

修士論文

題目

次期動画像符号化規格に対応する
分割形状高速決定法

指導教員

若林 哲史 教授

2019年

三重大学大学院 工学研究科 情報工学専攻
ヒューマンインターフェース研究室

飯田 拓真 (417M502)

内容梗概

スーパーハイビジョンの本放送開始など、動画像の高精細化が大きく進んでいる。この高精細化に伴い動画像のデータ量が大幅に増大することから、更なる圧縮率の向上に向けて、次期動画像符号化規格の開発が進められている。しかしながら、開発中の次期動画像符号化規格では分割の自由度を大幅に高めているために、最適な分割形状を定めようとすると、膨大な処理が必要になる。

そこで本論文では、この処理量低減を目指した対象 CTU の近傍 4 つの CTU で検出された動きベクトルの分散値と対象 CTU の輝度値分散を用いて、候補となる分割形状を低減する手法を提案する。また、提案手法を次期動画像符号化規格の参照ソフトウェア JEM-5.0 に実装し、許容されるブロック分割候補のすべてを試行する従来手法に比べ、BD-rate の増加を 5% 程度に抑えながら、エンコード時間を 30% 以上低減できることを示している。

Abstract

In recent years, high definition video technology has been greatly enhanced so that the commercial broadcasting of 8K Ultra HDTV started last this coming December. This enhancement causing explosive video data increase leads to Future Video Coding (FVC) standard. However, the FVC requires huge amount of computational complexity to select the best block partitioning from many candidates in each CUs. Therefore we had proposed two fast block partitioning methods one of which utilized the variance of motion vectors of the four CTU neighboring the target CTU and the other utilized the variance of pixel luminance values in the target CTU.

We implemented the proposed method to the FVC reference software (JEM-5.0) and evaluated its effectiveness that the encoding time was reduced by 30% compared with the current method incorporating in the JEM-5.0.

目次

| | | |
|----------|--------------------------|-----------|
| 1 | はじめに | 1 |
| 2 | 符号化の中核技術となるブロック分割 | 2 |
| 2.1 | ブロック分割の概要 | 2 |
| 2.2 | ブロック分割の拡張 | 2 |
| 2.3 | 関連研究 | 4 |
| 3 | HMでのブロック分割高速決定法 | 4 |
| 3.1 | 近傍ブロックの情報を用いた手法 | 4 |
| 3.2 | 対象ブロックの情報を用いた手法 | 4 |
| 3.3 | FVCへの適用性 | 5 |
| 4 | 分割傾向の分散との関連調査 | 5 |
| 4.1 | 動きベクトル分散値との関連 | 5 |
| 4.2 | 輝度値との関連 | 9 |
| 5 | 提案手法 | 13 |
| 5.1 | 近傍CTUの動きベクトル分散 | 13 |
| 5.2 | 対象CTUの輝度値分散 | 14 |
| 5.3 | 提案手法の流れ | 15 |
| 6 | 評価 | 15 |
| 6.1 | 実装方法 | 15 |
| 6.2 | 評価条件 | 15 |
| 6.3 | 評価結果 | 17 |
| 6.4 | 考察 | 22 |
| 7 | あとがき | 22 |
| | 謝辞 | 23 |
| | 参考文献 | 23 |

目 次

| | | |
|------|--|----|
| 2.1 | QTBT ブロック分割の例 | 3 |
| 4.2 | 動きベクトル分散値別の $2N \times 2N$ 選択率 1 | 6 |
| 4.3 | 動きベクトル分散値別の $2N \times 2N$ 選択率 2 | 6 |
| 4.4 | 動きベクトル分散値別の $2N \times 2N$ 選択率 3 | 7 |
| 4.5 | 動きベクトル分散値別の $2N \times 2N$ 選択率 4 | 7 |
| 4.6 | 輝度値分散値別の $2N \times 2N$ 選択率 1 | 10 |
| 4.7 | 輝度値分散値別の $2N \times 2N$ 選択率 2 | 10 |
| 4.8 | 輝度値分散値別の $2N \times 2N$ 選択率 3 | 11 |
| 4.9 | 輝度値分散値別の $2N \times 2N$ 選択率 4 | 11 |
| 5.10 | 動きベクトル分散の $2N \times 2N$ 限定 | 13 |
| 5.11 | 輝度値分散の $2N \times 2N$ 限定 | 14 |

表 目 次

| | | |
|------|---|----|
| 4.1 | 動きベクトルの分散値別 $2N \times 2N$ 選択個数 | 8 |
| 4.2 | 動きベクトルの分散値別分割種類選択率 | 8 |
| 4.3 | 輝度値の分散値別分割種類選択率 | 12 |
| 6.4 | テストシーケンス | 17 |
| 6.5 | 符号化条件 | 17 |
| 6.6 | 実行環境 | 17 |
| 6.7 | Random Access における評価結果 1 | 18 |
| 6.8 | Random Access における評価結果 2 | 19 |
| 6.9 | Random Access における評価結果 3 | 20 |
| 6.10 | Random Access における評価結果 4 | 21 |

1 はじめに

昨年の12月にスーパーハイビジョンの本放送が開始されるなど、動画像の高精細化が着実に進んでいる。この高精細化に伴い、動画像のデータ量が大幅に増加する中、圧縮率をH.264/AVCの2倍にまで高めるH.265/HEVC[1]が、2013年に標準化されている。さらに、H.265/HEVCの次世代符号化規格(Future Video Coding)の標準化に向けてJVET(Joint Video Exploration Team)が設置され、FVCの開発とその開発段階に対応する参照ソフトウェアJEM(Joint Exploration Model)[2]の提供が継続的に行われている。このJEMにより、現段階のFVCがH.265/HEVCに対して30%程度の性能改善が可能であることが示されている。FVCにおける圧縮率向上の中核技術の一つは、QTBT(Quad Tree plus Binary Tree)構造[3]と呼ばれるブロック分割構造である。基本的な分割方法はH.265/HEVCと変わらないが、分割単位の問題概念が変更されている。H.265/HEVCでは、CU(Coding Unit)とPU(Prediction Unit)を区別して処理をしていたのを、CUのみで処理するよう改めたのがその変更点である。分割の種類の問題概念を取り払い、CUの分割形状を柔軟にすることで、動画像の局所的な変化への対応力を大きく向上させている。しかし、分割形状の柔軟性を高めたことで、分割パターンの候補が増大し、候補を絞り込むための処理量がH.265/HEVCのそれより大幅に増えてしまう。最適なCUの分割パターンを決定するためには、各々のCUで、全ての分割パターンに対してRD(Rate Distortion)コストを評価する必要があるためである。

そこで本論文では、分割パターンの候補が増大してしまう点に着目し、対象CTU内における候補CUの種類を大幅に削減する手法を提案する。ブロック分割の問題点として、動きや模様が一樣な場合には被写体の動きにばらつきがなく、正方形のブロックが選択される傾向があるが、全てのサイズを用いて処理を行うため処理量が膨大となり、エンコード時間が増大する。そこで、これまでHMで効力を確認してきた、近傍CUの検出済み動きベクトルの分散値、対象CUの輝度値の分散値を求め、組み合わせる手法を提案する。この2つの特徴を組み合わせることで被写体の動きを推測する手法をJEMに適用する形で実装し、対象CTUにおける候補CUの種類を削減することによりエンコード時間の低減を試みた。その結果、動きベクトルの分散値が小さい場合には動きベクトル分散を利用し、動きベクトルの分散値が大きい場合には輝度値の分散を利用する新規の手法が、エンコード時間を30%以上低減と最も効果的であることを他の手法と共にJEM-5.0に実装し、比較評価して確認した。

2 符号化の中核技術となるブロック分割

2.1 ブロック分割の概要

予測ブロックのサイズを適切に設定することで圧縮率を大きく向上させることができることから、H.264/AVC で可変ブロックサイズが導入された。H.265/HEVC では、その可変ブロックサイズが、サイズをシーケンス毎に選択・シグナリングできる CTU と呼ばれるブロックを分割する方式に拡張された。この CTU は様々な動画像の局所特性に適応するために 4 分木構造を使用することによって CU に分割される。CU はインター予測やイントラ予測などの予測プロセスを適用するために PU に分割される。H.265/HEVC は CU, PU を含む複数の分割単位概念がある。

2.2 ブロック分割の拡張

FVC では、この CU, PU の分割の単位を、CU 単独とすることで分割パターンの柔軟性を向上させることで、動画像の局所的な変化への対応力を改善している。実際その QTBT と呼ばれる分割構造は、HM 比 30% の圧縮率向上に大きく貢献している。図 2.1 に示す通り QTBT では、従来の 4 分木分割に加えて、動画像の多様な特性に適応できるように水平分割と垂直分割の 2 種類の 2 分木分割も可能としている。図 2.1 で 0 は水平の 2 分木分割、1 は垂直の 2 分木分割を表している。H.265/HEVC の CU 毎に 7 種類の分割パターンの中から一つ選択することで PU を決定する分割法とは異なり、2 分木分割を再帰的に行うようにして、コーディングの柔軟性を大幅に改善している。

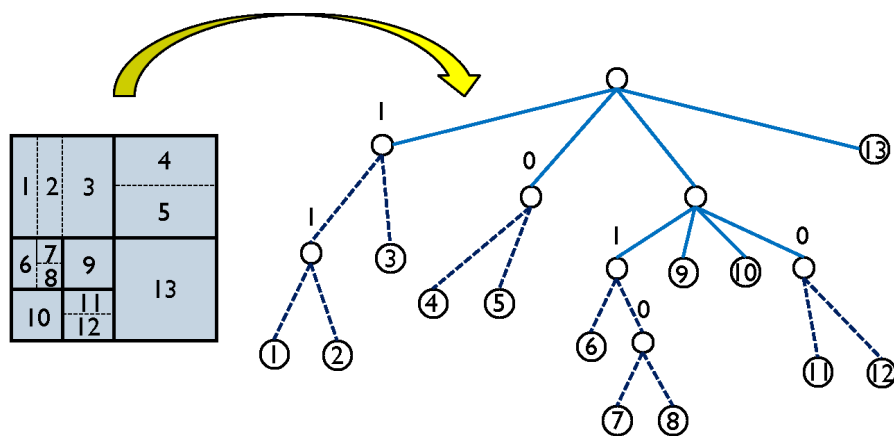


図 2.1: QTBT ブロック分割の例

2.3 関連研究

JEMにおける符号化の高速化については、少ないながらも研究が行われている。エンコード時間低減の効果が大きいものとして、分割パラメータ変数の変更と分割形状の限定を組み合わせた手法 [4] がある。この手法では、BD-rate を 1% 付近に抑えつつ、60% 以上のエンコード時間低減を可能としている。分割パラメータ変数を変更することで、不要な分割の繰り返しを避けることができるため、符号化性能と計算量に違いが出てくる。他の手法に組み合わせることで大きな効果が表れると予想される。また、空間的特徴に基づく FVC イントラ符号化における高速 CU 分割決定アルゴリズム [5] も提案されている。この手法では、BD-rate を 0.6% に抑え、エンコード時間を平均で 23% 低減可能としている。

3 HMでのブロック分割高速決定法

H.265/HEVC では、ブロック分割の問題点に着目して、符号化の高速化について様々な研究が行われている。

3.1 近傍ブロックの情報を用いた手法

符号化済み 4 近傍 CU の動きベクトル分散を用いる手法 [7] が報告されており、被写体の動きを推測することで候補 PU を正方形 ($2N \times 2N$) サイズに限定する。この手法では、エンコード時間を約 42% 低減している。

3.2 対象ブロックの情報を用いた手法

対象 CU 内の輝度値分散を用いて、候補 PU を正方形 ($2N \times 2N$) サイズに限定する手法 [6] が報告されており、この手法では、BD-rate の悪化を平均で 1.5% に抑えつつ、エンコード時間を平均で約 44% 低減している。

また機械学習を用いて CU のサイズを高速決定する手法 [8] も報告されている。この手法では、Intra-Only の演算時間を平均で 58% 低減し、LowDelay-P の演算時間を平均で 72% 低減している。しかし、判定器の判定精度が低くなっているために、BD-rate の悪化が非常に高くなってしまっている。

3.3 FVC への適用性

ブロック単位概念が変更されたが、分割パターンを限定する手法は適用できる。H.265/HEVC から FVC にかけて、30% の性能改善が可能となったが、エンコード時間は 10 倍以上に増加している。これはブロック分割が複雑になったことが大きいので、ブロック分割を限定することで、エンコード時間を低減することができ、エンコード時間が増えてしまう問題点の解決に大いに貢献することができる。

4 分割傾向の分散との関連調査

H.265/HEVC の分割パターン選択に効果のあった近傍動きベクトルの分散、CU 内輝度値分散と分割傾向との関連を調査し、FVC においても分散による分割形状決定が可能かどうかを探る。

4.1 動きベクトル分散値との関連

画像はラスタ走査順に符号化されるので、対象ブロックの左や上は既に符号化済みとなっている。そこで、CU のサイズと動きベクトルが決定している対象ブロックの左、左上、上、右上を 4 近傍のブロックを調査対象とする。4 近傍の動きベクトルを用いて分散を計算することで、動きベクトルの類似性を調べることができる。

図 4.2, 4.3, 4.4, 4.5 に 4 近傍の動きベクトル分散値における $2N \times 2N$ サイズの選択率を示す。表 4.1 に動きベクトルの分散値別 $2N \times 2N$ 選択個数を示す。なお、Random Access 符号化条件を選択し、QP=32 で BasketballDrill, BQMall, Kimono, ParkScene を 25 フレームを符号化した。

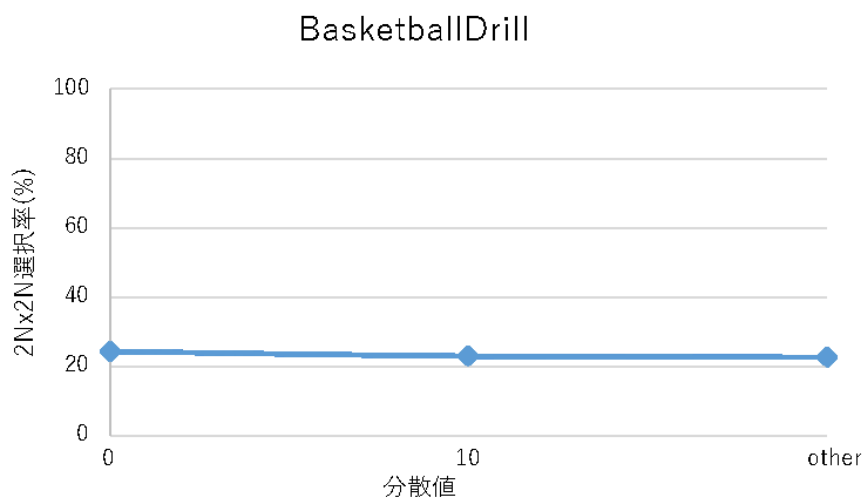


図 4.2: 動きベクトル分散値別の 2Nx2N 選択率 1

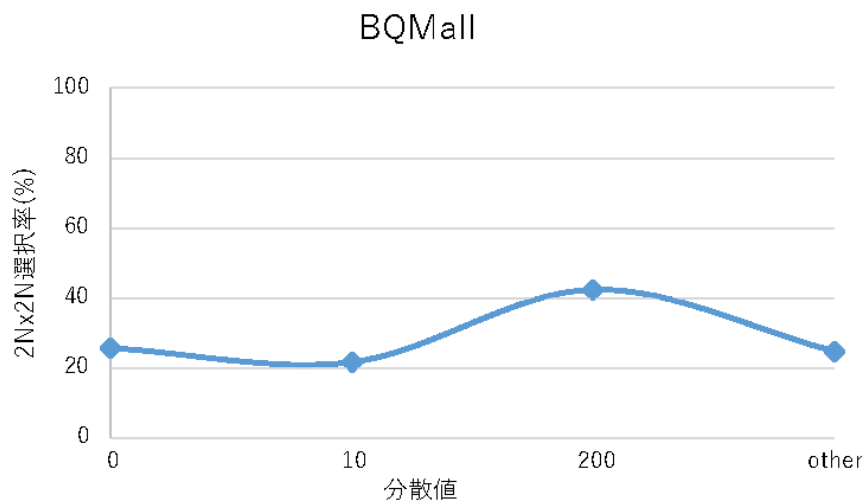


図 4.3: 動きベクトル分散値別の 2Nx2N 選択率 2

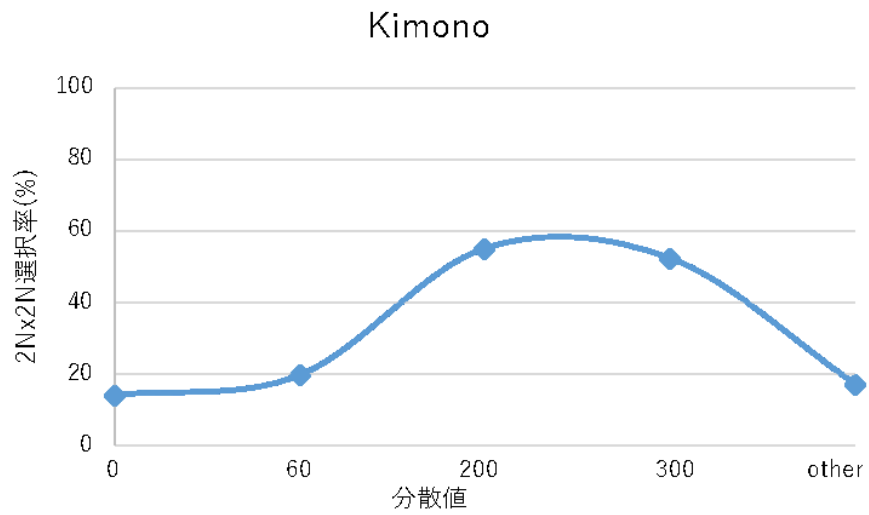


図 4.4: 動きベクトル分散値別の 2Nx2N 選択率 3

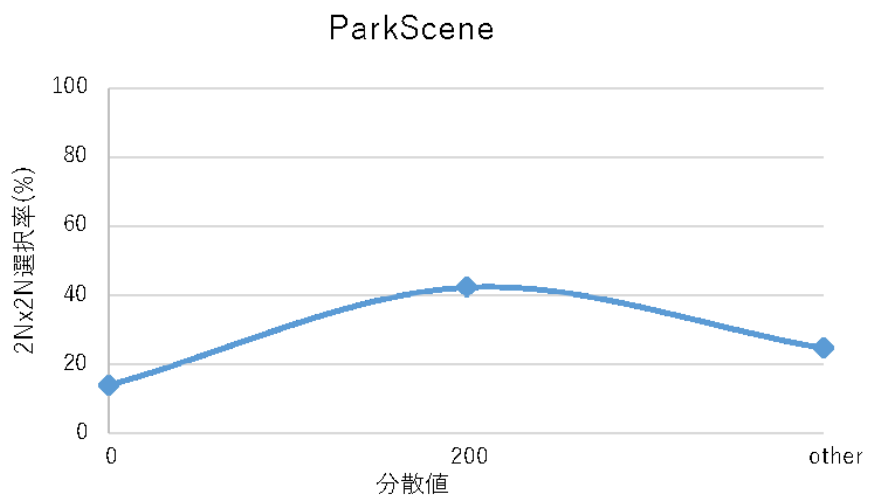


図 4.5: 動きベクトル分散値別の 2Nx2N 選択率 4

表 4.1: 動きベクトルの分散値別 2Nx2N 選択個数

| シーケンス名 | 分散値 | 2Nx2N | All |
|-----------------|--------|--------|---------|
| BasketballDrill | 0-1000 | 58510 | 245184 |
| | other | 119189 | 522957 |
| BQMall | 0-1000 | 69011 | 150924 |
| | other | 272057 | 608869 |
| Kimono | 0-1000 | 89767 | 450486 |
| | other | 633960 | 2627020 |
| ParkScene | 0-1000 | 89254 | 150924 |
| | other | 632849 | 608869 |

図 4.2, 4.3, 4.4, 4.5 より, 画像の動きにばらつきがない場合でも複雑なブロックが選択されるので, 処理が膨大となってしまうことが予想される. 分散値の 0 から 100 までを 10 刻みで, 100 から 1000 までを 100 刻みで分けて調査したが, 1000 以上の動きベクトルが 2 倍以上存在することが表 4.2 から分かる.

また, 表 4.2 に JEM での分割が試行されている割合を近傍ブロックの動きベクトルの分散別に示す. QP=32 で BasketballDrill を 5 フレーム符号化した. 4 分割 (QT Split) と 2 分割について調査し, 2 分割は水平分割 (BT Split Hor) と垂直分割 (BT Split Ver) の 2 種類がある.

表 4.2: 動きベクトルの分散値別分割種類選択率

| | QT Split | BT Split Hor | BT split Ver |
|-------|----------|--------------|--------------|
| 100 | 0.72% | 79.92% | 79.58% |
| 1000 | 0.56% | 91.79% | 90.12% |
| 2000 | 1.25% | 93.45% | 85.29% |
| 3000 | 0.00% | 100.00% | 100.00% |
| 4000 | 0.64% | 85.94% | 82.61% |
| 5000 | 0.64% | 100.00% | 100.00% |
| other | 0.65% | 83.26% | 83.18% |
| Sum | 0.71% | 81.12% | 19.91% |

表 4.2 より, QT(Quad Tree) よりも BT(Binary Tree) を選択する傾向がある. また, BT の中でも水平分割が多く選択されていることが分かる.

4.2 輝度値との関連

H.265/HEVC では, 対象 CU 内のテクスチャが一様であったり, テクスチャの複雑性が CU 内でばらつきがない場合には, CU 内の動きも一様であると考えられ, 分割されないブロックが選択される傾向がある [6]. FVC でも同様に輝度値に関連した分割パターンがあるか調査を行った. 図 4.6, 4.7, 4.8, 4.9 に対象 CU 内の輝度値の分散値に対する $2N \times 2N$ サイズの選択率を示す. なお, 4.1 章と同様に, Random Access 符号化条件を選択し, QP=32 で BasketballDrill, BQMall, Kimono, ParkScene を 25 フレームを符号化した.

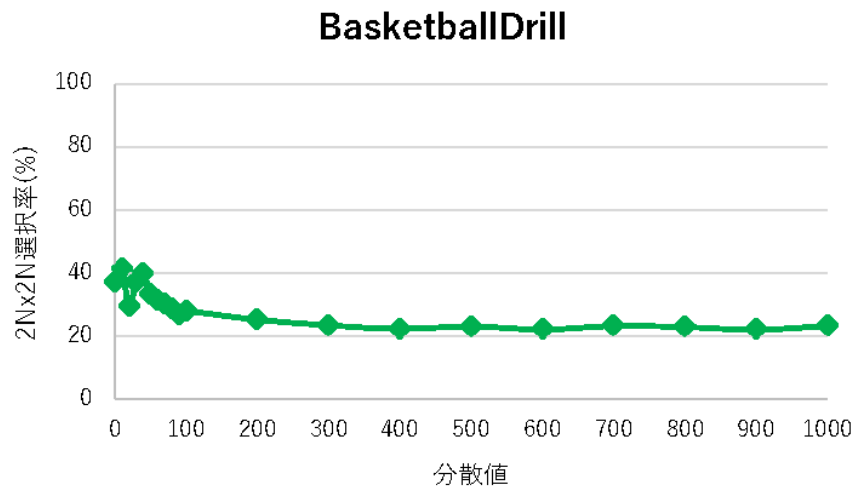


図 4.6: 輝度値分散値別の 2Nx2N 選択率 1

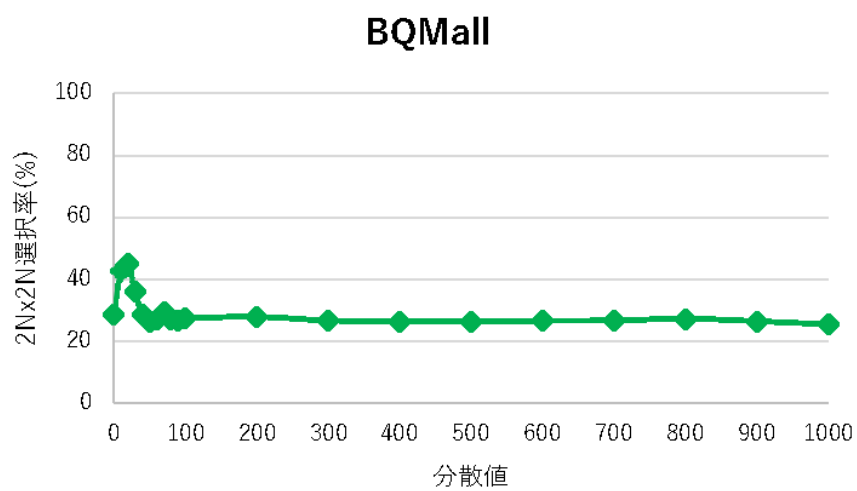


図 4.7: 輝度値分散値別の 2Nx2N 選択率 2

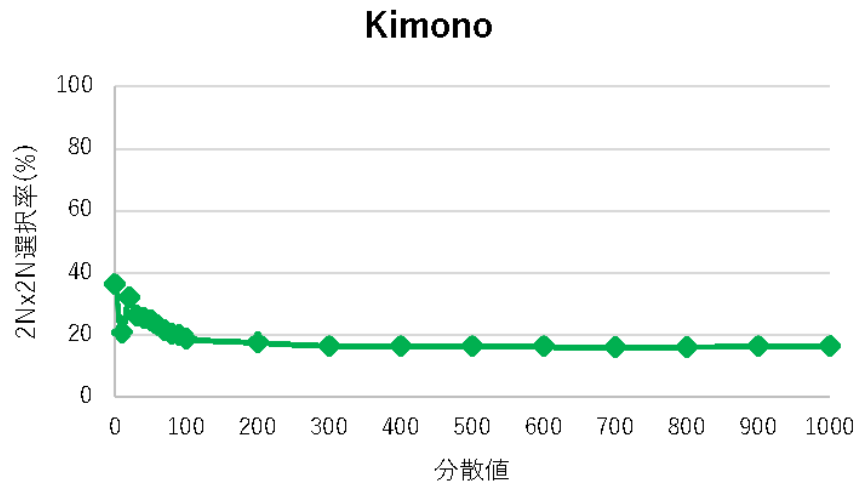


図 4.8: 輝度値分散値別の 2Nx2N 選択率 3

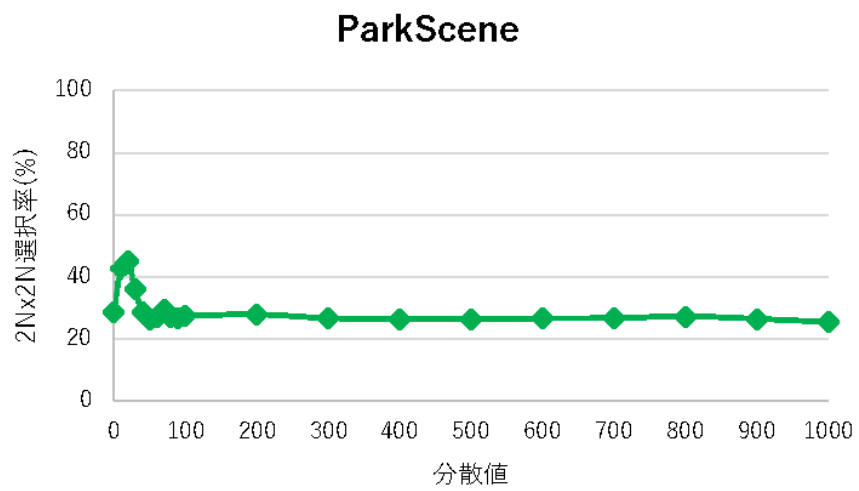


図 4.9: 輝度値分散値別の 2Nx2N 選択率 4

図 4.6, 4.7, 4.8, 4.9 より画像の様が一様な場合でも複雑なブロックが選択されるので, 処理が膨大となってしまうことが予想される。

また, 表 4.3 に JEM での分割が試行されている割合を対象ブロックの輝度値分散別に示す。QP=32 で BasketballDrill を 5 フレーム符号化した。動きベクトルと同様に, QT Split, BT split Hor, BT split Ver の 3 種類について調査した。

表 4.3: 輝度値の分散値別分割種類選択率

| | QT Split | BT Split Hor | BT split Ver |
|-------|----------|--------------|--------------|
| 100 | 0.01% | 11.52% | 11.01% |
| 1000 | 0.26% | 23.33% | 23.25% |
| 2000 | 0.76% | 27.91% | 27.86% |
| 3000 | 0.95% | 28.93% | 27.41% |
| 4000 | 1.10% | 29.16% | 28.34% |
| 5000 | 1.08% | 29.34% | 29.37% |
| other | 1.69% | 31.31% | 31.35% |
| Sum | 0.73% | 25.39% | 12.65% |

動きベクトルと同様に, 全体に QT よりも BT を選択する傾向がある。また, 分散値は動きベクトルの分散値と異なり, 大きな値まで広がっていることが分かる。

5 提案手法

5.1 近傍 CTU の動きベクトル分散

対象 CTU 内における四分木分割及び二分木分割は，対象 CTU 内の被写体の動きの複雑性と相関性が高いと考えられる．そこで，対象 CTU の左，左上，上，右上の符号化済み 4 近傍 CTU の検出済み動きベクトルの分散により被写体の動きの複雑性を推定する．対象 CTU を処理する際，近傍 CTU を構成している CU の動きベクトルの分散を，被写体の動きの複雑性を表す指標として計算する．その計算は，次の (1) 式により行う．ただし， n は動きベクトルの本数， ave は動きベクトルの平均値である．

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

図 5.10 は限定する場合の例を示す．

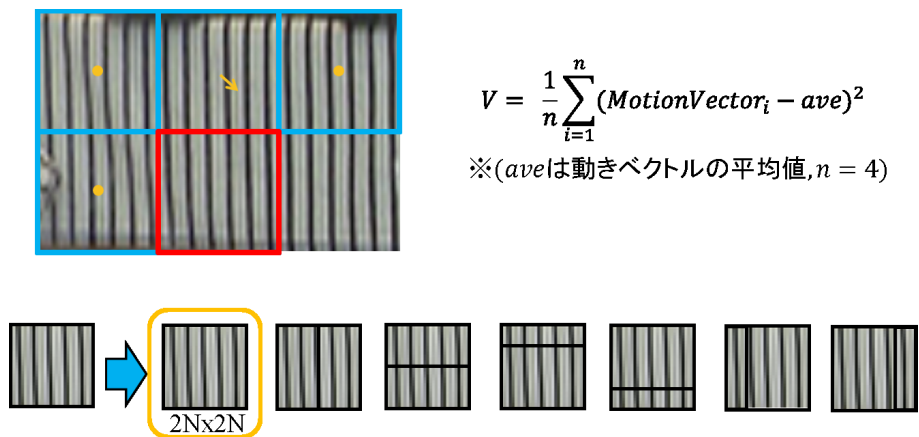


図 5.10: 動きベクトル分散の 2Nx2N 限定

3.1 章で明らかにしたように，QT よりも BT が選択される傾向，かつ全体的に分散値が小さいので BT 分割の閾値を 1000 にし，QT 分割の閾値は 50 に設定した．この値は，実験的に試した際に，閾値が高くて低くてもエンコード時間が低減できないためである．

5.2 対象 CTU の輝度値分散

輝度値の分散特徴として、対象 CTU 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) を計算する。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。

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

図 5.11 は限定する場合の例を示す。

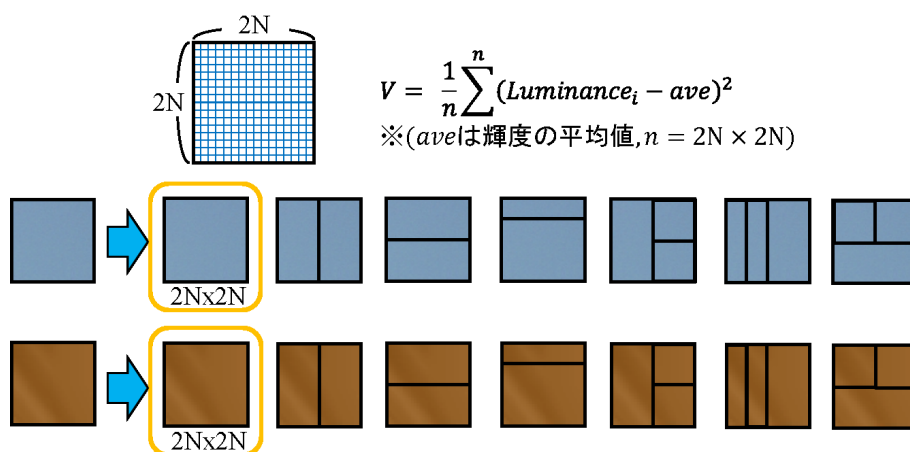


図 5.11: 輝度値分散の $2N \times 2N$ 限定

3.2 章で明らかにしたように、QT よりも BT が選択される傾向がある。しかし、分散値によって偏りがあまりないので、総数の 5 割程度削減するように BT 分割の閾値を 5000、QT 分割の閾値を 500 に設定した。動きベクトル手法と同様に、実験的に試した際に、BD-rate の悪化とエンコード時間の低減のバランスを取ったため、この値に設定した。

5.3 提案手法の流れ

4.1, 4.2 章より実装した手法の流れを説明する.

Step1) 動きベクトルの分散と輝度値の分散を計算する.

Step2) 2分割での閾値判定.

Step3) 動きベクトルの分散が小さい場合, 動きベクトルを用い, 閾値以下なら水平分割と垂直分割の試行をやめる.

Step4) 動きベクトル分散が大きい場合, 輝度値の分散を用い, 閾値以下なら水平分割と垂直分割の試行をやめる.

Step5) 4分割での閾値判定.

Step6) 動きベクトルの分散が小さい場合, 動きベクトルを用い, 閾値以下なら4分割の試行をやめる.

Step7) 動きベクトル分散が大きい場合, 輝度値の分散を用い, 閾値以下なら4分割の試行をやめる.

6 評価

6.1 実装方法

提案手法では, 2つの分散値を求める. 分散値の求め方は, 従来のHMで使用していた手法 [6][7] と同様であるが, JEMに組み込むために変更を加えた. HMでは近傍のブロックを参照, または輝度値を参照する際にCUを用いていたが, JEMではCTUを用いて処理を行う.

6.2 評価条件

FVCの参照ソフトウェアJEM-5.0に提案手法を実装し, その性能を確認した. 表6.4に用いたテストシーケンス, 表6.5に符号化条件, 表6.6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), 予測処理で用いられる画面内予測のイントラ予測処理時間IntraT(Intra prediction Time)とフレーム間予測であるインター予

測処理時間 (Inter prediction Time) を取得し, BD-rate と式 (3)(4)(5) を用いてエンコード時間低減率を表す ETRR(Encoding Time Reduction Ratio), イントラ予測処理時間低減率を表す IntraTRR(Intra prediction Time Reduction Ratio), インター予測処理時間低減率を表す InterTRR(Inter prediction Time Reduction Ratio) も算出した.

$$ETRR = \frac{ET_{JEM-5.0} - ET_{proposed}}{ET_{JEM-5.0}} \times 100 \quad (3)$$

$$IntraTRR = \frac{IntraT_{JEM-5.0} - IntraT_{proposed}}{IntraT_{JEM-5.0}} \times 100 \quad (4)$$

$$InterTRR = \frac{InterT_{JEM-5.0} - InterT_{proposed}}{InterT_{JEM-5.0}} \times 100 \quad (5)$$

表 6.4: 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 6.5: 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 6.6: 実行環境

| | |
|-----|-----------------------------------|
| OS | Windows Server 2012 |
| CPU | Intel Xeon CPU E3-1246 V3 3.50GHz |
| RAM | 16.0 GB |

6.3 評価結果

評価結果を表 6.7, 6.8, 6.9, 6.10 に示す. それぞれ, 動きベクトル手法, 輝度値手法, 動きベクトル輝度値組み合わせ手法となっている.

表 6.7: Random Access における評価結果 1

| Sequence | QP | JEM | | | | | | 動きベクトル手法 | | | | | |
|-----------------|----|-------------------|----------------|-------------|-----------------|-----------------|-------------------|----------------|-------------|-----------------|-----------------|--|--|
| | | Bitrate (kbps) | Y-PSNR (dB) | ET (sec) | IntraT (sec) | InterT (sec) | Bitrate (kbps) | Y-PSNR (dB) | ET (sec) | IntraT (sec) | InterT (sec) | | |
| BasketballDrill | 22 | 3045.65 | 41.62 | 89931.74 | 19508.54 | 28794.44 | 3127.68 | 41.49 | 58300.34 | 10452.27 | 18895.59 | | |
| | 27 | 1423.69 | 38.42 | 70692.75 | 14176.32 | 23000.46 | 1455.47 | 38.29 | 49561.47 | 8221.55 | 16529.26 | | |
| | 32 | 714.87 | 35.46 | 57295.36 | 10312.68 | 19099.21 | 726.99 | 35.35 | 41491.57 | 6166.15 | 14157.00 | | |
| | 37 | 365.13 | 32.81 | 46145.51 | 6717.58 | 15926.21 | 371.89 | 32.71 | 35906.99 | 4416.46 | 12664.09 | | |
| BQMall | 22 | 3136.60 | 40.72 | 94419.16 | 18638.49 | 27529.09 | 3207.57 | 40.65 | 72868.11 | 12024.75 | 22607.25 | | |
| | 27 | 1446.72 | 38.36 | 68975.13 | 13073.08 | 20346.99 | 1481.76 | 38.26 | 57736.79 | 8874.81 | 18245.88 | | |
| | 32 | 747.41 | 35.79 | 56422.12 | 9606.55 | 17324.67 | 764.67 | 35.66 | 47082.64 | 6420.72 | 15096.68 | | |
| | 37 | 395.25 | 33.08 | 45734.77 | 6896.67 | 14104.21 | 402.40 | 32.93 | 37490.21 | 4359.23 | 12216.22 | | |
| Kimono | 22 | 4169.90 | 41.97 | 202923.67 | 39904.01 | 60676.41 | 4173.71 | 41.95 | 168129.36 | 25956.96 | 53862.47 | | |
| | 27 | 1758.42 | 40.09 | 140711.19 | 22837.81 | 44175.44 | 1767.38 | 40.06 | 119068.50 | 15048.88 | 39481.28 | | |
| | 32 | 852.76 | 37.84 | 107444.69 | 15585.36 | 35054.56 | 857.55 | 37.81 | 89472.00 | 9609.71 | 30649.44 | | |
| | 37 | 416.25 | 35.39 | 83565.26 | 10217.50 | 27945.26 | 417.08 | 35.34 | 70048.02 | 6084.16 | 24775.90 | | |
| ParkScene | 22 | 6432.77 | 40.48 | 162612.49 | 35205.40 | 38409.73 | 6465.12 | 40.42 | 116518.07 | 18648.92 | 30502.15 | | |
| | 27 | 2683.74 | 38.04 | 113429.84 | 22744.68 | 28010.78 | 2698.35 | 37.97 | 88101.02 | 12478.57 | 23825.08 | | |
| | 32 | 1237.71 | 35.56 | 88439.30 | 16215.18 | 22878.89 | 1240.11 | 35.48 | 64551.16 | 8109.05 | 17975.94 | | |
| | 37 | 573.43 | 33.11 | 67270.35 | 10873.80 | 18095.28 | 573.29 | 33.02 | 49595.18 | 5278.78 | 14280.18 | | |

表 6.8: Random Access における評価結果 2

| Sequence | QP | JEM | | | | | | 輝度値手法 | | | | | |
|-----------------|----|-------------------|----------------|-------------|-----------------|-----------------|-------------------|----------------|-------------|-----------------|-----------------|--|--|
| | | Bitrate (kbps) | Y-PSNR (dB) | ET (sec) | IntraT (sec) | InterT (sec) | Bitrate (kbps) | Y-PSNR (dB) | ET (sec) | IntraT (sec) | InterT (sec) | | |
| BasketballDrill | 22 | 3045.65 | 41.62 | 89931.74 | 19508.54 | 28794.44 | 3108.55 | 41.45 | 51372.93 | 9685.56 | 16767.16 | | |
| | 27 | 1423.69 | 38.42 | 70692.75 | 14176.32 | 23000.46 | 1444.88 | 38.27 | 46101.16 | 8091.29 | 15491.11 | | |
| | 32 | 714.87 | 35.46 | 57295.36 | 10312.68 | 19099.21 | 719.47 | 35.33 | 39301.11 | 6400.88 | 13477.44 | | |
| | 37 | 365.13 | 32.81 | 46145.51 | 6717.58 | 15926.21 | 367.95 | 32.72 | 34525.90 | 4736.14 | 12512.34 | | |
| BQMall | 22 | 3136.60 | 40.72 | 94419.16 | 18638.49 | 27529.09 | 3192.54 | 40.62 | 62360.77 | 10166.39 | 18507.55 | | |
| | 27 | 1446.72 | 38.36 | 68975.13 | 13073.08 | 20346.99 | 1467.47 | 38.23 | 52699.39 | 8388.95 | 16016.74 | | |
| | 32 | 747.41 | 35.79 | 56422.12 | 9606.55 | 17324.67 | 754.49 | 35.66 | 45620.76 | 6846.93 | 14112.61 | | |
| | 37 | 395.25 | 33.08 | 45734.77 | 6896.67 | 14104.21 | 396.89 | 32.96 | 37595.39 | 5158.44 | 11898.85 | | |
| Kimono | 22 | 4169.90 | 41.97 | 202923.67 | 39904.01 | 60676.41 | 4231.70 | 41.90 | 68105.30 | 10182.45 | 21461.48 | | |
| | 27 | 1758.42 | 40.09 | 140711.19 | 22837.81 | 44175.44 | 1794.45 | 39.99 | 59320.79 | 7250.25 | 19552.06 | | |
| | 32 | 852.76 | 37.84 | 107444.69 | 15585.36 | 35054.56 | 868.25 | 37.73 | 52373.55 | 5435.61 | 17796.69 | | |
| | 37 | 416.25 | 35.39 | 83565.26 | 10217.50 | 27945.26 | 420.47 | 35.26 | 43087.29 | 3839.99 | 14914.75 | | |
| ParkScene | 22 | 6432.77 | 40.48 | 162612.49 | 35205.40 | 38409.73 | 6514.19 | 40.33 | 74531.76 | 11672.68 | 18952.12 | | |
| | 27 | 2683.74 | 38.04 | 113429.84 | 22744.68 | 28010.78 | 2709.78 | 37.88 | 64011.12 | 9173.69 | 17221.50 | | |
| | 32 | 1237.71 | 35.56 | 88439.30 | 16215.18 | 22878.89 | 1240.35 | 35.41 | 54229.27 | 7057.09 | 15120.09 | | |
| | 37 | 573.43 | 33.11 | 67270.35 | 10873.80 | 18095.28 | 573.21 | 32.98 | 42867.77 | 5138.80 | 12219.49 | | |

表 6.9: Random Access における評価結果 3

| Sequence | QP | JEM | | | | | | 提案手法 (動きベクトル+輝度値) | | | | | |
|-----------------|----|----------------|-------------|-----------|--------------|--------------|----------------|-------------------|-----------|--------------|--------------|--|--|
| | | Bitrate (kbps) | Y-PSNR (dB) | ET (sec) | IntraT (sec) | InterT (sec) | Bitrate (kbps) | Y-PSNR (dB) | ET (sec) | IntraT (sec) | InterT (sec) | | |
| BasketballDrill | 22 | 3045.65 | 41.62 | 89931.74 | 19508.54 | 28794.44 | 3133.41 | 41.47 | 50072.92 | 9462.43 | 15814.60 | | |
| | 27 | 1423.69 | 38.42 | 70692.75 | 14176.32 | 23000.46 | 1458.76 | 38.27 | 42362.62 | 7406.49 | 13836.91 | | |
| | 32 | 714.87 | 35.46 | 57295.36 | 10312.68 | 19099.21 | 728.26 | 35.33 | 35837.12 | 5638.83 | 12108.72 | | |
| | 37 | 365.13 | 32.81 | 46145.51 | 6717.58 | 15926.21 | 372.63 | 32.70 | 30828.27 | 4086.85 | 10701.77 | | |
| BQMall | 22 | 3136.60 | 40.72 | 94419.16 | 18638.49 | 27529.09 | 3226.51 | 40.61 | 72654.49 | 12246.43 | 22195.82 | | |
| | 27 | 1446.72 | 38.36 | 68975.13 | 13073.08 | 20346.99 | 1489.28 | 38.22 | 46056.21 | 7297.41 | 14242.99 | | |
| | 32 | 747.41 | 35.79 | 56422.12 | 9606.55 | 17324.67 | 766.35 | 35.63 | 48434.94 | 6755.32 | 15460.93 | | |
| | 37 | 395.25 | 33.08 | 45734.77 | 6896.67 | 14104.21 | 403.59 | 32.91 | 36726.45 | 4468.89 | 11968.32 | | |
| Kimono | 22 | 4169.90 | 41.97 | 202923.67 | 39904.01 | 60676.41 | 4217.97 | 41.92 | 103084.18 | 16964.82 | 33006.06 | | |
| | 27 | 1758.42 | 40.09 | 140711.19 | 22837.81 | 44175.44 | 1789.41 | 40.01 | 84318.63 | 11173.75 | 27790.74 | | |
| | 32 | 852.76 | 37.84 | 107444.69 | 15585.36 | 35054.56 | 868.38 | 37.75 | 71832.50 | 7949.73 | 24482.81 | | |
| | 37 | 416.25 | 35.39 | 83565.26 | 10217.50 | 27945.26 | 420.21 | 35.27 | 61572.53 | 5570.90 | 21422.95 | | |
| ParkScene | 22 | 6432.77 | 40.48 | 162612.49 | 35205.40 | 38409.73 | 6520.39 | 40.36 | 90581.55 | 15597.79 | 23010.60 | | |
| | 27 | 2683.74 | 38.04 | 113429.84 | 22744.68 | 28010.78 | 2716.35 | 37.90 | 75717.21 | 11299.30 | 20168.67 | | |
| | 32 | 1237.71 | 35.56 | 88439.30 | 16215.18 | 22878.89 | 1246.25 | 35.43 | 64564.47 | 8499.39 | 17916.31 | | |
| | 37 | 573.43 | 33.11 | 67270.35 | 10873.80 | 18095.28 | 575.26 | 32.97 | 53506.64 | 6027.77 | 15375.37 | | |

表 6.10: Random Access における評価結果 4

| | | 動きベクトル手法 | | | |
|-----------------|----|-------------------|---------|-------------|-------------|
| Sequence | QP | BD-rate(%) | ETRR(%) | IntraTRR(%) | InterTRR(%) |
| BasketballDrill | 22 | 5.1 | 35.17 | 46.42 | 34.38 |
| | 27 | | 29.89 | 42.01 | 28.14 |
| | 32 | | 27.58 | 40.21 | 25.88 |
| | 37 | | 22.19 | 34.26 | 20.48 |
| BQMall | 22 | 5.3 | 22.82 | 35.48 | 17.88 |
| | 27 | | 16.29 | 32.11 | 10.33 |
| | 32 | | 16.55 | 33.16 | 12.86 |
| | 37 | | 18.03 | 36.79 | 13.39 |
| Kimono | 22 | 1.5 | 17.15 | 34.95 | 11.23 |
| | 27 | | 15.38 | 34.11 | 10.63 |
| | 32 | | 16.73 | 38.34 | 12.57 |
| | 37 | | 16.18 | 40.45 | 11.34 |
| ParkScene | 22 | 2.8 | 28.35 | 47.03 | 20.59 |
| | 27 | | 22.33 | 45.14 | 14.94 |
| | 32 | | 27.01 | 49.99 | 21.43 |
| | 37 | | 26.27 | 51.45 | 21.08 |
| Ave | | 3.7 | 22.37 | 40.12 | 17.95 |
| | | 輝度値手法 | | | |
| Sequence | QP | BD-rate(%) | ETRR(%) | IntraTRR(%) | InterTRR(%) |
| BasketballDrill | 22 | 4.7 | 42.88 | 50.35 | 41.77 |
| | 27 | | 34.79 | 42.92 | 32.65 |
| | 32 | | 31.41 | 37.93 | 29.43 |
| | 37 | | 25.18 | 29.50 | 21.44 |
| BQMall | 22 | 4.6 | 33.95 | 45.45 | 32.77 |
| | 27 | | 23.60 | 35.83 | 21.28 |
| | 32 | | 19.14 | 28.73 | 18.54 |
| | 37 | | 17.80 | 25.20 | 15.64 |
| Kimono | 22 | 5.5 | 66.44 | 74.48 | 64.63 |
| | 27 | | 57.84 | 68.25 | 55.74 |
| | 32 | | 51.26 | 65.12 | 49.23 |
| | 37 | | 48.44 | 62.42 | 46.63 |
| ParkScene | 22 | 5.7 | 54.17 | 66.84 | 50.66 |
| | 27 | | 43.57 | 59.67 | 38.52 |
| | 32 | | 38.68 | 56.48 | 33.91 |
| | 37 | | 36.28 | 52.74 | 32.47 |
| Ave | | 5.1 | 39.08 | 50.12 | 36.58 |
| | | 提案手法 (動きベクトル+輝度値) | | | |
| Sequence | QP | BD-rate(%) | ETRR(%) | IntraTRR(%) | InterTRR(%) |
| BasketballDrill | 22 | 5.8 | 44.32 | 51.50 | 45.08 |
| | 27 | | 40.08 | 47.75 | 39.84 |
| | 32 | | 37.45 | 45.32 | 36.60 |
| | 37 | | 33.19 | 39.16 | 32.80 |
| BQMall | 22 | 6.7 | 23.05 | 34.29 | 19.37 |
| | 27 | | 33.23 | 44.18 | 30.00 |
| | 32 | | 14.16 | 29.68 | 10.76 |
| | 37 | | 19.70 | 35.20 | 15.14 |
| Kimono | 22 | 4.5 | 49.20 | 57.49 | 45.60 |
| | 27 | | 40.08 | 51.07 | 37.09 |
| | 32 | | 33.14 | 48.99 | 30.16 |
| | 37 | | 26.32 | 45.48 | 23.34 |
| ParkScene | 22 | 5.4 | 44.30 | 55.69 | 40.09 |
| | 27 | | 33.25 | 50.32 | 28.00 |
| | 32 | | 27.00 | 47.58 | 21.69 |
| | 37 | | 20.46 | 44.57 | 15.03 |
| Ave | | 4.6 | 32.43 | 45.52 | 29.41 |

表 6.7, 6.10 より, 動きベクトル手法は標準の JEM-5.0 と比べ, BD-rate の悪化を最大でも 5.3% に抑えつつ, エンコード時間を平均で 22% 低減することができた. また, 表 6.8, 6.10 より輝度値手法では, BD-rate の悪化を最大でも 5.7% に抑えつつ, エンコード時間を平均で 39% 低減することができた. 表 6.9, 6.10 より組み合わせ手法では, BD-rate の悪化を平均で 4.6% に抑えつつ, エンコード時間を平均で 32% 低減することができた. 全ての手法において, イントラ予測処理時間を約 40% 以上低減することができているので, イントラ予測処理の時間が膨大となっている FVC において, 非常に有効性があることが伺える.

6.4 考察

二つの手法での結果を H.265/HEVC での評価結果 [6][7] と比較してみると, BD-rate の悪化が見られ, 動きベクトル手法ではエンコード時間低減率が下がった. BD-rate の悪化は必要な分割まで限定してしまっているため, 必要な分割が省かれてしまっているからだと考えられる. エンコード時間低減率が下がるのは, FVC の場合, 必ずしも分散値が小さければ大きいブロックが選択されるとは限らないからだと考えられる. この二つの問題点を基に, 今後の課題と方針として検討していく.

7 あとがき

本研究では, 近傍の符号化済み CTU の動きベクトルの分散と対象 CTU の輝度値分散を用いて, 対象 CTU 内の動きを推定し, 適応的に候補 CU を限定することで BD-rate の悪化を平均で 4.6% に抑えつつ, エンコード時間を平均で 32.43% 低減することができた. しかし, 輝度値分散の手法では全てのシーケンスで BD-rate が 5% 程度の悪化が見られる. また, 動きベクトル分散では BasketballDrill と BQMall のシーケンスにおいて 5% 程度の悪化が見られる. この悪化は, 必要な分割が行われ難くなるためと考えられる. 今後は, 閾値の最適化を行いエンコード時間を低減できないか検討していきたい. また, 別の局所的特徴を組み合わせることで, 更なる性能向上へと繋げていきたい.

現在では, JEM の開発は終了し, 新しい映像符号化規格である Versatile Video Coding(VVC) の標準化が開始されている. VVC では新たな分割構造が採択され, 現時点で HEVC を大きく上回る符号化効率を実現しつつ

ある。この新たな分割構造に対しても、本手法が有効であるかを検討すると同時に、新規分割構造に適応した手法を考案することも必要となる。

謝辞

本論文の執筆にあたりご指導を頂いた若林哲史教授，近藤利夫教授，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] JVET Software Repository. [Online]. Available:https://jvet.lhi.fraunhofer.de/svn/svn_HMJEMSoftware/branches/HM-16.6-JEM-5.0-dev/
- [3] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, “ Probabilistic Decision Based Block Partitioning for Future Video Coding, ” IEEE Transactions on Image Processing, vol. 27, pp.1475-1486, 2017.
- [4] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, “ Effective Quadtree Plus Binary Tree Block Partition Decision for Future Video Coding, ” Data Compression Conference (DCC), 2017.
- [5] Ting-Lan Lin, Hui-Yu Jiang, Jing-Ya Huang, and Pao-Chi Chang, “ Fast Intra Coding Unit Partition Decision in H.266/FVC Based on Spatial Features, ” IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), 2018.
- [6] 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.

- [7] 飯田拓真, 近藤利夫, 佐々木敬泰, 深澤祐樹, “近傍 CU の動きベクトル分散を利用した PU インター予測モード高速決定法,” 信学技報, vol. 117, no. 330, IE2017-86, pp. 67-72, 2017.
- [8] 蔡豊宇, 古田溪, 藤田玄, “HEVC における機械学習を利用した Intra/Inter CU サイズ決定の高速化,” 信学技報, vol. 116, no. 464, IE2016-109, pp. 241-245, 2016.