		42.12	49.30
	37		42.68	48.37
BQMall	22	1.4	29.39	39.28
	27		29.71	37.05
	32		30.90	36.27
	37		30.76	34.94
Kimono	22	1.4	54.04	69.13
	27		55.06	67.17
	32		55.23	64.96
	37		55.34	63.05
ParkScene	22	1.5	47.56	61.14
	27		49.35	58.59
	32		49.26	56.20
	37		50.10	55.51
Ave		1.4	43.76	52.75
		Low Delay		
		提案手法 (250, 5000, 高速化オプション有り)		
Sequence	QP	BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	1.7	48.79	65.31
	27		50.57	63.29
	32		52.38	61.74
	37		53.36	60.40
BQMall	22	2.2	45.81	61.09
	27		46.69	58.13
	32		47.78	56.64
	37		48.77	55.72
Kimono	22	1.4	56.03	71.80
	27		56.91	69.95
	32		57.70	68.26
	37		58.01	66.12
ParkScene	22	1.7	51.92	67.02
	27		54.03	64.79
	32		54.90	62.60
	37		55.05	61.09
Ave		1.8	52.42	63.37

表 5.4, 5.5 より, Random Access 符号化構造において提案手法 (100, 2000, 高速化オプション無し) はBD-rateの悪化を平均1.0%に抑えつつ, エンコード時間を平均で35.86%削減, インター予測時間を平均で43.73%削減できた. 提案手法 (250, 5000, 高速化オプション無し) ではBD-rateの悪化を平均1.5%に抑えつつ, エンコード時間を平均で43.56%削減, インター予測時間を平均で53.12%削減できた. 提案手法 (250, 5000, 高速化オプション有り) ではBD-rateの悪化を平均2.0%に抑えつつ, エンコード時間を平均で51.32%削減, インター予測時間を平均で62.82%削減できた.

また, 表 5.6, 5.7 より, Low Delay 符号化構造において提案手法 (100, 2000, 高速化オプション無し) はBD-rateの悪化を平均1.0%に抑えつつ, エンコード時間を平均で36.55%削減, インター予測時間を平均で43.86%削減できた. 提案手法 (250, 5000, 高速化オプション無し) ではBD-rateの悪化を平均1.4%に抑えつつ, エンコード時間を平均で43.76%削減, インター予測時間を平均で52.75%削減できた. 提案手法 (250, 5000, 高速化オプション有り) ではBD-rateの悪化を平均1.8%に抑えつつ, エンコード時間を平均で52.42%削減, インター予測時間を平均で63.37%削減できた.

近傍のCTUの深さを参照する手法や, HMに組み込まれているECUオプションのような予測モードを用いる手法では, QP毎にエンコード時間削減率に大きな差が出てしまう. しかしながら, 提案手法ではQPによらず高いエンコード時間削減率を実現していることがわかる. また1920x1080解像度のシーケンスの方がエンコード時間削減率が高いことも分かる. これは同じ映像でも, 解像度が大きい方がそれぞれのブロックのテクスチャが平坦になるため, 提案手法の効果が大きくなったと考えられる. このため, 提案手法は高精細画像に対して特に有効な手法であるともいえる.

ここで, 関連研究である[9]と提案手法の性能を比較する. [9]は $2N \times 2N$ サイズのPUを用いたインター予測エラーの分散やSATDを用いた手法で, 非常に高性能な手法として知られている. そこで実験に用いたシーケンスの中で, 共通するBasketballDrillとKimonoにおいてBD-rateとエンコード時間削減率を比較する. ここではBD-rateとエンコード時間削減率ともに上回った提案手法 (250, 5000, 高速化オプション無し) の結果を表5.8, 5.9に示す. ただしHMのアンカーやエンコードするフレーム数が異なるなど, 実験条件が異なる点もあるため, 提案手法の有効性を示すための参考程度の比較である.

表 5.8: 関連研究との比較 (Random Access)

	Random Access			
	[9]		提案手法 (高速化オプション無し)	
	BD-rate (%)	ETRR (%)	BD-rate (%)	ETRR (%)
BasketballDrill	1.8	30.3	1.1	39.5
Kimono	1.5	61.0	1.4	53.1
Ave	1.7	45.7	1.3	46.3

表 5.9: 関連研究との比較 (Low Delay)

	Low Delay			
	[9]		提案手法 (高速化オプション無し)	
	BD-rate (%)	ETRR (%)	BD-rate (%)	ETRR (%)
BasketballDrill	1.9	28.4	1.2	40.9
Kimono	1.8	42.1	1.4	54.9
Ave	1.9	35.3	1.3	47.9

表 5.8, 5.9 より, 関連研究 [9] と比較して BD-rate と ETRR 両指標で提案手法の方が勝っているため, 提案手法が非常に高性能であることが分かる.

## 5.4 分散の計算時間

5.1 章に提案手法の実装方法を示し分散の計算コストの小ささを述べたが、実際の計算時間についても評価した。Random Access 及び Low Delay 符号化構造における符号化時間全体に占める提案手法の分散の計算時間の割合を表 5.10, 5.11 に示す。Random Access, Low Delay いずれも 25 フレーム符号化した結果である。

表 5.10, 5.11 より、分散の計算時間は全体の 0.2% 以下であることから、計算量が非常に少ないことが分かる。

全体のエンコード時間削減率が高くても、手法の実装コストが高いものがある中、提案手法は計算コストが小さい上に輝度値の分散を計算するだけで良いのでハードウェアでも実装が容易なため複雑な手法に比べて利点がある。

表 5.10: Random Access における分散の計算時間

Random Access				
提案手法 (100, 2000, 高速化オプション無し)				
Sequence	QP	ET (sec)	分散の計算時間 (sec)	割合 (%)
BasketballDrill	22	218.16	0.24	0.11
	27	181.24	0.27	0.15
	32	155.74	0.09	0.06
	37	138.70	0.16	0.12
BQMall	22	217.59	0.24	0.11
	27	181.24	0.23	0.13
	32	159.54	0.17	0.11
	37	144.90	0.12	0.08
Kimono	22	991.39	0.83	0.08
	27	815.94	0.76	0.09
	32	699.96	0.89	0.13
	37	623.88	0.79	0.13
ParkScene	22	930.73	0.77	0.08
	27	760.79	0.84	0.11
	32	658.12	0.86	0.13
	37	606.58	0.68	0.11
提案手法 (250, 5000, 高速化オプション無し)				
Sequence	QP	ET (sec)	分散の計算時間 (sec)	割合 (%)
BasketballDrill	22	197.13	0.12	0.06
	27	162.96	0.26	0.16
	32	136.52	0.21	0.15
	37	121.44	0.17	0.14
BQMall	22	197.57	0.2	0.10
	27	164.52	0.22	0.13
	32	145.65	0.25	0.17
	37	131.18	0.16	0.12
Kimono	22	895.76	0.97	0.11
	27	718.94	0.77	0.11
	32	611.98	0.91	0.15
	37	548.67	0.92	0.17
ParkScene	22	804.43	0.85	0.11
	27	635.37	0.83	0.13
	32	558.41	0.75	0.13
	37	507.79	0.61	0.12
提案手法 (250, 5000, 高速化オプション有り)				
Sequence	QP	ET (sec)	分散の計算時間 (sec)	割合 (%)
BasketballDrill	22	170.63	0.2	0.12
	27	136.07	0.22	0.16
	32	114.39	0.13	0.11
	37	100.27	0.12	0.12
BQMall	22	158.90	0.19	0.12
	27	127.08	0.16	0.13
	32	111.77	0.14	0.13
	37	100.83	0.27	0.27
Kimono	22	861.48	0.84	0.10
	27	687.33	0.88	0.13
	32	581.95	0.89	0.15
	37	512.58	0.75	0.15
ParkScene	22	736.96	0.66	0.09
	27	583.41	0.77	0.13
	32	507.32	0.67	0.13
	37	465.59	0.91	0.20

表 5.11: Low Delay における分散の計算時間

Low Delay				
提案手法 (100, 2000, 高速化オプション無し)				
Sequence	QP	ET (sec)	分散の計算時間 (sec)	割合 (%)
BasketballDrill	22	278.14	0.11	0.04
	27	229.01	0.16	0.07
	32	196.79	0.11	0.06
	37	176.19	0.20	0.11
BQMall	22	283.31	0.14	0.05
	27	234.02	0.16	0.07
	32	202.24	0.06	0.03
	37	185.33	0.17	0.09
Kimono	22	1291.78	0.63	0.05
	27	1086.42	0.96	0.09
	32	941.44	0.96	0.10
	37	832.76	0.91	0.11
ParkScene	22	1257.21	0.87	0.07
	27	1009.90	0.86	0.09
	32	869.51	0.61	0.07
	37	786.51	0.81	0.10
提案手法 (250, 5000, 高速化オプション無し)				
Sequence	QP	ET (sec)	分散の計算時間 (sec)	割合 (%)
BasketballDrill	22	252.35	0.17	0.07
	27	207.62	0.20	0.10
	32	174.37	0.10	0.06
	37	156.99	0.19	0.12
BQMall	22	261.03	0.16	0.06
	27	214.31	0.26	0.12
	32	185.62	0.08	0.04
	37	170.51	0.11	0.06
Kimono	22	1145.13	1.02	0.09
	27	951.11	0.85	0.09
	32	818.12	0.75	0.09
	37	717.72	0.73	0.10
ParkScene	22	1074.03	1.03	0.10
	27	845.18	1.09	0.13
	32	735.48	0.74	0.10
	37	660.51	0.54	0.08
提案手法 (250, 5000, 高速化オプション有り)				
Sequence	QP	ET (sec)	分散の計算時間 (sec)	割合 (%)
BasketballDrill	22	209.73	0.14	0.07
	27	171.96	0.13	0.08
	32	143.44	0.17	0.12
	37	127.72	0.19	0.15
BQMall	22	200.32	0.17	0.08
	27	162.54	0.06	0.04
	32	140.27	0.23	0.16
	37	126.16	0.20	0.16
Kimono	22	1095.57	0.89	0.08
	27	911.87	0.74	0.08
	32	773.06	0.81	0.10
	37	674.88	0.72	0.11
ParkScene	22	984.75	0.95	0.10
	27	767.15	0.77	0.10
	32	653.75	0.72	0.11
	37	594.90	1.11	0.19

## 5.5 低遅延および蓄積用途における有効性

動画像符号化の用途は、放送用リアルタイム符号化などの低遅延用途、記憶媒体に映像を蓄積する蓄積用途に大別できる。低遅延用途においては、エンコード中の各フレームにおいて高い処理時間削減率を維持することが望ましい。そこで従来の HM による手法、HM に組み込まれている ECU オプション、提案手法についてフレーム別の実行時間及び実行時間削減率を評価する。ここで ECU オプションとは、処理中 CU のベストモードが近傍の動き情報をそのまま用いる Skip モードであった場合に、CTU を枝切る手法である。この ECU オプションは表 5.12, 5.13 に示すように、BD-rate の悪化を平均で 0.4% 以下に抑えてエンコード時間を 30% 以上削減できる手法である。

表 5.12: ECU の性能 (Random Access)

		Random Access		
		ECU		
		BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	0.2	17.97	22.00
	27		26.84	30.63
	32		35.75	39.70
	37		46.13	49.37
BQMall	22	0.6	22.79	27.61
	27		32.35	37.70
	32		42.65	46.51
	37		50.28	53.05
Kimono	22	0.3	12.71	15.24
	27		23.86	26.63
	32		36.42	39.48
	37		46.94	49.70
ParkScene	22	0.6	26.93	31.63
	27		40.87	44.61
	32		52.92	55.97
	37		61.04	62.88
Ave		0.4	36.03	39.54

表 5.13: ECU の性能 (Low Delay)

		Low Delay		
		ECU		
		BD-rate (%)	ETRR (%)	IPTRR (%)
BasketballDrill	22	0.1	10.14	13.06
	27		20.47	23.45
	32		31.06	33.83
	37		42.57	44.77
BQMall	22	0.0	18.47	22.62
	27		26.10	30.26
	32		36.72	40.27
	37		45.15	47.46
Kimono	22	0.1	7.92	9.13
	27		15.20	17.09
	32		25.90	28.41
	37		38.85	41.03
ParkScene	22	0.5	17.12	20.10
	27		30.99	34.08
	32		44.26	46.90
	37		55.90	57.74
Ave		0.2	29.18	31.89

図 5.13, 5.14 に Random Access 条件で QP=32 で Kimono シーケンスを 25 フレームエンコードした場合の，エンコード時間及びエンコード時間削減率を示す．また図 5.15, 5.16 に Low Delay 条件で QP=32 で Kimono シーケンスを 25 フレームエンコードした場合の，エンコード時間及びエンコード時間削減率を示す．

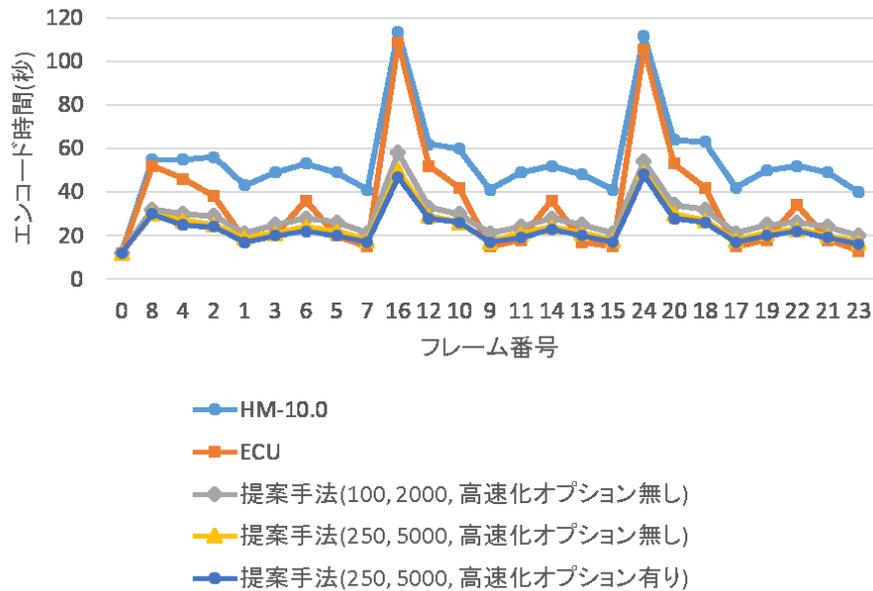


図 5.13: フレーム別エンコード時間 (Random Access)

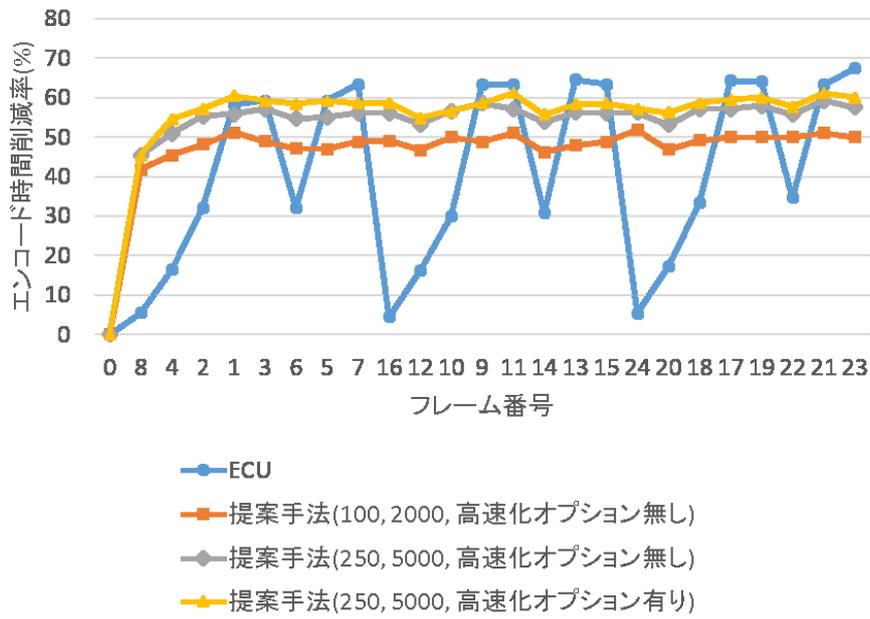


図 5.14: フレーム別エンコード時間削減率 (Random Access)

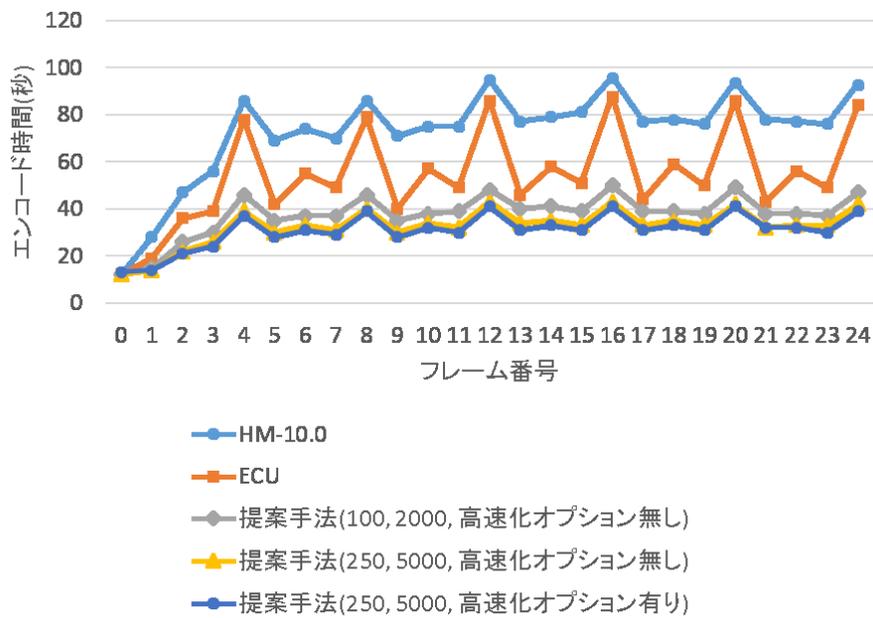


図 5.15: フレーム別エンコード時間 (Low Delay)

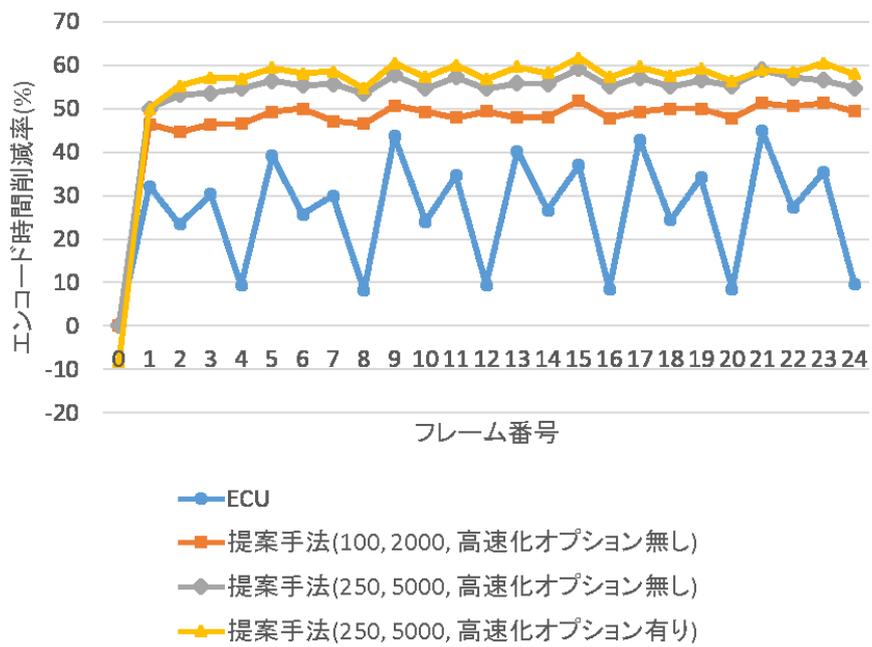


図 5.16: フレーム別エンコード時間削減率 (Low Delay)

図 5.13, 5.14 より, ECU オプションは Random Access 符号化条件において, 特にエンコードが必要なフレームに対してエンコード時間をあまり削減できていないことが分かる. 更にはフレーム毎にエンコード時間削減率に大きなばらつきがあることも分かる. しかしながら提案手法では, エンコード時間が特に必要なフレームにおいても高いエンコード時間削減率を維持していることが分かり, 全てのフレームで安定した性能を発揮できていることが分かる. 同様に Low Delay 符号化条件においても ECU では上記のような欠点があるのに対して, 提案手法は全てのフレームで高いエンコード時間削減率を維持していることから, リアルタイム符号化のような低遅延用途に提案手法が非常に優れていることが分かる. また 5.2 章に示すように全フレーム合計のエンコード時間削減率でも提案手法は高い性能を実現できているため, 蓄積用途にも有効であることが分かる.

## 5.6 考察

5.3, 5.4, 5.5 章では様々な項目で提案手法の有効性を示した。その結果、提案手法は低遅延用途でも蓄積用途でも従来法より高い処理時間低減率の得られることが確認できた。これに加えて提案手法の計算コストの小ささや実装容易性も評価できる。また提案手法では対象 CU 内の輝度値のみしか情報として用いないため、近傍の符号化済みブロックの深さ情報などを参照する関連手法 [8] と比較して高並列化が実現しやすい利点もあることから、総合的にみて非常に優れた手法であるといえる。

## 6 あとがき

本論文では、分散値別に  $2N \times 2N$  サイズ PU の選択率を調査し、それに基づき CU 全体の  $2N \times 2N$  画素の分散値や、CU を構成する  $N \times N$  画素 4 組の各々の分散値の分散、さらには長方形分割でも分散値を用いて候補 PU を大幅に削減する手法を提案した。その結果、Random Access 符号化構造において提案手法は BD-rate の悪化を平均 1.0% に抑えつつ、エンコード時間を平均で 35.86% 削減、インター予測時間を平均で 43.73% 削減できた。また閾値を変更し高速化オプション有効の場合では BD-rate の悪化を平均 2.0% に抑えつつ、エンコード時間を平均で 51.32% 削減、インター予測時間を平均で 62.82% 削減できた。また提案手法の計算コストは非常に小さく、符号化時間全体の 0.2% 以下であり実装の容易性も高い上に、近傍の符号化済みブロックの情報を参照せず処理中 CU 内の情報のみを用いるため高並列化にも優位性がある。また、フレーム別の実行時間削減率も高い数値を維持できるため、リアルタイム符号化などの低遅延用途にも有効であることを示した。

今後の課題として、量子化パラメータに応じた分散値の閾値変更による符号化効率の改善や、動きベクトルなどを用いる手法と組み合わせることによる一層の高速化が挙げられる。

## 謝辞

本論文の執筆にあたりご指導を頂いた近藤利夫教授、佐々木敬泰助教、深澤祐樹研究員に感謝の意を表します。

## 参考文献

- [1] G.J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, “ Overview of the high efficiency video coding (HEVC) standard ”, Circuits and Systems for Video Technology, IEEE Transactions, vol. 22, no.12, pp.1649-1668, 2012.
- [2] A. Lee, D Jun, J. Kim, J.S Choi and J. Kim, “ Efficient Inter Prediction Mode Decision Method for Fast Motion Estimation in High Efficiency Video Coding ”, ETRI Journal, vol. 36, no. 4, August 2014.

- [3] 西村隆太郎, 近藤利夫, 佐々木敬泰, 深澤祐樹, “ H.265/HEVC 対応ブロックサイズ適応型拡大ダイヤモンド動き探索法, ” 信学技報, vol. 115, no. 96, IE2015-39, pp. 41-46, 2015.
- [4] T. Nishikori, T.Nakamura, T Yoshitome, K. Mishiba, “ A fast CU decision using image variance in HEVC intra coding ”, Industrial Electronics and Applications (ISIEA), IEEE Symposium, 2013.
- [5] 宮澤一之, 坂手寛治, 井口和久, “ HEVC インター予測処理における CU 分割決定の高速化, ” 電子情報通信学会ソサイエティ大会講演論文集, A-4 信号処理, pp.78, 2012.
- [6] 小林大祐, 新田高庸, 池田充郎, 上倉一人, 如澤裕尚, “ HEVC イントラ予測処理における CU 分割手法に関する一検討, ” 電子情報通信学会総合大会, D-11 画像工学, pp.85, 2012.
- [7] S. Yang, H. Lee, H.J. Shim and B. Jeon, “ Fast inter mode decision process for HEVC encoder ”, IVMSWP Workshop, pp.1-4, 2013.
- [8] Z. Pan, S.Kwong, Y. Zhang, J. Lei, H. Yuan, “ Fast coding tree unit depth decision for high efficiency video coding ”, Image Processing (ICIP), pp. 3214-3218, 2014.
- [9] X. Shen, L. Yu, and J. Chen, “ Fast coding unit size selection for HEVC based on Bayesian decision rule ”, in Proc. PCS, pp.453-456, 2012.
- [10] 西村隆太郎, 飯田拓真, 近藤利夫, 佐々木敬泰, 深澤祐樹, “ H.265/HEVC インター予測対応 PU 高速決定法, ” 映像情報メディア学会年次大会, 11B 画像符号化, 2016.
- [11] 西村隆太郎, 近藤利夫, 佐々木敬泰, 深澤祐樹, “ 対象 CU 内の輝度値分散特徴を利用したインター予測 PU 高速決定法, ” 信学技報, vol. 116, no. 347, IE2016-97, pp. 103-108, 2016.