

HEVC 技術解説【エントロピー符号化】

Entropy Coding in HEVC

服部 亮史†
Ryoji Hattori

1. まえがき

HEVC (High Efficiency Video Coding)[1]におけるエントロピー符号化手法について解説する。HEVC ではエントロピー符号化手法として、H.264/AVC[2]にも用いられている CABAC (Context Adaptive Binary Arithmetic Coding)[3]が全てのプロファイルに導入されている。HEVC の CABAC は、H.264/AVC と比較して演算量・メモリ量が削減されており[4]、また並列処理を志向したツールも導入され、エントロピー符号化・復号化処理のスループット向上が図られている。

2. CABAC

2-1. 概要

CABAC は、二値算術符号化、および 2 値シンボルの生起確率を周囲のパラメータの状態 (コンテキスト) に基づいて推定するコンテキスト適応処理を用いたエントロピー符号化手法である。CABAC で符号化されるパラメータはスライスヘッダより下位のパラメータであり、SPS, PPS, スライスヘッダなど上位シンタックスのパラメータは固定長符号、指数ゴロム符号などを用いて符号化される。

HEVC 標準化作業の過程では、CABAC の他に CAVLC, PIPE/V2V, PIPE/V2F と呼ばれるエントロピー符号化方式も検討された[5]。CABAC は他の方式と比較するとスループットの面で不利であったものの、符号化効率の高さ、H.264/AVC で運用されている実績により採用された。

2-2. 二値算術符号化

二値算術符号化は、入力される 2 値シンボルの値および MPS (Most Probable Symbol: 0 と 1 のうち出現する確率が高いシンボル) 生起確率に基づいて数直線を区間分割していき、最終的に得られた区間の 2 進数表現を入力 2 値シンボル系列の符号語とする符号化手法である[6]。この処理は、2 値シンボルの情報源を、全ての 2 値シンボルの個数まで拡大する処理に相当し、理論限界に近い効率の符号化が実現される。CABAC の二値算術符号化は整数演算のみで符号化/復号化が処理できるよう設計されている。

2-3. コンテキスト適応処理

CABAC は、ある 2 値シンボルの MPS 生起確率を、それまでに出現した 2 値シンボルの値に基づいて推定する。すなわち、所定の確率推定モデルに基づき、2 値シンボルが MPS であった場合は MPS 生起確率を高め、MPS でなかった場合は MPS 生起確率を低下させる制御により、その後出現する 2 値シンボルの MPS 生起確率を更新する。この際、確率更新処理は 2 値シンボルのコンテキスト種類別に独立に実施する。各シンボルのコンテキストは、符号化パラメータの種類や近傍領域の符号化パラメータの値より適応的に選択される。このコンテキスト適応処理により、高精度な MPS 生起確率推定が実現される。

なお、MPS 生起確率の推定が困難とされる 2 値シンボルに対しては、MPS 生起確率を 0.5 に固定した Bypass モードを用いて符号化を行う (Bypass シンボル)。Bypass モードで

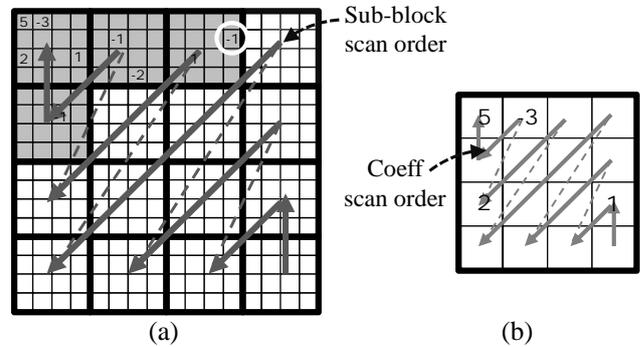


図 1. TU のサブブロック分割およびスキャン順

は符号量の圧縮が行われないものの、確率推定処理のスキップおよび数直線分割処理の演算量低減により高速に処理を実行できる。また Bypass シンボルが連続した場合にはまとめて符号化・復号化処理を実行できる。

3. スループット向上のためのアイデア

3-1. 動きベクトルの符号化

動きベクトルは、時空間近傍の動きベクトルから導出される予測ベクトルとの差分ベクトルとして符号化する。

HEVC では差分ベクトルの水平・垂直成分は以下のプロセスで符号化される。

1. 各成分の絶対値が 0 超であるか否かを示すフラグ、1 超であることを示すフラグを順次符号化
2. 各成分の絶対値が 3 以上の場合、絶対値を符号化 (Bypass シンボルを用いた指数ゴロム符号)
3. 各成分の正負の符号を符号化 (Bypass シンボル)

時間・空間的に近接するブロックの動きベクトルは類似している場合が多いため、差分ベクトルの水平・垂直成分として 0 や 1 など小さな値が出現することが多く、上記のようなプロセスを用いることでトータルの 2 値シンボル数が削減される。また 2 と 3 のプロセスで連続して Bypass モードを用いることでスループットの向上を図っている。

3-2. 係数の符号化

HEVC では、 $4 \times 4 \sim 32 \times 32$ の変換ブロック (Transform Unit: TU) が導入されているため、係数のエントロピー符号化手法も以下のとおり変更された。HEVC では、変換後の TU を図 1(a) に示すように 4×4 のサブブロックに分割し、サブブロック単位に係数を符号化する。サブブロック内の係数は図 1(b) に示す順序でスキャンし、各サブブロックは TU 内で図 1(a) に示す順序で処理される。なお TU が属する予測ブロックがイントラモードである場合、イントラの予測方向に応じてスキャン順を切り替える。係数の符号化は、値が 0 以外の係数 (有意係数) の位置をまず符号化し、その後各有意係数の値を符号化する。符号化のプロセスを以下に示す。

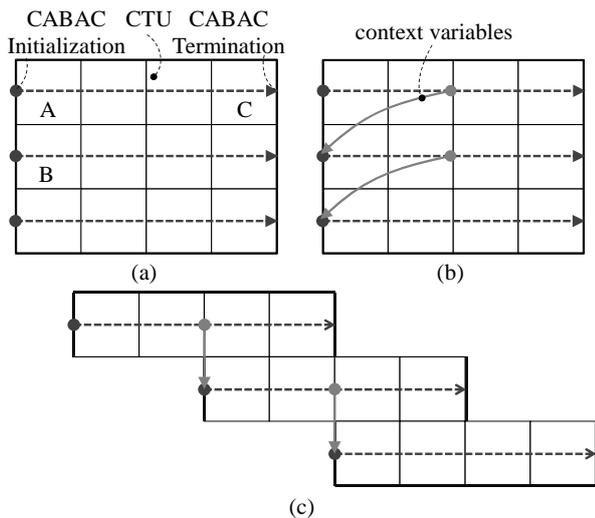


図 2. Wavefront Parallel Processing

1. TU 内においてスキャン順で先頭の有意係数(図 1 (a)の白丸で囲った係数)の位置を符号化
2. 先頭の有意係数よりもスキャン順で後のサブブロック毎に、サブブロックに有意係数が含まれているかを示すフラグを符号化(以下 3~6 は有意係数が含まれているサブブロック(図 1 (a) 灰色のブロック)に対して実施)
3. サブブロック内の各位置の有意係数フラグを符号化
4. 有意係数の絶対値が 1 超であることを示すフラグ, 2 超であることを示すフラグを順次符号化
5. 有意係数の正負符号を符号化(Bypass シンボル)
6. 絶対値が 3 以上の有意係数の絶対値を符号化(Bypass シンボル)

サブブロック分割を行うことで、有意係数位置の有無をサブブロックレベルで伝送し、有意係数を含まないサブブロックは係数情報伝送を省略することが可能となるため、特に大きなサイズの TU で効率的な符号化が可能となる。

なおサブブロック内においては同じ種類のフラグを連続して符号化し、コンテキスト切り替えの頻度を抑制することで、処理分岐の回数が少ない設計となっている。また動きベクトルの符号化と同様に、Bypass シンボルが連続して用いられるように設計されている。

3-3. WPP

HEVC には、CABAC 符号化・復号化処理を符号化効率をできる限り落とすことなく並列実行可能とするために、WPP (Wavefront Parallel Processing) と呼ばれる技術が導入されている。

CABAC は算術符号化の特性上、符号化・復号処理を初期状態から逐次的に進めなければならない、並列的に処理することが困難である。例えば図 2 (a)の CTU (符号化の単位となるブロック)-A と CTU-B から CABAC をそれぞれ並列実行するためには、CTU-B の直前である CTU-C で CABAC を終了させ、CTU-B からは初期状態を開始しなければならない。しかし、CABAC を初期状態にリセットすると、それまでの処理で学習した 2 値シンボルの MPS 生起確率が初期化されてしまうため、符号化効率が低下してしまう。

WPP の概略を図 2 (b)に示す。WPP は、CTU ライン先頭で CABAC を初期化することで並列処理性を実現するとともに、初期化時に上 CTU ラインの途中から学習済み MPS

生起確率を引き継ぐことで、初期化に伴う符号化効率の低下を抑制する。図 2 (c)に示すように、CABAC の符号化・復号化処理を、各 CTU ラインの処理タイミングをずらしながら並列的に実行することが可能となる。

4. H.264/AVC との性能比較

4-1. スループットの向上

HEVC の CABAC では、スループット向上のため Bypass シンボルの使用範囲が H.264/AVC と比較して拡大されている。例えば動きベクトルの要素値の符号化は、H.264/AVC では先頭 9 シンボルを Context シンボル(コンテキスト適応で符号化されるシンボル)とし、その後のシンボルを Bypass シンボルとする。一方 HEVC では、先頭 2 シンボルのみ Context シンボルであり、残りのシンボルは全て Bypass シンボルである。その他イントラ予測モード、参照ピクチャインデックスなど、H.264/AVC の CABAC では Bypass シンボルが用いられていなかったシンタックスで Bypass シンボルが用いられている。

また、コンテキスト選択の際に近傍領域のパラメータ値を参照する処理が H.264/AVC と比較して削減されており、コンテキスト選択処理の並列処理性が向上している。

4-2. メモリ量の削減

HEVC では、エントロピー符号化・復号化に要するメモリ量の最悪値が削減されている。例えば HEVC のコンテキストの種類数は、H.264/AVC と比較して 3 分の 1 程度に削減されており、コンテキスト別に保存される MPS 生起確率情報のためのメモリ量が削減されている[7]。また、上述の通りコンテキスト選択の際の近傍領域パラメータ参照が削減されているため、符号化パラメータを保存するラインメモリサイズも削減されている。CABAC の初期化テーブルパターン数の削減およびテーブル要素の bit 幅削減により、CABAC の初期化テーブルのサイズも削減されている。

5. まとめ

HEVC のエントロピー符号化手法として用いられている CABAC は、他の符号化ツールと合わせて効率的な符号化が行えるように設計されているとともに、H.264/AVC の CABAC と比較して必要メモリ量や演算量の削減が図られている。また符号化効率の低下を抑えつつ CTU ラインレベルの並列処理を可能とする WPP も導入されており、エントロピー符号化・復号化処理のスループット向上が図られている。

参考文献

- [1] Recommendation ITU-T H.265: High efficiency video coding, 2013.
- [2] Recommendation ITU-T H.264: Advanced Video Coding for Generic Audiovisual Services, 2003.
- [3] D. Marpe, et.al, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 620-636, Jul. 2003.
- [4] 村上, 浅井, 関口, 「高効率映像符号化技術 HEVC/H.265 とその応用」, オーム社 (2013)
- [5] 杉本, 服部, 関口, 村上, 「HEVC 映像符号化におけるエントロピー符号化方式の検討」, SITA2012.
- [6] 今井秀樹, 「情報理論」, 昭晃堂 (1984)
- [7] V. Sze, et.al., "High Throughput CABAC Entropy Coding in HEVC," IEEE Transaction on Circuits and Systems for Video Technology, vol. 22, no. 12 Dec., 2012.