

VVC 対応の 4K リアルタイムソフトウェアデコーダの試作開発

木谷 佳隆 河村 圭

KDDI 総合研究所 〒356-0056 埼玉県ふじみ野市大原 2-1-15

E-mail: yo-kidani@kddi-research.jp

あらまし HEVC を超える最新の動画像符号化方式として、まもなく国際標準化される VVC (Versatile Video Coding) は、HEVC と比較して客観画質で約 30% の符号化性能の改善を達成している。この性能改善は、VVC に新たに採用された多様なブロック分割構造や Intra/Inter 予測や変換、ループ内フィルタの拡張によってもたらされている。しかし、これらの新規ツールは符号化および復号処理時間を大幅に増加させており、VVC の実用化を見据えた場合、これら処理の高速化が必須である。本稿では、VVC の特徴を活かして高速化を行い、4K リアルタイムソフトウェアデコーダの試作開発結果を報告する。

キーワード Versatile video coding, 4K 映像, リアルタイムデコーダ, ソフトウェアデコーダ.

Prototype Development of 4K Real-time Software Decoder for VVC

Yoshitaka KIDANI Kei KAWAMURA

KDDI Research, Inc. 2-1-15 Ohara, Fujimino-shi, Saitama, 356-8502 Japan.

E-mail: yo-kidani@kddi-research.jp

Abstract Versatile Video Coding (VVC), which will be soon international standardized as the state-of-the-art video coding beyond HEVC, achieves about 30% coding performance improvement in objective image quality compared to that by HEVC. This coding performance realizes by new coding tools adopted to VVC, which are various block partitioning structure and some extensions of intra-/inter- predictions, large transform, and additional in-loop filter. These new tools, however, increase the encoding and decoding run-time significantly. The acceleration of their processing is essential to the practical use case of VVC. In this manuscript, we report the feasibility via prototype development of 4K real-time software decoder of VVC.

Keyword Versatile video coding, 4K video, real-time decoder, software decoder.

1. はじめに

最新の動画像符号化方式 VVC (Versatile Video Coding) が JVET (Joint Video Expert Team) で議論され、2020 年 7 月会合で国際標準化作業が完了した[1].

本稿では、VVC の適用先として 4K ライブ映像のネットワーク伝送を想定して、VVC で符号化された 4K/60fps 映像をネットワーク越しに、リアルタイムで受信および復号可能なソフトウェアデコーダ (以下、デコーダ) を試作開発した結果を報告する。

2. 参照ソフトウェアとデコーダの要求条件

VVC の標準化にあたり、参照ソフトウェア VTM (VVC Test Mode) が作成された[2]. VTM では客観品質における最高性能を引き出すことを念頭に設計されている。具体的には、HEVC (High Efficiency Video Coding)[3]の参照ソフトウェア HM(HEVC Test Model)[4]と比較して、VTM は約 30% の符号化性能の改善を達成している一方で、符号化時間が約 10 倍、復号時間が 2 倍に増加している[5]. 実際の復号時間は、対象映像の映像フォーマット (解像度、フレームレート、ビット深度、カラーフォーマット) や符号化ビットレ

ト、および符号化難易度に依存する。たとえば JVET の共通実験条件 (以下、JVET-CTC) [6]にある試験映像素材のひとつである DaylightRoad2 (3840 画素 2160 ライン, 60fps, 合計 300 フレーム, 5 秒) を、VTM-9 を用いて復号する際の処理時間は、最新の CPU を用いたシングルスレッド実行における符号化レート約 10Mbps の条件で 156.33 秒要する。したがって、本試験映像をリアルタイムすなわち 60fps で映像出力できるように復号するには、VTM-9 の復号処理を 30 倍以上高速化する必要がある。

VTM の復号処理を高速化するアプローチとして、VVC の特徴を利用した並列分散処理が考えられる。具体的には、参照関係が存在しないピクチャ単位の並列処理 (以下、ピクチャ並列) やピクチャを任意の形状に分割したスライス単位の並列処理 (以下、スライス並列) がある。ここで、ピクチャ間の参照関係の有無は、ピクチャタイプやピクチャの参照構造を示すパラメータによって識別可能であり、たとえば JVET-CTC の RA (Random Access) 条件下では I ピクチャには参照関係がないため、I ピクチャは独立した処理が可能

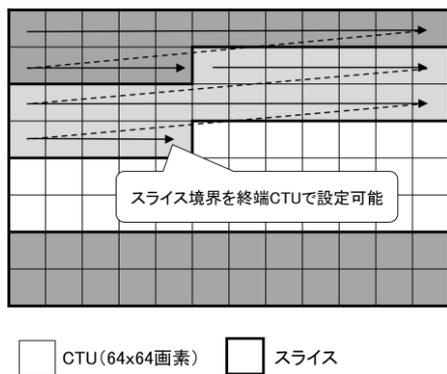


図 1 HEVC のスライス分割

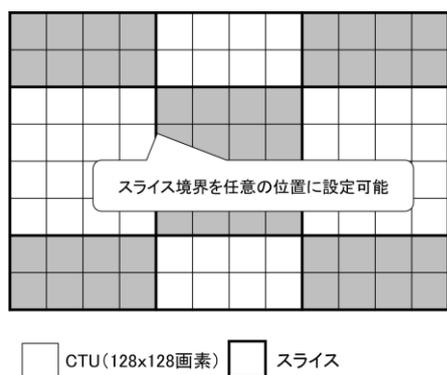


図 2 VVC のスライス分割

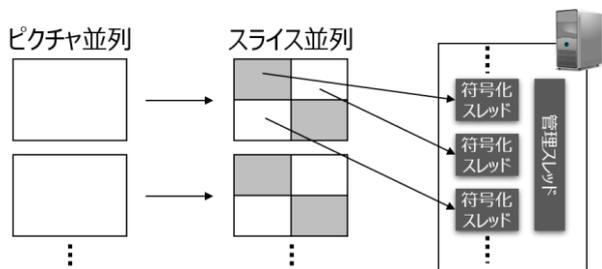


図 3 ピクチャ並列とスライス並列による分散処理

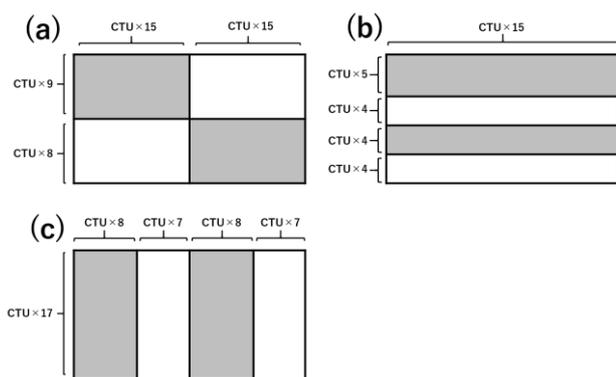


図 4 4K ピクチャをスライス 4 分割した分割形状例

である．一方，B ピクチャには参照関係が存在するため，参照先ピクチャの処理が完了しなければ，処理が開始できない．

スライス分割は，AVC (Advanced Video Coding) [7] のほか HEVC や VVC において仕様化されている．特に，HEVC ではスライス境界が図 1 のように終端 CTU

(Coding Tree Unit) で設定できる一方で，VVC ではスライス境界が図 2 のように任意の位置で設定できる．なお，VVC における CTU サイズは，HEVC の CTU サイズ (64x64 画素) から 4 倍 (128x128 画素) に拡大している．

また，VVC には AVC に具備されていた，ネットワーク伝送時に発生しうるパケットロスに対するエラー耐性強化ツールが入っておらず，HEVC と同様にエラーフリーを前提に仕様化されている．したがって，ネットワーク伝送にも対応したデコーダを実現する場合は，VVC の仕様の外側でエラー耐性を強化する必要がある．

3. 報告するデコーダ構成

3.1. 並列化方式

報告するデコーダは，VVC の特徴を利用したピクチャ並列とスライス並列をマルチスレッド処理で実現する．具体的には，図 3 に示すように，符号化処理用スレッドには，各スライス単位での処理が動的に割り当てられる．これにより，スライスタイプの違いによる処理負荷の偏在化が生じていても，低負荷なスレッドには処理可能なスライスが順次割り当てられるため，全体の処理速度が高負荷なスレッドに律速する状況が回避される．

さらに，4K 解像度のピクチャをスライス分割する際の形状は，図 4a のように田の字の長方形スライス 4 つに設定した．このスライス分割数および分割形状の設定理由は以下の通りである．まず，スライス分割数が 4 つである理由は，分割数を必要最小限に留めて，符号化性能の低下を抑制するためである．次に，スライス 4 分割の設定方法は，図 4a 以外に，たとえば図 4b や図 4c のような横長や縦長のスライス分割形状が考えられる．ここで，図 4a の分割形状を選択する理由は，カメラで撮影された自然映像に対する各スライスの処理負荷の平準化と汎化性能の向上である．一般に，自然映像では空のような背景は画面上部に配置されやすく，人物のような前景は画面下部に配置されやすい．そのため，自然映像が図 4b のように横長でスライス分割されると，画面下部のスライスに処理負荷が偏在化しやすい．また，自然映像ではカメラパンが上下方向よりも左右方向に起こりやすい．そのため，図 4c のような縦長スライス分割は横方向の参照範囲が狭いため，予測性能が低下しやすい．図 4a は，上記観点で最適なスライス分割形状と考えられる．

3.2. エラー耐性強化

報告するデコーダは，VTM をベースに開発したが，VTM には符号化ビットストリームを多重化してネットワーク伝送する機能がない．そのため RTP パケット

のペイロードに NAL ユニットの格納して送受信するソフトを開発した。RTP を選択した理由は、ライブ映像のネットワーク伝送を適用先と想定する場合、TCP のように再送制御がなく遅延が少ないためである。一方で、RTP によるデータ転送はパケットロスの回復機能がなく、さらに VVC には、エラー耐性機能がないため、Pro-MPEG FEC COP3[8]準拠した FEC (Forward Error Correction) 機能をデコーダに実装した。

4. 実験と考察

4.1. 復号処理速度性能評価

報告するデコーダ構成を、VTM-9.3 をベースに実装し、復号処理速度をいくつかの条件下で測定した。図 5 に同デコーダの概観を示す。目標とする 4K/60fps のリアルタイム復号処理の達成可否を確認するため、著者らが所有する異なる計算機 (CPU 構成) で性能を評価した。具体的には、Xeon Gold 6254 が 2 基搭載の PC (以下、XG 6254) と Thread Ripper 3970X が 1 基搭載の PC (以下、TR 3970X) である。表 1 に、それぞれの CPU のクロック数と総スレッド数を示す。評価対象の 4K 映像については、JVET-CTC の 4K/60fps の試験素材 4 点を使用した。評価実験では、あらかじめ符号化したビットストリームを RTP 配信サーバから、デコーダに対して送信した。このときの符号化ビットレートは、国内で現在サービス提供されている HEVC を用いた 4K/60fps 映像配信サービスのビットレートが約 20Mbps であることを考慮して、その半分の約 10Mbps とした。

表 2 に、XG 6254 と TR 3970 の復号処理速度の測定結果を示す。それぞれ Intra Period (RA 条件におけるピクチャ参照構造上の I ピクチャ挿入間隔) は 32 とし、DPB (Decoded Picture Buffer、デコーダで蓄積するピクチャ枚数) を変更して性能測定を行った。表 2 を見ると、XG 6254 については、ピクチャバッファサイズが 64 枚の場合は、処理速度が 60fps に達しないが、これを 128 枚に増やせば、60fps 以上となることが確認された。また、TR 3970 については、ピクチャバッファサイズが 64 枚の場合は、処理速度が 60fps に達しないが、これを 96 枚に増やせば、60fps 以上を達成することが確認された。また、TR 3970X において、バッファサイズを 128 枚、160 枚、192 枚と変化させて性能評価を行った結果、復号処理速度は上昇するが、192 枚付近で処理速度の上昇幅が飽和している。さらに、バッファサイズは、表 2 に示すように、遅延とのトレードオフになるため、目標性能に見合ったスイートスポットを見つけることが重要となる。以上により、報告するデコーダ構成にて、4K/60fps の復号処理の実現性が確認された。



図 5 VVC リアルタイムデコーダの概観

表 1 復号処理速度性能評価に使用した CPU 構成

計算機の CPU 構成	Freq.	総 Thread 数
XG 6254×2 基	3.1GHz	72
TR 3970X×1 基	3.7GHz	64

表 2 CPU 別/バッファ別の復号処理速度

CPU	I. Period	DPB/遅延	処理速度
XG 6254	32	64 枚/1.0sec	49.0fps
XG 6254	32	128 枚/2.0sec	74.0fps
TR 3970X	32	64 枚/1.0sec	53.1fps
TR 3970X	32	96 枚/1.5sec	64.7fps
TR 3970X	32	128 枚/2.0sec	67.8fps
TR 3970X	32	160 枚/2.5sec	72.6fps
TR 3970X	32	192 枚/3.0sec	73.7fps

4.2. エラー耐性評価

FEC によるエラー耐性評価を行った。具体的には、IP ネットワーク・エミュレータを用いて、ネットワーク伝送上でパケットロスを疑似的に発生させ、その間の映像崩れの発生有無やデコーダのステータスログから FEC によるエラー回復効果を確認した。ここで、FEC のパリティパケット数は IPTV フォーラムが IPTV の運用仕様[12]で推奨する 2 次元 FEC (10x10) とし、パケットロス率 10^{-4} 以下とする場合と、 10^{-4} を超過する場合の 2 通りの実験を行った。なお、パケットロス率の閾値を 10^{-4} とした理由は、総務省省令の IP 放送技術基準において、誤り訂正前のパケットロス率が 10^{-4} 以下であることが条件となっているためである[13]。7.5 時間の長時間試験を行った結果、前者はその間映像崩れなどが一度も発生せずに連続動作することが確認された。後者は想定通り、FEC によるパケットロスの回復許容値を超過しているために、途中何度も映像崩れなどが発生することを確認した。以上により、報告するデコーダは、IP 実用網におけるネットワーク伝送に耐えうる性能を有していることが確認された。

4.3. スライス分割形状に対する符号化性能評価

3 節で述べたスライス分割形状に対する符号化性能を評価した。なお、符号化性能の評価指標には、JVET-CTC で使用される BD-rate を用いた。BD-rate は同一画質における符号量の削減効果を示す客観指標であり、負値が符号化性能の改善を示す[9]。報告するデコーダは、4K 映像素材におけるリアルタイム性能実現を目的としているため、4K 映像素材のみで評価した。具体的には、JVET-CTC における 4K 試験素材 (以下、

表 3 スライス分割形状に対する符号化性能評価

	分割なし	図 4b	図 4c
Class A1	-1.62%	0.31%	0.80%
Class A2	-1.04%	0.33%	0.47%
ITE 4K	-2.26%	-0.61%	1.17%
Overall	-0.98%	0.01%	0.49%

表 4 ITE 4K 素材別の符号化性能評価

	図 4b	図 4c
a06_Festival	0.06%	0.19%
a08_JapaneseMaple	-0.03%	0.10%
b05_HorseRace-Finish	-1.67%	2.54%
c14_Volleyball_follow	-1.32%	1.94%
c15_Paddock_fixed	-0.10%	1.08%



図 6 ITE 4K テストチャート B05, 競馬 (ゴール)



図 7 ITE 4K テストチャート C14, バレーボールフォロ

Class A1/A2) と映像情報メディア学会の 4K 試験素材 (以下, ITE 4K) の 5 点 (表 4 最左列に記載) を使用した. 符号化条件は JVET-CTC の RA 条件に準拠した. 表 3 に, 図 4a をアンカーとした, スライス分割なし, 図 4b のスライス分割, 図 4c のスライス分割の結果が示される. まず, 分割なしの結果は, 全シーケンスの平均で -0.98% の符号化性能の改善を示している. これは, 前述の通り, スライス分割がない場合, 分割がある場合と比較して, 参照範囲が広いために予測性能が向上したことに起因すると考えられる. 次に, 図 4b の結果は, 全シーケンス平均では 0.01% と図 4a の符号化性能と同程度であるが, 個別にみると Class A1/A2 の符号化性能低下分を ITE 4K の符号化性能増加分が打ち消している. 表 3 は ITE 4K のシーケンス別の

BD-rate を示している. 図 4b で大きな符号化性能の改善がみられた図 6 に示される b05_HorseRace-Finish[10] と図 7 に示される c14_Volleyball_follow[11] は, シーン内で左右方向にカメラパンが複数回現れる. そのため, 横長に参照範囲を確保できる図 4b で符号化性能が大きく改善したと考えられる. 最後に, 図 4c の結果は, 全シーケンス平均では 0.49% の符号化性能の低下を示している. また, シーケンス個別の結果でも, 符号化性能は低下している. これは, 前述の通り, 縦長のスライス分割は自然映像に対して不利な結果が如実に現れた結果である. 以上より, スライス分割による性能低下を抑制し, かつ汎化性能が高い分割方法は図 4a の方法であることが確認された.

5. まとめ

本稿では, VVC 方式で符号化した 4K 映像をリアルタイムに復号処理可能なソフトウェアデコーダの試作開発結果を報告した. 報告したデコーダは, スライス並列とピクチャ並列をマルチスレッドで実現し, 4K/60fps 映像のリアルタイム復号処理の実現を実験により確認した. 今後は, VVC の適用先拡大を目指して, 更なる高解像度や高フレームレートへの対応を検討する.

謝辞: 本研究開発は総務省 SCOPE (国際標準獲得型) JPJ000595 の委託を受けたものです.

文 献

- [1] B. Bross, J. Chen, and S. Liu, "Versatile Video Coding (Draft 10)," JVET-S2001, teleconference, July 2020.
- [2] Y.-K. Wang, J. Chen, Y. Ye, and S. Kim, "Algorithm description for Versatile Video Coding and Test Model 9 (VTM 9)," JVET-R2002, teleconference, April 2020.
- [3] 大久保 榮, 鈴木輝彦, 高村誠之, 中條健, "H.265/HEVC 教科書", インプレスジャパン, 2013 年 10 月.
- [4] C. Rosewarne, K. Sharman, R. Sjöberg, and G. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 13," Brussel, January 2020.
- [5] F. Bossen, X. Li, and K. Sühring, "JVET AHG report: Test model software development (AHG3)," JVET-S0003, teleconference, July 2020.
- [6] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "JVET common test conditions and software reference configurations for SDR video," JVET-N1010, Geneva, March 2019.
- [7] 大久保 榮, 角野真也, 菊池義浩, 鈴木輝彦, "改訂三版 H.264/AVC 教科書", インプレスジャパン, 2008 年 12 月.
- [8] "Pro-MPEG Forum Code of Practice 3 Release 2", Transmissions of Professional MPEG-2 Transport Streams Over IP Networks, 2004.
- [9] G. Bjøntegaard, "Calculation of average PSNR

differences between RD-curves,” in *ITU-T SG 16 WP 3*, VCEG-M33, Texas, April 2001.

- [10] https://www.ite.or.jp/contents/chart/uhdtv_b/mp4/b05_HorseRace-Finish4K.mp4
- [11] https://www.ite.or.jp/contents/chart/uhdtv_hdr/mp4/c14_Volleyball_follow_4K_960x540.mp4
- [12] 「IPTV 規定 地上デジタルテレビジョン放送 IP 再送信運用規定(IPTVFJ STD-0005 1.4 版)」第零編, 第 6 編, 第 7 編(抜粋), IPTV フォーラム, 2017 年 7 月.
- [13] パケロス率に関する総務省省令, 有線一般放送の品質に関する技術基準を定める省令 (最終更新日: 令和元年 5 月 20 版を確認) https://elaws.e-gov.go.jp/search/elawsSearch/elaws_search/lsg0500/detail?lawId=423M60000008095#G