

コンシューマ・デバイス論文

動画像リアルタイムエンコーダにおける H.264 CABACの並列処理技術

谷田部 祐介^{1,a)} 小味 弘典¹ 海野 恭平¹ 吉田 大輔¹ 伊藤 浩朗¹ 吉岡 理文²

受付日 2015年5月21日, 採録日 2015年10月1日

概要: 動画像をリアルタイムに符号化するハードウェアエンコーダでは, 画像の高解像度化にともない, 処理の高速化が求められている. 従来はフレーム内を複数のスライスに分割し, スライスを単位として並列処理することで高速化を実現する手法が用いられているが, 本方式ではスライス境界部での画質劣化が問題となっていた. 本論文では, フレームを単位として CABAC (Context-based Adaptive Binary Arithmetic Coding) 処理の並列化を行うことで, 高速かつ高画質な符号化を実現したので報告する.

キーワード: H.264, CABAC, リアルタイムエンコーダ, ハードウェア並列化

A Parallelization Technology of H.264 CABAC for Real Time Encoder of Moving Picture

YUSUKE YATABE^{1,a)} HIRONORI KOMI¹ KYOHEI UNNO¹ DAISUKE YOSHIDA¹ HIROAKI ITO¹
MICHIFUMI YOSHIOKA²

Received: May 21, 2015, Accepted: October 1, 2015

Abstract: In a hardware encoder for encoding moving images in real time with high resolution, high-speed processing is required. Conventionally, techniques for achieving a high speed by parallel processing each slice in frame are used. However, the image quality in the boundary slice is degraded by this process. In this paper, it is possible to perform the parallelization of the CABAC (Context-based Adaptive Binary Arithmetic Coding) for each frame. We have achieved high-speed, high-quality H.264 encoding.

Keywords: H.264, CABAC, real time encoder, hardware parallelization

1. はじめに

近年, 一眼レフカメラやビデオカメラなどのモバイル機器の動画撮影機能において, 4k などの高精細化が急速に進んでいる. また, 高画質や編集の利便性から動画像撮影において, 1 枚のフレームで完結した圧縮を行うイントラフ

レームのみで符号化を行う (ALL-I-picture) 機能の搭載も注目を集めている. この場合, 高精細化による画素数の増加や, ALL-I-picture 機能にともなう圧縮率低下に対応するため, 高ビットレートでの符号化が重要となる.

これら機器における動画像符号化機能は, リアルタイム処理が前提であり, 処理速度や消費電力の点からハードウェアで実装される場合が多く, 高ビットレート化に対応するため処理の高速化が求められている. ハードウェアの高速化は, クロックアップによる処理速度向上が用いられる場合が多いが, 消費電力が増大してしまう問題 [1] があるため, モバイル機器では処理を並列化して高速化を図る手法が多く採用されている.

動画像符号化については, 2003 年に H.264 [2] が規格

¹ (株) 日立製作所研究開発グループシステムイノベーションセンター
Hitachi, Ltd., Research & Development Group, Center for Technology Innovation – Systems Engineering, Kokubunji, Tokyo 185-8601, Japan

² (公) 大阪府立大学大学院工学研究科電気・情報系専攻知能情報工学分野

Graduate School of Engineering, Department of Computer Science of Intelligent Systems, Osaka Prefecture University, Sakai, Osaka 599-8531, Japan

^{a)} yusuke.yatabe.wj@hitachi.com

化された。この規格では可変ブロックサイズ、コンテキスト適応型 2 値算術符号化 (Context-based Adaptive Binary Arithmetic Coding (CABAC)) が導入された。この CABAC は、算術符号化を用いすでに符号化した周辺の情報を用いることで高圧縮率を実現している。しかし、中間出力である 2 値符号 1 bit (以下 bin) をシーケンシャルに符号化する必要があるため、処理の並列化は困難であるという問題があった。

CABAC 高速化については、シーケンシャル処理の処理速度向上について多くの研究 [3], [4], [5] がなされているが、急激な高ビットレート化への対応には限界があると考えられる。

一方、画像の並列符号化については、従来、画面をスライスに分割して並列処理させる方法がとられているが、境界付近での予測効率劣化やスライスヘッダの付加情報による画質劣化が問題 [6] となっている。

筆者らは、CABAC 処理はフレームごとの処理依存性がないことに着目し、画面をスライスで分割せずにフレーム単位で並列処理させることで、処理高速化と高画質化を実現可能な手法を提案する。本論文は 2 章で CABAC 処理概要を説明し、3 章で従来研究、4 章で提案方式について説明し、5 章で実験結果と考察について述べる。

2. CABAC 処理の概要

本章では、CABAC 処理が画面内で並列処理が困難である理由について説明する。CABAC 処理は、図 1 に示すように大きく 3 つの処理から構成される。

初めに、2 値化部において、動ベクトル、予測モード、DCT 係数などの多値入力データを、規格で定められた方法で 2 値 (0/1) データへ変換する。

次に、2 値データをコンテキスト計算部で算出した確率モデルをもとに算術符号化部にて 2 値算術符号化し、エンコーダの出力であるビットストリームを生成する。このとき、入力される 2 値データ 1 ビットごとに、直前に符号化した 2 値データの値 (0/1) などの情報をもとにコンテキスト計算部にて算術符号化に用いる確率モデルの更新を行う。

このように、CABAC では 2 値データ 1 ビットごとに確率モデルを更新しながらシーケンシャルに符号化処理を行うため、並列化が困難であるという特徴がある。なお、こ

の確率モデルの更新や算術符号化は、スライスごとに閉じた処理となる。

ここで、2 値化部の処理は入力データに 1 対 1 で対応する 2 値データを出力するため、リアルタイムで符号化処理を行う場合の単位時間あたりの処理量は、入力データの数、すなわち画像サイズやフレームレートに依存する。

これに対し、確率モデル更新や算術符号化部の処理量は単位時間あたりに入力される 2 値データ量に依存する。2 値データ量とビットレートの間には相関があるため、確率モデル更新部や算術符号化部の処理量はエンコーダのビットレートに依存する [3] といえる。

3. 従来研究

CABAC の高速処理における従来研究について説明する。CABAC 処理自体の高速化については、主に処理速度のボトルネックとなる 2 値化処理の高速化について検討されている。文献 [3], [4], [5] では、CABAC 処理の独立部分に着目し、それらの部分の並列化を行うことで、デコード側で約 1 bin/clock 以下、エンコード側で約 1~2 bin/clock 程度の処理を可能としている。

しかし、近年の画像サイズやフレームレートの増大は著しく、CABAC 処理自体の高速化には限界がある。さらに、CABAC 処理単体の高速化にはクロックアップがともなうため、消費電力の問題から並列化による高速化手法の導入も合わせて考慮する必要がある。

次に、画面内をスライスに分割して並列処理する手法について説明する。本手法は、高速化に対して通常用いられる手法であり、文献 [6] では、17 スライス分割を用い、スライス境界での画質劣化の問題について記載されている。

また、CABAC の各フレームの発生符号量の違いに着目し、CABAC 処理の前後でパイプラインの処理を分けて、CABAC 前の中間データをメモリに数フレーム分バッファリングすることで、リアルタイムの符号化を実施する手法 [7] が報告されている。本技術は CABAC のリアルタイム処理化には必須の技術であるが、CABAC 処理の高ビットレート対応については考慮されていない。

よって本論文では、従来方式で問題であった CABAC のスライス並列処理にともなうスライス境界部の画質の劣化を抑制し、スライス並列化と同等の高速化を実現可能とする手法について検討した。

4. 提案方式

提案方式について説明する。本提案方式では、CABAC 処理をスライス分割することなしに、フレーム単位に並列化する手法を提案する。

図 2 は、提案方式の構成を示す図である。本説明では、論文 [5] をベースに、エントロピー符号化部の前段処理は中間情報をメモリにいったん格納する構成とし、CABAC

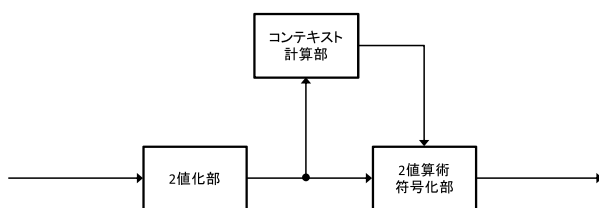


図 1 CABAC の処理概要

Fig. 1 Overview of CABAC.

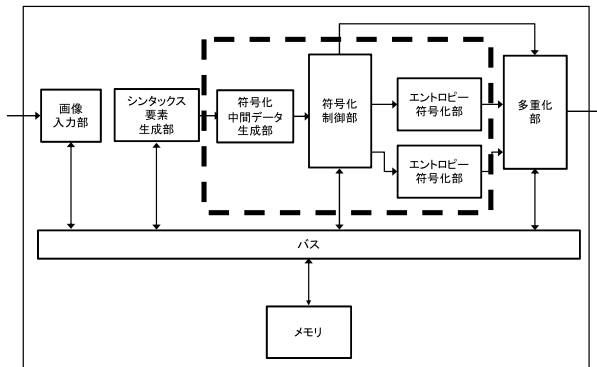


図 2 提案方式のブロック図

Fig. 2 Block diagram of proposed method.

を行うエントロピー符号化部は最大 50 Mbps の符号化処理能力を有し、2 並列処理させることで 80 Mbps のビットレートを実現する場合を例に説明する。

符号化処理は、画像入力部、シンタックス要素生成部、中間データ生成部、符号化制御部、それぞれ CABAC を行う第 1 エントロピー符号化部、第 2 エントロピー符号化部、多重化部、メモリ、バスより構成され、画像入力部から符号化対象画像データを入力し、多重化部から符号化後のビットストリームを出力する。破線は CABAC 処理を行う構成部分であり中間データ生成部、符号化制御部、第 1 エントロピー符号化部、第 2 エントロピー符号化部が相当する。

メモリ部は、SDRAM を採用することを想定し、複数のバッファ領域を有し、符号化処理にともなう各種データを記憶する。以下、符号化処理の各部の動作を説明する。

画像入力部は符号化対象の画像データを入力し、バス経由でメモリ内の符号化対象画像用バッファに書き込む。シンタックス要素生成部は、入力した画像データについて、 16×16 画素サイズのマクロブロック（以下、MB）ごとに予測符号化や離散コサイン変換、量子化などの処理を行う。そして、ヘッダ情報、動きベクトル、量子化後の変換係数などの H.264 で規定されたシンタックス要素を生成する。その際予測符号化処理に必要な画像データは、符号化対象画像については符号化対象画像用バッファ、参照画像である復号画像については復号画像用バッファからそれぞれ読み出して使用する。また、生成したシンタックス要素に基づいて当該 MB の復号画像を作成し、復号画像用バッファに書き込む。生成されたシンタックス要素は、中間データ生成部へ出力される。

中間データ生成部は図 1 の 2 値化部に相当し、CABAC で規定されている方法でシンタックス要素を 2 値化し、符号化中間データ（中間データ）を生成する。生成された中間データは符号化制御部へ出力される。

符号化制御部は、中間データ生成部から出力される中間データを、フレーム単位で、第 1 エントロピー符号化部、

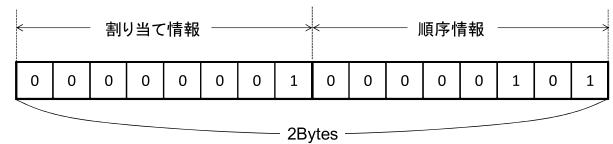


図 3 多重化制御情報

Fig. 3 Multiplexing information.

または第 2 エントロピー符号化部のいずれかへ出力する。その際、中間データをバス経由でメモリ内の中間データ用バッファへいったん書き込んでおき、次のフレームの符号化処理を行うエントロピー符号化部を決定し、バッファから該当フレームの中間データを読み出して前記決定したエントロピー符号化部へ出力する。後述するが、次のフレームの符号化処理を行うエントロピー符号化部に、符号化処理が終了しているエントロピー符号化部を割り当てる。

また符号化制御部は、エントロピー符号化部へ出力するフレームに関する多重化制御情報を生成して、多重化部に通知する。この多重化制御情報には、次に出力するフレームが符号化制御部へ何番目に入力されたフレームであるか（符号化順序情報）と、次に出力するフレームがいずれのエントロピー符号化部で処理されるか（符号化割り当て情報）が含まれる。なお、次フレームに対するエントロピー符号化部の決定手法と多重化制御情報については後述する。

第 1 エントロピー符号化部と第 2 エントロピー符号化部はいずれも同一の構成であり、図 1 のコンテキスト計算部と 2 値算術符号化部に相当する。ここでは、符号化制御部から出力される中間データ（2 値化データ）に対し、確率モデルに基づいてエントロピー符号化（2 値算術符号化）を行い、それぞれ第 1 および第 2 ビットストリームを生成して多重化部へ出力する。

多重化部は、第 1 および第 2 エントロピー符号化部で符号化された第 1 および第 2 ビットストリームを 1 つのビットストリームに多重化処理する。そのために、エントロピー符号化部から出力される第 1 および第 2 ビットストリームを、バスを経由して、メモリの第 1 および第 2 ビットストリーム用バッファに書き込む。また、バッファから第 1 および第 2 ビットストリームを読み出して、多重化処理を行う。その際、符号化制御部から受け取った多重化制御情報に従い、ビットストリームをフレーム単位に区別してバッファに書き込み、またフレーム単位で順番に読み出しを行う。

図 3 は、符号化制御部から多重化部に送る多重化制御情報を示す図である。多重化制御情報は符号化順序情報と符号化割り当て情報からなり、今後複数のエントロピー符号化部を有することも考慮してそれぞれ 1 バイト（全 2 バイト）で記述している。符号化順序情報は、符号化制御部に前記中間データが 1 フレーム分入力されるごとにインクリメントされる値である。また、符号化割り当て情報は、第

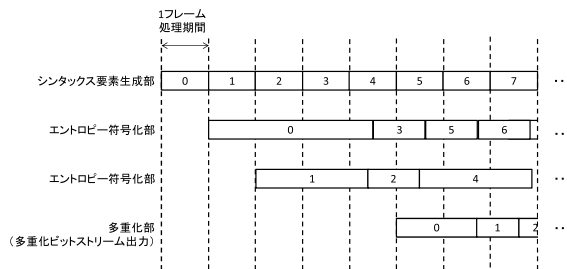


図 4 タイミングチャート

Fig. 4 Timing chart.

1 エントロピー符号化部を使う場合は「0」、第2エントロピー符号化部を使う場合は「1」を与えている。多重化部では、多重化制御情報を参照することで、バッファに分かれて書き込まれた第1および第2ビットストリームをフレーム順に読み出し、もとの順序のビットストリームに変換することができる。

次に、本手法における符号化処理を説明する。図4は、符号化処理動作のタイミングを示す図である。ここでは、シンタックス要素生成部、第1および第2エントロピー符号化部、多重化部の処理タイミングを示す。横軸は時間軸で1フレーム処理期間TFを単位として表示している。1フレーム処理期間TFとは、符号化対象画像を実時間で符号化する場合に1フレームの処理に割り当て可能な時間であり、たとえばフレームレートが30fpsの画像の場合、TF=約33msecとなる。図中の0, 1, 2...は処理されるフレーム番号を示している。

シンタックス要素生成部では、1フレーム処理期間TF内に各フレームのシンタックス要素を順次生成する。シンタックス要素生成部での処理時間は符号化対象画像の画素数に依存し、各フレームの処理時間は一定となる[7]。第1エントロピー符号化部と第2エントロピー符号化部は、各フレームの中間データのエントロピー符号化（以下、単に符号化という）を行いビットストリームに変換する。その際符号化制御部は、次のフレームの符号化を符号化処理が終了しているエントロピー符号化部に割り当てる。

また、第1および第2エントロピー符号化部での符号化処理時間は、各フレームの符号量の大小に依存する。エントロピー符号化部の処理能力はいずれも最大50Mbpsであり、1フレーム処理期間TFに最大で約1.7Mbitを処理できる。これらエントロピー符号化部での処理フレームの割り当てについては後述する。

割り当てられたエントロピー符号化部では、割り当てられたフレームのすべての中間データについて符号化処理を行い、他のエントロピー符号化部では当該フレームの中間データの処理は行わないようにする。また、それぞれのエントロピー符号化部では、割り当てられたフレームの中間データの符号化処理を完了するまでは、他のフレームの中間データの符号化処理を行わないようにする。

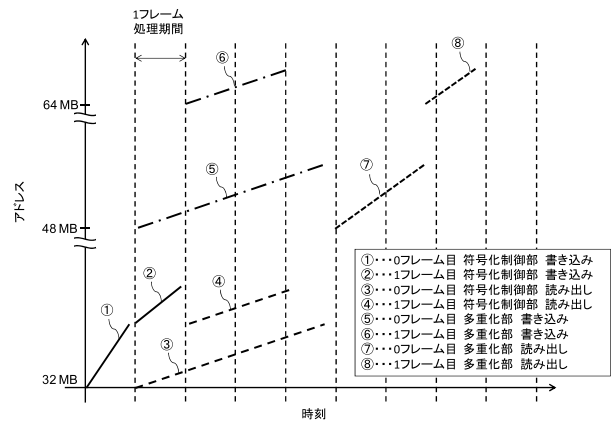


図 5 メモリ部への R/W タイミング

Fig. 5 R/W timing for memory block.

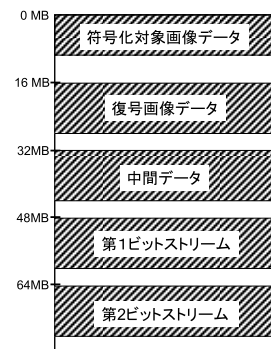


図 6 メモリマッピング

Fig. 6 Memory mapping.

すなわち、1つのフレームの中間データが、2つのエントロピー符号化部にわたって処理されることはなく、また、1つのフレームの中間データの処理が途中で休止し、他のフレームの処理に切り替わることはない。

図5は、メモリへのデータ読み書きタイミングを示す図である。縦軸はメモリアドレス、横軸は時間軸を示し、フレーム0, 1に対して、符号化制御部の書き込みと読み出し、多重化部の書き込み、読み出しを直線で表記している。

メモリマッピングについて図6に示す。メモリには、画像符号化に必要な符号化対象画像データ、復号画像データ、中間データ（32～48MB）、第1ビットストリーム（48～64MB）、第2ビットストリーム（64MB～）を格納する。

中間データの書き込みを説明する。符号化制御部は、中間データ生成部から出力されるフレームの中間データをバッファにアドレスを連続させて書き込む（実線①, ②）。

書き込みタイミングは、シンタックス生成部のデータ生成と同時に、1フレーム処理時間TFかけて行う。中間データの量はフレーム0と1で異なるため、①と②の書き込み量は異なっている。このとき符号化制御部は、1フレームの書き込みが終わるごとに符号化順序情報と終端のアドレスを記憶している。

中間データの読み出しでは、符号化制御部は、バッファから各フレームの中間データを読み出す（破線③, ④）。そ

の際、前記書き込み時に記憶している各フレームの符号化順序情報と終端のアドレスをもとに、中間データを符号化処理順序に、かつ1フレーム単位で読み出す。読み出した中間データは、後述する割り当て規則に従うが、本説明では、フレーム0は第1エントロピー符号化部へ、フレーム1は第2エントロピー符号化部へ出力するものとする。読み出しタイミングはエントロピー符号化部での符号化開始に合わせ、シンタックス要素生成部での処理終了時から開始する。また読み出し量は、エントロピー符号化部の処理速度に合わせ、一定としている。そのため、フレーム0, 1, ... の読み出し時間は、それぞれの中間データの量に応じて異なる。

次にビットストリームの書き込みを説明する。多重化部は、第1エントロピー符号化部から出力されるフレーム0のビットストリーム、第2エントロピー符号化部から出力されるフレーム1のビットストリームをバッファに書き込む（点鎖線⑤、⑥）。書き込みタイミングは、それぞれのエントロピー符号化部での符号化処理に同期し、フレーム0, 1はそれぞれエントロピー符号化部の処理開始から開始する。多重化部は符号化制御部から受け取った多重化制御情報（符号化順序情報）に基づき、書き込むビットストリームが何番目に出力するフレームであるかを記憶し、またバッファのどこに書き込んだか（先頭アドレス情報）と各フレームのサイズ情報を記憶しておく。

ビットストリームの読み出しでは、多重化部は、バッファからフレーム0のビットストリームを、続いてバッファからフレーム1のビットストリームを読み出す（点線⑦、⑧）。読み出しのタイミングは、符号化制御部から受け取った多重化制御情報（符号化順序情報）に従い、書き込み時に記憶している各フレームの先頭アドレス情報を参照し、当該フレームのサイズ分の読み出し処理を行う。これにより、符号化制御部へ中間データが入力された順序でビットストリームを多重化して、規格に準拠したビットストリームを出力することができる。

次に、符号化制御部において、各フレームの中間データを第1エントロピー符号化部と第2エントロピー符号化部のいずれで符号化するかを決定する手法を説明する。

図7は、中間データの符号化割り当ての決定法を示すフローチャートである。

図の①において、メモリの中間データ用バッファに格納されている中間データの中で、まだエントロピー符号化部でエントロピー符号化処理を開始していないフレームの中間データ（未符号化フレームデータ）の有無を判定する。ここでは、シンタックス要素生成部での処理が1フレーム分終了した時点で、初めて未符号化フレームデータと認める。すなわち、シンタックス要素生成部で処理中のフレームのデータが存在しても、未符号化フレームデータとは認めない。

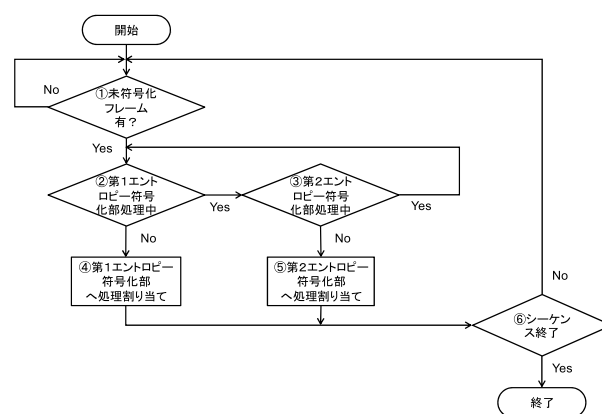


図7 処理フローチャート

Fig. 7 Processing flow cart.

この判定で未符号化フレームデータなし（No）と判定された場合は、未符号化フレームデータありと判定されるまで、ループ処理を繰り返す。一方、未符号化フレームデータあり（Yes）と判定された場合は②へ進む。

②では、第1エントロピー符号化部が他のフレームデータについて符号化処理中であるか否かを判定する。処理中である（Yes）場合は③へ、処理中でない（No）場合は④へ進む。

④では、未符号化フレームデータの符号化処理を第1エントロピー符号化部で行うことに決定する。そして、該当フレームの多重化制御情報を多重化部へ通知した後、未符号化フレームデータをメモリから読み出し、第1エントロピー符号化部へ出力する。

③では、第2エントロピー符号化部が他のフレームデータについて符号化処理中であるか否かを判定する。処理中である（Yes）場合は②に戻り、②または③のいずれかの判定が処理中でない（No）となるまでループ処理する。③の判定が処理中でない（No）場合は⑤へ進む。

⑤では、未符号化フレームデータの符号化処理を第2エントロピー符号化部で行うことに決定する。そして、該当フレームの多重化制御情報を多重化部へ通知した後、未符号化フレームデータをメモリから読み出し、第2エントロピー符号化部へ出力する。

⑥では、符号化割り当てのシーケンスを終了するか否かを判定する。終了ではない場合は、①に戻り前記処理を繰り返す。

上記フローチャートによれば、未符号化フレームデータがある場合、符号化処理が早く終わったエントロピー符号化部を用いて次のフレームの符号化処理を連続的に実行することが可能となる。これより、第1および第2エントロピー符号化部の休止期間を最小とし、処理パフォーマンスを最大限に活かすことが可能となる。すなわち本手法は、フレームごとに符号量が異なっても各エントロピー符号化部はほぼ100%の稼働率で動作するため、エントロ

ピー符号化部に課される処理能力は最小限のビットレートの値で十分であるという特徴を有する。

5. 実験結果

提案方式で説明した CABAC の処理能力が 50 Mbps, 目標ターゲットビットレート 80 Mbps の条件において, 実際にソフトウェアエンコーダを用いて画質評価を行った. 評価条件を表 1 に示す. 評価は, 再現性を考慮して H.264 のリファレンスソフト (JM18.6) を用いて, 画面内のスライス分割を行わない提案方式と, スライス分割を行う従来方式において輝度情報の定量画質指標 (PSNR) を比較した. この際スライス分割は, スライス分割内の符号化対象マクロブロック数が均等となるように, 画面内を横方向にそれぞれ, 2, 4, 17, 34, 68 分割を行った.

評価結果を図 8 に示す. 本結果より, 提案方式となるスライス分割をしない方式は, 従来スライス分割を行う方式に比較して, 最大約 0.9 dB の画質の劣化を抑制できることを確認した.

次に, 低ビットレート時の方式間の性能差を確認するために, 各 CABAC コアの処理能力を半分 (25 Mbps) と想定し, 40 Mbps をエンコードした際の結果を, 図 9 に示す.

本結果では, 提案方式は従来方式に比較して, 最大 1.2 dB の画質の劣化を抑制できることを確認した.

次に, 両方式の主観画質の比較について示す. 図 10 は, s209 画像における 80 Mbps エンコード時 329 frame 目の再生画像 (一部拡大) を示している. 図 10 (a) は, スライス分割を行わない提案方式. 図 10 (b) は, 17 スライス分割した際のスライス境界, 図 10 (c) は, 17 スライス分割を実

施した従来方式の再生画像である.

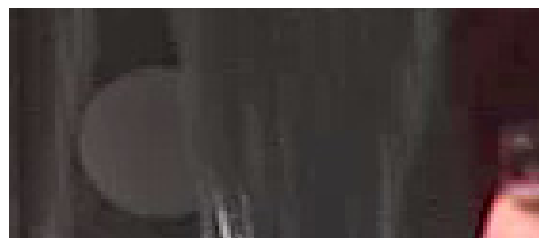
結果より, 図 10 (c) 従来方式ではスライス境界の上下で画質 (特に色) が不連続になり, 原画像には存在しないスライス境界が発生しているが, 図 10 (a) 提案方式ではそのようなスライス境界での画質劣化は生じていないことが主観的に確認できる.

実際に提案方式で示したハードウェアを FPGA に実装した (図 11 のファンの下に存在). 実装したハードウェアは, 論文 [5] の構成をベースに, 2 倍のビットレートに対応するため, CABAC 部を 2 個用いて並列処理可能な構成としている. おのものの CABAC 部は, それぞれ独立にメモリへのアクセスを可能とし, メモリ部は, 中間データ, ビットストリームの領域を 2 倍化した. また, 2 つの CABAC 回路への振り分け制御は, RISC プロセッサにより実装した.

方式		PSNR [dB]					
		s204	s209	s210	s213	s264	s265
提案方式		26.8	35.7	35.4	41.2	40.9	32.8
従来方式	2スライス	26.8	35.7	35.4	41.2	40.8	32.7
	4スライス	26.8	35.7	35.3	41.2	40.8	32.7
	17スライス	26.7	35.6	35.1	41.1	40.7	32.5
	34スライス	26.6	35.4	34.8	41.0	40.4	32.3
	68スライス	26.5	34.9	34.2	40.8	40.3	31.8

図 9 画質評価結果 (40 Mbps)

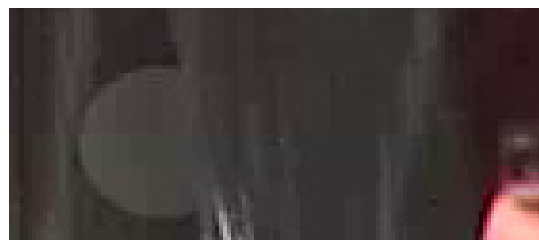
Fig. 9 Simulation results (40 Mbps).



(a) 提案方式 (s209, 80Mbps)



(b) 従来方式におけるスライス境界



(c) 従来方式 (s209, 80Mbps, 17 分割)

図 10 主観画質の比較

Fig. 10 Comparison of subjective image quality.

表 1 符号化条件

Table 1 Coding conditions.

項目	条件
ソフトウェア	JM 18.6 ^[8]
評価画像	ITE標準画像 ^[9] s204, s209, s210, s213, s264, s265
画像サイズ	1920 x 1088
枚数	それぞれ900枚
フレームレート	60fps
符号化構造	ALL Iピクチャ
レートコントロール	ON
Rate Distortion Optimization	OFF
ターゲットビットレート	40Mbps, 80Mbps
スライス分割	2, 4, 17, 34, 68

方式		PSNR [dB]					
		s204	s209	s210	s213	s264	s265
提案方式		30.3	39.0	39.1	42.9	42.8	36.8
従来方式	2スライス	30.2	38.9	39.0	42.9	42.8	36.8
	4スライス	30.2	38.9	39.0	42.9	42.8	36.7
	17スライス	30.1	38.8	38.9	42.8	42.7	36.6
	34スライス	30.0	38.7	38.7	42.7	42.5	36.3
	68スライス	29.9	38.4	38.4	42.6	42.4	35.9

図 8 画質評価結果 (80 Mbps)

Fig. 8 Simulation results (80 Mbps).



図 11 エンコーダ FPGA 基板

Fig. 11 FPGA board for encoding.

本実装のハードウェアによりリアルタイムで符号化動作を確認している。

5.1 考察

本論文では、CABAC 処理の高速化を目的に、フレーム間で並列処理する手法を提案した。本手法は、画面をスライスに分割する必要がないため、分割による画質劣化を抑制することが可能となる。この分割による画質劣化は、実験の結果、スライス数を増やすほど、また、ビットレートを下げるほど大きくなることが分かった。本実験では、スライス分割を用いることで、約 1.0dB 程度の画質劣化が発生してしまう場合があることを確認したが、画像符号化において 1.0dB 程度の劣化は、圧縮率に換算すると 10%以上の劣化との報告 [11] もあり、非常に大きな劣化であると考えられる。

また、文献 [6] では、17 スライスに分割して符号化を行う手法が紹介されている。今後も画像の高精細化、高フレームレート化などの進化は急激に進むと推測され、さらに多くのスライス分割による並列処理を行う必要性が増えていく。そのため、本手法の適用が有効になっていくと考えられる。

また、本技術は、従来から研究が行われている CABAC 自体の高速化の手法と組み合わせることが可能であり、両者を組み合わせることで、より高精細、高フレームレートの画像を符号化することが可能となると考えられる。

本論文では、FPGA に提案方式の実装も行った。本実装により、従来構成 [5] に比較して、ビットレート増加にとりなう中間データのメモリ量や、メモリ R/W のバンド幅の増加が見込まれる。しかし、CABAC の並列化手法として、スライス単位の並列化とフレーム単位の並列化を比較すると、同ビットレートである場合、両者の違いは並列処理単位が異なる点となり、ほぼ同等の規模で実現が可能であると考えられる。

5.2 まとめ

高速化が求められる CABAC 処理において、フレーム単位に並列化することで、高速かつ高画質なエンコードを可能とする手法の有効性を示した。本手法は、今後のさらなる情報量の増大にも並列度を増加させることにより対応が可能な拡張性のある技術である。今後は、近年規格化された H.265 [10] に対してもさらに導入の検討を進めていきたい。

参考文献

- [1] 十山, 他: CPU 消費電力削減のための周波数—電圧協調型電力制御方式の設計ルールとフィードバック予測方式による適用, 電子情報通信学会論文誌 D, Vol.J87-D1, No.4, pp.452-461.
- [2] Recommendation ITU-T H.264 | International Standard ISO/IEC 14496-10.
- [3] 藤田, 他: 残差信号アクセラレータによる H.264 CABAC 復号器の高速化, 電子情報通信学会技術研究報告, SIP, 信号処理, Vol.111, No.257, pp.31-35 (Oct. 2011).
- [4] Yahya et al.: CABAC Accelerator Architectures for Video Compression in Future Multimedia: A Survey, Springer, *Embedded Computer Systems: Architectures, Modeling, and Simulation*, Lecture Notes in Computer Science Vol.5657, pp.24-35 (2009).
- [5] Mizosoe et al.: A Single Chip H.264/AVC HDTV Encoder/Decoder/Transcoder System LSI, *IEEE Trans. Consumer Electronics*, Vol.53, Issue 2, pp.630-635 (May 2007).
- [6] 境田, 他: 8K スーパーハイビジョン HEVC/H.265 エンコーダ装置の開発, 映像情報メディア学会誌, Vol.69, No.1, pp.J23-J29 (2015).
- [7] 岩田, 他: 携帯機器向けフル HD 対応 H.264 ハイプロファイルビデオコーデック IP の開発, 情報処理学会研究報告, 計算機アーキテクチャ研究会報告, Vol.2009, No.1, pp.111-116 (Jan. 2009).
- [8] Recommendation ITU-T H.264 | International Standard ISO/IEC 14496-10, reference software JM version 18.6, available from <http://iphome.hhi.de/suehring/tml/download/>.
- [9] ハイビジョン・システム評価用標準動画像第 2 版, 入手先 <https://www.nes.or.jp/gaiyo/pdf/manual-rev1.3.pdf>.
- [10] Recommendation ITU-T H.265 | International Standard ISO/IEC 23008-2 HEVC.
- [11] Calculation of Average PSNR Differences between RD-curves, ITU-T VCEG-M33 (2001), available from http://wftp3.itu.int/av-arch/video-site/0104_Aus/VCEG-M33.doc.



谷田部 祐介 (正会員)

1999 年東京理科大学理工学部電気工学科卒業。2001 年同大学大学院修士課程修了。同年 (株) 日立製作所に入社。2015 年大阪府立大学博士課程入学現在在学中。画像圧縮伸張技術, 画像高画質化技術の研究開発に従事。映像情報メディア学会会員。



小味 弘典

1969 年生. 1992 年大阪府立大学電気工学科卒. 1994 年同大大学院修士課程修了. 同年 (株) 日立製作所に入社. 画像圧縮伸張技術, 映像メディア処理の研究開発に従事. 1999~2000 年南カリフォルニア大学客員研究員. 映像

情報メディア学会会員.



海野 恭平

2009 年東京理科大学理工学部電気電子情報工学科卒業. 2011 年同大学大学院理工学研究科電気工学専攻修士課程修了. 同年 (株) 日立製作所入社. 画像符号化技術の研究開発に従事. 2011 年情報処理学会山下記念研

究賞受賞. 映像情報メディア学会会員.



吉田 大輔

2000 年東京大学工学部システム量子工学科卒業. 2002 年同大学大学院修士課程修了. 同年 (株) 日立製作所に入社. 画像圧縮伸張技術, 画像高質化技術の研究開発に従事.



伊藤 浩朗

1990 年横浜国立大学工学部電子情報工学科卒業. 1992 年同大学大学院修士課程修了. 2004 年 (株) 日立製作所に入社. 画像圧縮伸張技術, 画像高質化技術の研究開発に従事. 映像メディア学会会員.



吉岡 理文 (正会員)

1968 年生. 1996 年東京大学大学院工学系研究科博士課程修了. 博士 (工学). 同年 4 月大阪府立大学工学部助手. 1998 年 11 月同講師, 2004 年 4 月同助教授, 2010 年 4 月同教授となり

現在に至る. 主として画像処理等の研究に従事. 電気学会, システム制御情報学会, 計測自動制御学会各会員.