

HEVC における色差直交変換ブロック分割に関する一検討 A Study on Chroma Transform Block Partitioning for HEVC

峯澤 彰†
Akira Minezawa

杉本 和夫†
Kazuo Sugimoto

関口 俊一†
Shun-ichi Sekiguchi

市ヶ谷敦郎‡
Atsuro Ichigaya

境田 慎一‡
Shinichi Sakaida

1. まえがき

現在, ISO/MPEG と ITU-T/VCEG は, 共同で次世代映像圧縮方式 HEVC (High Efficiency Video Coding) の規格化作業を進めている[1]. HEVC は MPEG-4 AVC/H.264[2]に対して約 2 倍の圧縮性能を目標としており, MPEG-4 AVC/H.264 と同様のハイブリッド符号化をベースに様々な新しい符号化ツールの検討が行われている. 2012 年 5 月現在, HEVC のテストモデルでは直交変換単位が符号化対象ブロックからの四分木分割によって表現され, 輝度信号と色差信号で同一の分割が適用されている. しかし, 一般に輝度信号と色差信号の信号特性は大きく異なるため, 最適なブロック分割状態はそれぞれ異なる可能性がある.

本稿では, 特に 4K×2K 映像やハイビジョン映像の 16 倍の空間解像度を持つスーパーハイビジョン映像といった高解像度映像 (UHD 映像) の符号化効率改善を目的として, 輝度信号とは独立した色差信号の直交変換ブロック分割手法について検討する.

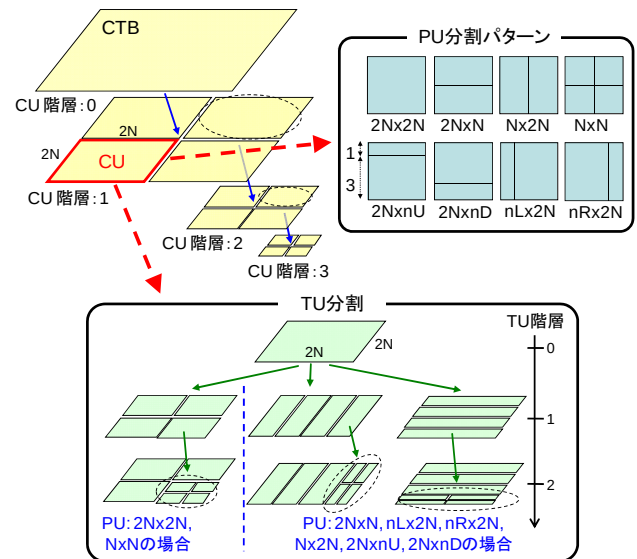


図1 HEVC のブロック分割

2. HEVC における適応ブロック分割手法

MPEG-4 AVC/H.264 に対する HEVC の特徴的な点として, 符号化, 予測, 変換のブロックサイズが拡張され, 予測・変換処理の画面内適応化の自由度が高まったことが挙げられる. 具体的には, HEVC では最大符号化ブロック (CTB) サイズ及び CTB を四分木分割することで表現される符号化ブロック (CU) サイズを自由に選択することができるため, MPEG-4 AVC/H.264 では 16×16 画素で固定であった符号化単位を柔軟に設定可能である. さらに HEVC は CU 単位に予測ブロック (PU) サイズ及び直交変換ブロック (TU) サイズを変更することができる. 図1に HEVC での CU, PU 及び TU 分割を示す. この図のように, PU 分割は CU に対して予め用意されている複数の固定の PU 分割パターンから選択することで決定する. 一方, TU 分割については, CU を起点とした四分木分割で表現される. また, HEVC は現在 YUV4:2:0 フォーマットのみに対応しており, CU, PU 及び TU の分割情報は輝度信号と色差信号で共通とし, TU の最大サイズは輝度信号で 32×32 画素, 色差信号で 16×16 画素に制限されている.

HEVC では, 各 CU で PU の分割状態によらずに TU を自由に決定できるため, PU 内で生じる予測残差の空間的な偏りが大きい場合において, 予測単位よりも細かい単位に TU を分割して直交変換を実施することで, 局所的な残差分布に応じた効率的な信号圧縮が期待できる. ただし, 一般に輝度信号に比べて色差信号は信号電力が非常に小さいことから, 輝度信号において PU よりも小さい TU 分割が

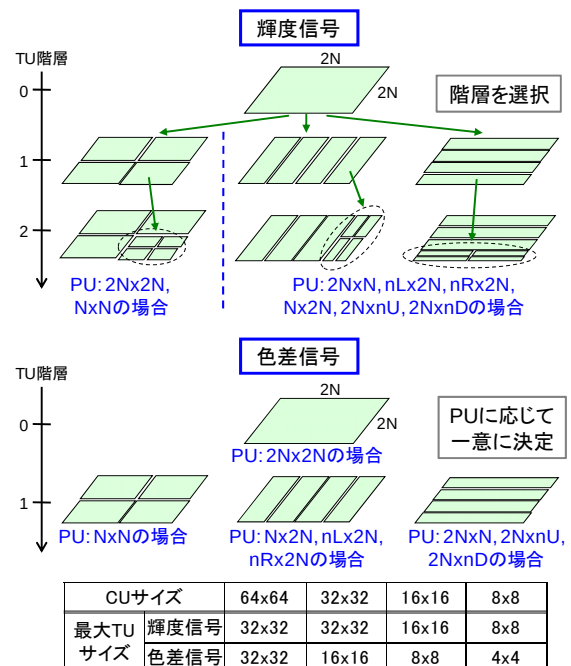


図2 提案 TU 分割手法

効率的である場合でも, 色差信号では過分割となり最適な分割でない可能性がある. 特に, 空間方向の相関の高い高解像度映像においては 32×32 画素や 64×64 画素といった大きいサイズの直交変換が効果的であることが知られており[3], 信号電力の小さい色差信号では分割し過ぎずに大きいサイズの直交変換を積極的に利用することで符号化効率の改善が期待できる.

† 三菱電機株式会社 情報技術総合研究所

‡ NHK 放送技術研究所

そこで本稿では、色差信号の最大直交変換サイズを 32×32 画素まで拡張可能とする、輝度信号と独立した色差信号の TU 分割手法について検討する。

3. 色差直交変換ブロック独立分割手法

図 2 に提案 TU 分割手法を示す。図 2 より、本手法では輝度信号の TU 分割状態に関わらず PU に応じて色差信号の TU 分割が一意に決定される。これにより、色差信号の TU 分割を行う際に必要な符号化パラメータが新たに追加されることなく、PU サイズを越えるような TU の過分割を抑制することができる。また、本手法では色差信号の PU サイズが 32×32 画素の際に同じサイズの直交変換を使用するため、従来の色差信号での最大の直交変換 (16×16 画素) よりも大きいサイズの直交変換を利用することが可能となる。なお、 32×32 画素の直交変換については従来から輝度信号で用いられているため、提案手法による新たな直交変換処理の実装は必要としない。

4. シミュレーション結果

提案方式の符号化効率を評価するため、HEVC のテストモデル第 4 版 (HM4.0) [4] に対して提案方式を実装したソフトウェアを用い、符号化シミュレーションを実施した。

本シミュレーションにおける共通符号化条件を表 1 に示す。表 1 の条件の下、解像度の異なる複数のシーケンスを用いてシミュレーションを行った際の HM4.0 に対する提案方式の符号化性能を表 2 に示す。ただし、表中の $\Delta \text{bitrate}$ は文献[5]の平均符号量変化量を示す評価値であり、値が負である場合符号量が削減されていることを示している。表 2 より、HM4.0 と比較して提案方式は平均で Y 信号で 0.1%、U 信号及び V 信号で 2.1% 符号化効率が改善していることが確認できる。解像度別に確認すると、特に 1920×1080 以上の高解像度映像において高い符号化効率を有しており、色差信号の符号化効率が最大で約 16% 改善されていることがわかる。本結果より、提案方式によって色差信号の最大直交変換サイズを拡張しつつ TU の過分割を抑制することの優位性が確認できる。ただし、シーケンス単位に確認すると、特に 416×240 画素といった低解像度のシーケンスを中心に僅かに符号化効率の低下が生じている。これは、低解像度では色差信号での 32×32 画素の直交変換が効果的でない上に、輝度信号と色差信号の変換サイズを独立にすることによる性能改善よりも色差信号の直交変換サイズを PU によって固定化することによる性能劣化が大きいためと考えられる。

なお、演算負荷については、シミュレーションを実行した際の CPU 時間を確認することで行った。表 3 に HM4.0

表 1 共通符号化条件

GOP 構造	階層 B ピクチャ構造
エントロピー符号化	CABAC
CU サイズ	$64 \times 64 \sim 8 \times 8$
PU サイズ	$64 \times 64 \sim 4 \times 4$
TU サイズ	$32 \times 32 \sim 4 \times 4$
RD 最適化	有り
QP	22, 27, 32, 37

に対する提案方式の CPU 時間比率を示す。これより、提案方式では HM4.0 と比較して約 3% 符号化時間が削減されていることがわかる。これは、提案方式では色差信号の TU 分割は PU に応じて一意に決まっており、輝度信号のみを考慮した TU 分割の選択が行えるためである。

5. まとめ

本稿では、次世代映像圧縮方式 HEVC における高解像度映像の符号化効率改善の一検討として、輝度信号と独立した色差信号の直交変換ブロック分割手法について検討を行った。その結果、提案方式は平均で Y 信号で 0.1%、U 信号及び V 信号で 2.1% の符号化効率改善を達成した。特に高解像度映像では最大 16% の色差信号の符号化効率改善効果が得られており、本方式の有効性が確認された。

参考文献

- [1] 関口, 浅井, “次世代映像符号化 (HEVC) の標準化動向”, 映像学誌 vol65, pp.70-76, March 2011.
- [2] ISO/IEC 14496-10 | ITU-T Recommendation H.264, March 2005.
- [3] 関口 他, “SHV 符号化における動き予測・変換ブロックサイズの効果”, 信学総大, DS-2-1, pp.S.1-S.2, March 2011.
- [4] B. Bross, et al., “WD4: Working Draft 4 of High-Efficiency Video Coding,” JCTVC-F803, July 2011.
- [5] G. Bjontegaard, “Calculation of average PSNR differences between RD-Curves,” VCEG-M33, April 2001.

表 2 シミュレーション結果

Sequence	Size [pel]	$\Delta \text{bitrate-Y} [\%]$	$\Delta \text{bitrate-U} [\%]$	$\Delta \text{bitrate-V} [\%]$
Traffic	2560 x1600	-0.2	-1.0	-1.0
PeopleOnStreet		0.0	-1.8	-2.9
NebutaFestival		0.1	-6.4	-3.6
SteamLocomotive		0.0	-15.8	-16.6
Kimono	1920 x1080	-0.2	-1.4	-0.7
ParkScene		-0.2	-1.1	-0.5
Cactus		-0.1	-1.7	-1.5
BasketballDrive		0.0	-4.7	-3.5
BQTerrace		-0.1	-1.4	-3.9
BasketballDrill	832 x480	0.0	-0.4	-0.4
BQMall		0.0	-1.7	-1.4
PartyScene		-0.1	0.1	0.3
RaceHorses		-0.2	0.3	0.0
BasketballPass	416 x240	0.0	-0.1	-0.1
BQSquare		0.0	0.2	-0.2
BlowingBubbles		0.0	0.2	0.0
RaceHorses		0.0	0.1	-0.1
2560x1600		0.0	-6.3	-6.0
1920x1080		-0.1	-2.0	-2.0
832x480		-0.1	-0.4	-0.4
416x240		0.0	0.1	-0.1
Average		-0.1	-2.1	-2.1

表 3 HM4.0 に対する提案方式の CPU 時間比率 [%]

符号化時間	96.8
復号時間	100.4