可変長符号化

正会員 長谷川まどか[†]

1. ま え が き

デジタル画像を符号化する処理は、大きく分けて、画像信号から冗長性を取り除くために何らかの変換を行う符号化モデル部と、これによって得られたデータ列に対して効率の良い符号語を割当てるエントロピー符号化部から構成されている(図1)。本稿では、後者のエントロピー符号化部に焦点を当てる。

エントロピー符号化部では、符号化対象系列において出現する可能性のあるシンボル(情報源シンボルと呼ぶ)の出現頻度分布をもとに、出現頻度の高いシンボルには短い符号語、出現頻度の低いシンボルには長い符号語というように、可変長の符号語を割当てる。これにより、情報源のシンボルの発生確率により定められる理論的圧縮限界であるエントロピー(平均情報量)に近い平均符号長で、対象系列を符号化できる。

本稿では、可変長符号化に関連するトピックとして、ラン-レベル符号化、Exponential Golomb符号、コンテキスト適応型可変長符号、算術符号化の原理、コンテキスト適応型算術符号について概説する。

AVC/H.264では、変換係数に加え、各種パラメータ、識別子、デリミタ、予測タイプ、動きベクトルなども情報源シンボルに含まれ、それぞれ異なる符号化を行っている。符号化処理の単位となるマクロブロックの集合であるスライスレベルより上位のシンタクスのシンボルには、固定長符号またはExponential Golomb符号が用いられる。スライスレベル以下のシンタクスのシンボルに対しては、変換係数はコンテキスト適応型可変長符号(CAVLC: Context-based Adaptive Variable Length Coding)で符号化し、他のシンボルは、固定長符号またはExponential Golomb符号で符号化するモードや、全シンボルをCABAC(Context-based Adaptive Binary Arithmetic Coding)で符号化するモードが使用されている¹⁾。

図1 画像符号化のブロック図

2. ラン-レベル符号化

離散コサイン変換(DCT)などを用いてブロックベースの変換を行った後に量子化を行うと、得られる変換係数には、値がゼロとなるものが多数出現する。特に、画像の高周波成分は大きな量子化ステップサイズで粗く量子化されるため、値がゼロとなる場合が多い。そこで、変換係数を低域から高域へとジグザグにスキャンし、変換係数を並べ替えると、下記の入力例に示すように、変換係数値0が連続して現れる系列になる。ラン-レベル符号化では、このような系列をゼロラン長とレベルのペアで表現することでコンパクトな表現に置き換える。ここで、ゼロラン長とは非ゼロ係数の前に存在するゼロの数であり、レベルとは非ゼロ係数の値である。

【例】ラン-レベル符号化

入力: 15, 0, 0, -4, 5, 8, 0, 0, 0, 0, 0, -7, ...

出力: (0, 15), (2, -4), (0, 5), (0, 8), (5, -7), ...

得られたラン-レベルの対は、この後さらにエントロピー符号化される。

なお、MPEG-2やMPEG-4などでは、このようにランとレベルを組合せたものに対し符号化を行うが、AVC/H.264では、4章で後述するように、ランとレベルそれぞれに対し異なる可変長符号表を用いて符号化している²⁾。

3. Exponential Golomb符号

Exponential Golomb符号は、出現頻度の高いシンボルに対して短い符号語、出現頻度の低いシンボルに対して長い符号語を割当てる可変長符号である。表1は、符号番号 n に対する符号語の一部である。Exponential Golomb符号の符号

[†] 宇都宮大学 大学院工学研究科

"Basic Technologies of Image Coding (2): Variable Length Coding" by Madoka Hasegawa (Graduate School of Engineering, Utsunomiya University, Utsunomiya)

表1 Exponential Golomb符号の例

符号番号 n	符号語
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
8	0001001
...	...

図2 CAVLCエンコーダの処理の概要

語の構造は次のとおりである：

[Zero prefix] [1] [INFO]

符号語は、 M 個 ($M \geq 0$) のゼロからなる prefix, 1ビットの“1”, M ビットのINFOフィールドから成る。したがって、符号語長は $2M+1$ ビットとなる。シンボル数 n が与えられると、 M とINFOは次式で求められるため、符号語が容易に求められる。

$$M = \text{floor}(\log_2(n+1)) \quad (1)$$

$$\text{INFO} = n + 1 - 2^M \quad (2)$$

一方、シンボル n の復号は以下の手順で行われる。

Step1：1が出現するまで、1ビットずつ読み込み、連続する0の個数 (M) を数える。

Step2：1ビットの“1”を読み込む。

Step3： M ビットのINFOを読み込む。

Step4： $n = 2^M + \text{INFO} - 1$ により、 n を求める。

なお、AVC/H.264では、マッピングタイプごとに、シンボル k と符号番号 n とのマッピング法が定義されており、マッピングを行った後にExponential Golomb符号の符号語を割当てている。

4. コンテキスト適応型可変長符号

AVC/H.264のコンテキスト適応型可変長符号 (CAVLC) では、直近で符号化した変換係数の統計量をもとに適応的にコンテキスト、すなわち、符号化対象の周囲の状況を判断し、複数の符号語表の中から適切な可変長符号表を選択して符号化する。

CAVLCは、変換係数の次の特徴を利用している。

- (1) 変換と量子化を経ると、ブロック内の変換係数は0が多くなる。CAVLCでは、この0の系列をコンパクトに表現するために、ラン-レベル符号化を利用する。
- (2) ブロック内をスキャン後の高周波の非ゼロ変換係数は+1または-1になることが多い。CAVLCでは、これをTrailing1sと呼び、この絶対値1の変換係数の個数と正負の符号を符号化する。
- (3) 隣接するブロック内の非ゼロ変換係数の個数には相

図3 CAVLCにおける情報の種類

関があるため、隣接ブロックの非ゼロ変換係数の個数に応じて符号表を選択し、非ゼロ変換係数の個数を符号化する。

- (4) 非ゼロ変換係数のレベルは、スキャンの始めのDC係数が大きく、高周波になるほど小さい傾向がある。この性質を利用し、CAVLCでは、直近で符号化したレベルの大きさに応じて可変長符号表を切替えて、レベルパラメータを符号化する。

なお、CAVLCは、 4×4 のブロックごとに処理を行うため、 8×8 整数変換を用いている場合は、量子化後の変換係数ブロックを四つの 4×4 ブロックに分割して符号化/復号を行う。

図2にCAVLCエンコーダの処理の概要を示す。ブロック内の変換係数は、ジグザグスキャンにより1次元ベクトル化され、表を用いて可変長符号が割当てられる。可変長符号の表は、隣接ブロックの非ゼロ係数の個数などの局所的性質に応じて適応的に選択される。

CAVLCでは図3に示すように、ブロック内の変換係数

を下記の6種の情報で表現し、スキャンで得られた順序の逆順でそれぞれ符号化を行っている。

- (1) 非ゼロ係数の個数 (TotalCoeff)
- (2) ブロック末尾の絶対値1の係数の個数 (Trailing1s)
- (3) Trailing1sの符号 (+/-) (T1sign)
- (4) 非ゼロ係数値 (level)
- (5) 末尾の非ゼロ係数以前のゼロ係数の個数 (TotalZeros)
- (6) 非ゼロ係数の前のゼロラン長 (run_before)

5. 算術符号の原理

先に述べたラン-レベル符号化, Exponential Golomb符号, CAVLCなどは, ハフマン符号と同様に, 符号化対象シンボルと符号語が1:1に対応するエントロピー符号化法であり, これらは一般にブロック符号と呼ばれる。ブロック符号においては, 符号語はパターンとして扱われ, また, 符号化は符号語の連結(concatenation)によって行われる。これに対し, 符号化対象シンボルと符号語が必ずしも1:1には対応しない非ブロック符号化方式と呼ばれる方法があり, この中の一手法として, 符号語を2進数値として扱い, 算術演算によって符号化を行う算術符号が知られている³⁾。算術符号は, 半開数直線 $[0, 1)$ を通報の出現確率に応じて Augend と呼ばれる領域に分割し, 符号化対象系列に対応する Augend を, 他の Augend と区別できる最小精度の2進数で表現する方法であり, 数直線符号化とも呼ばれている⁴⁾。

図4は, 例として, 2値無記憶定常情報源においてシンボル0および1の出現確率 p_0, p_1 がそれぞれ $p_0=0.7, p_1=0.3$ で与えられる場合に, 2元シンボル5個の系列00101を2値算術符号化する場合の概念図を示している。まず, 半開数直線 $[0, 1)$ を2元シンボル (0, 1) の出現確率に応じて0.7:0.3に分割し, シンボル0および1に対する Augend をそれぞれ生成する。最初に符号化すべきシンボルは0であるので, このシンボルに対応する Augend (0.7と記されたグレーの領域) が選択される。次に, この選択された領域が確率比0.7:0.3に分割されるが, 続いて符号化されるシンボルも0であるので, 先頭からのシンボル系列00に対する Augend は, 図中 $0.7^2=0.49$ と記されたグレーの領域に狭められる。このように, シンボルの符号化が進むにつれ Augend の分割と選択が再帰的に行われ, 最終的にシンボル系列00101に対応する Augend は, 図のやや濃いグレーで示された領域となり, この中の1点を確定できる最小精度の2進数値が符号となる。

この算術符号化処理は, 図5に示すステップで表すことができる。ここで, A, A_0, A_1, C はそれぞれ V ビットのレジスタであり, 先頭に小数点が付けられた2進小数とみなすこととしている。 A, A_0, A_1 は, Augend レジスタ, C は符号レジスタと呼ばれる。

図5において, ステップ(3)が算術符号作成部であるが,

図4 算術符号化の原理

- (1) 初期設定
 $A = .11 \dots 1$ (all 1のビット列)
 $C = .00 \dots 0$ (all 0のビット列)
 - (2) Augendレジスタ値のセット
 $A_1 \leftarrow A \times P_1$
 $A_0 \leftarrow A \times P_0 = A \times (1 - P_1) = A - A_1$
 - (3) シンボル x_i の符号化と Augend レジスタの変更
 $x_i = 0$ ならば $C \leftarrow C, A \leftarrow A_0$
 $x_i = 1$ ならば $C \leftarrow C + A_0, A \leftarrow A_1$
 - (4) 正規化処理 (Renormalization)
 A レジスタのMSB(最上位ビット)が0ならば, 1になるまで A レジスタを左シフトする。同時にここでシフトしたビット数だけ C レジスタも左シフトする。
このとき, C レジスタからあふれたビットが符号出力用の待機バッファに出力される。
- (2) ~ (4) の処理を繰り返す。

図5 算術符号化処理

図4との関連でいえば, これは, 符号を, 選択された Augend の下端部にセットすることと等価である。算術符号ではこのように, 符号化対象シンボルごとに逐次 Augend の分割と選択が再帰的に行われ, そのたびに乗算と加算が行われるため, 無限精度の演算が必要となり, これが算術符号実現上の大きな問題点となる。このため, 符号化効率の低下を許容して, 正規化処理 (Renormalization) により浮動小数点演算を行い, 演算を有限桁で打ち切ること, さらには, 乗算をシフト演算や加算への置き換えなど,

処理の簡易化を図ることにより実現を図っている。上記(4)の正規化処理が浮動小数点演算を行う部分である。さらに、符号作成は2進小数の加算で行われるため、桁上がりによる上位ビットへのビット反転が波及する場合がある。この影響を避けるため、符号出力用の待機バッファを監視し、1があるビット数連続して出現したら強制的にビット0を挿入するビットスタッフィング、あるいは、バイトスタッフィングの処理が行われる。

さて、JPEGやMPEG、AVC/H.264で対象としている画像情報は多値信号であるが、多値情報源をそのエントロピーの増大なしに2値系列に分解することが可能であるため、AVC/H.264など多値信号を扱う場合においても2値算術符号による符号化が行われる。2値シンボルに対して高い符号化効率を得るためには、複数のシンボルをまとめて符号化対象通報として扱う情報源拡大処理が必須であるが、ブロック符号を用いてコンテキスト適応型の符号化を行う場合、コンテキスト毎に別々に情報源拡大を行う必要がある、これは符号設計上の大きな負担となる⁴⁾。

この点に関し、算術符号は、符号化シンボル1個ごとにAugendすなわちシンボル出現確率を変更することが可能であり、情報源拡大を意識することなく効果的な圧縮が可能であり、画像情報源のようなコンテキスト適応型の符号化が要求されるような場合のエントロピー符号化方式として、極めて有用な方式である。

6. コンテキスト適応型算術符号

算術符号は、その性質上コンテキスト適応化が原理的に可能である。本章では、コンテキスト適応型算術符号の系譜について概説する。

6.1 確率値の2のべき乗近似による算術符号

算術符号の画像信号符号化への最初の応用は、Langdon-Rissanenらによる2のべき乗近似型算術符号に見られる⁵⁾。彼らは劣勢シンボル(LPS: Less Probable Symbol)の出現確率 p_1 を2の整数べき乗 2^{-Q} で近似し、優勢シンボル(MPS: More Probable Symbol)のそれについては $1-2^{-Q}$ とすることにより、乗算フリーの算術符号を提案している。このような近似を行った場合、LPSが出現すれば必ず正規化処理が行われ Q ビットが符号として出力され、また、MPSが出現すれば平均で約 $1/[-\log(1-2^{-Q})]$ 回ごとに正規化処理が行われ1ビットシフトされるから、その平均符号長 L は、

$$L = P_1 Q - (1 - P_1) \log_2(1 - 2^{-Q}) \quad [\text{bit/symbol}]$$

与えられることとなる。

ファクシミリ2値画像を対象に、10画素参照マルコフモデルに適用することにより、当時のファクシミリ国際標準可逆符号化方式であるMR(Modified READ)と比較して20～30%の圧縮率改善を得ている。

6.2 画像符号化国際標準方式における算術符号

画像符号化国際標準方式に用いられている算術符号は表2

表2 画像符号化国際標準方式における算術符号

算術符号の種類	国際標準方式
QMコード	JBIG, JPEG
MQコード	JBIG2, JPEG2000
CJMコード	JPEG-LS(part-2)
CABAC	AVC/H.264

に示すとおりであり、これらの源流はQコード⁶⁾に見ることができる。以下では、Qコード、QMコード、MQコードおよびCABACについて簡単に説明する。

Qコード

Qコードは、テーブル駆動型算術符号であり、高速化かつ適応化を実現した方法である。Qコードでは、図5の(2)におけるAugend演算式に乗算を用いず、単なる置数と減算

$$A_1 \leftarrow Q_e, A_0 \leftarrow A - A_1$$

で済ませている。ここで Q_e はLPSの出現確率の推定値であり、これを数種類テーブルの形で保存しておき、正規化処理のたびに異なる Q_e 値に遷移することにより動的適応化を図っている。

QMコード

QMコードでは、Bayes推定に基づく確率推定手法を簡易化した手法を用いることにより、初期段階における確率推定の高速化を図っている。また、減算型算術符号では、 A_0 が A_1 より小さくなってしまふ場合が起こり得るため、これを防ぐために、条件付き交換と呼ばれる手法を採用している。さらに、復号側におけるキャリーオーバ処理(演算により最上位桁より上の桁への繰り上がりが発生した場合の処理)を簡易化するために、ビットスタッフィングではなくバイトスタッフィングを採用している。

MQコード

MQコードは、JPEG2000におけるエントロピーコードEBCOT⁷⁾(Embedded Block Coding with Optimized Truncation)に用いられている手法であり、QMコードと類似した手法である。Qコードでは29種類の Q_e 値を用いていたが、QMコードでは113種の、また、MQコードでは47種の確率推定値を用いており、さらに、ビットスタッフィング処理にも工夫を施している。

CABAC

AVC/H.264のメイン/ハイプロファイルにおけるオプションの高圧縮エントロピー符号化方式としてCABACが標準化されている。図6にCABACのブロック図を示す。

CABACは、次の基本ステップにより構成されている。

(1) 2値化(Binarization)

符号化対象となるシンタックス要素(Syntax Element: SE)は、変換係数や動きベクトルなど必ずしも2値信号であるわけではない。一方、算術符号は、2値シンボルの符号化を対象としているため、これら多値信号は算術符号化

図6 CABACのブロック図

に先立って2値に変換される。変換には、前節で述べたようなExponential Golomb符号などを用いて可変長2値化され、その後算術符号化される。

(2) コンテキストモデリング (Context Model Selection)

あらかじめ用意されている複数の確率モデルの中から、すでに符号化されたデータ系列にしたがって着目データの符号化に使用するコンテキストモデルを選択する。

(3) 算術符号化 (Arithmetic Coder)

CABACの算術符号化部における確率推定は、LPSに対して64種の確率状態間で遷移させることにより行われる。なお、出現確率が均一に近いデータの場合には、エントロピー符号化を省略して処理の高速化を図るバイパスモードも設けられている。

7. む す び

本稿では可変長符号化に関連するトピックとして、ランレベル符号化、Exponential Golomb符号、コンテキスト適

応型可変長符号、算術符号化の原理、コンテキスト適応型算術符号とその系譜について概説した。紙面の制約上、省略した点も多い。詳細については、参考文献等をあたりたい。

(2012年9月23日受付)

〔文 献〕

- 1) I.E. Richardson: "The H.264 Advanced Video Compression Standard", pp.52-56, pp.207-221, Wiley (2010)
- 2) 大久保榮, 角野真也, 菊池義浩, 鈴木輝彦: "H.264/AVC教科書", pp.144-149, インプレス (2005)
- 3) J. Rissanen, G.G. Langdon Jr: "Arithmetic Coding", IBM J. Res. Develop., 3, 2, pp.149-162 (1979)
- 4) 小野文孝: "算術符号化とその画像符号化への適用に関する考察", 画電誌, 31, 5, pp.745-754 (2002)
- 5) G.G.Langdon Jr, J. Rissanen: "Compression of Black-White Images with Arithmetic Coding", IEEE Trans. on Comm., COM-29, 6, pp.858-867 (1981)
- 6) W.B. Pennebaker, J.L. Mitchell, G.G. Langdon Jr, R.B. Arps: "An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder", IBM J. Res. Develop., 2, 6, pp.717-726 (1988)
- 7) D. Taubman: "High performance scalable image compression with EBCOT", IEEE Trans. on Image Processing, 9, 7, pp.1158-1170 (2000)

長谷川まどか はせがわ 1999年、宇都宮大学大学院博士後期課程修了。現在、宇都宮大学大学院工学研究科情報システム科学専攻准教授。画像符号化、画像信号処理に関する研究に従事。博士(工学)。正会員。