

## ■2群 (画像・音・言語) -5編 (画像符号化)

# 7章 符号割り当て

(執筆者：米山暁夫) [2013年3月 受領]

符号割り当てでは、量子化レベルや変換係数などの情報に「0」、「1」の符号を効率良く割り当て、冗長な符号の削減をすることでより少ないビット量でより多くの情報を格納することを指す。

符号化方式の基本的な分類としては固定長符号 (Fixed Length Coding) と可変長符号 (Variable Length Coding) がある。固定長符号化は、シンボルごとへの割り当てる符号の長さを一定とした符号化方法である。一般に符号化効率は可変長符号よりも低下するが、固定長であるため、データ途中からの復号が可能となる利点もある。可変長符号は、割り当てる符号の長さを可変とする符号化方法である。発生確率の低い事象に長い符号を、発生確率の高い事象に短い符号を割り当てることにより、高い符号化効率を実現することができる。ハフマン符号化と算術符号化が代表である。

あるシンボル列は、各シンボルの発生確率により、エントロピー (平均情報量) が定まり、どのような符号化手法を採っても平均符号長はこのエントロピー値を下回ることはいできない。エントロピー符号化とは、平均符号長をこのエントロピーに近づける符号化のことを指す。現在は計算機の処理能力向上によりエントロピー値に限りなく近づく符号割り当て技術が主流となっている。

## ■2 群 - 5 編 - 7 章

### 7-1 ハフマン符号化

(執筆者：米山曉夫) [2013年3月 受領]

ハフマン符号化は、1952年にD. A. Huffmanにより開発されたシンボル出現確率に基づく符号化手法である<sup>1)</sup>。発生頻度の動的更新を行わずに符号化する手法を静的ハフマン符号(Static Huffman Coding)と呼び、シンボル発生により発生頻度を更新してハフマン符号を割り当て直す手法を動的ハフマン符号(Adaptive Huffman Coding)と呼ぶ。

#### 7-1-1 ハフマン符号化の原理

ハフマン符号化の原理は、はじめに対象データ列に出現するシンボルごとの個数を求め、そのシンボルが木構造の末端(葉)に相当するとみなし、末端より根幹に向けて木を構成する。二つの葉または節点を束ねると新しい節点が生成できるものとする。

まず、すべての葉を並べ、個数が最小のものと2番目に小さいものを取り出し、それらを束ねる新しい節点を生成する。このとき、新しい節点の値は、両方の葉の個数の和とする。

次に、この新しい節点を含め、残りの葉と共に並べ、最小の個数の葉または節点と、2番目に小さい葉または節点を束ね、新しい節点を生成し、その節点の値として束ねられた二つの葉または節点の値の和をおく。以上を繰り返して根節点まで到達することで木が完成される。

更に、根から順に枝分かれする左右に0と1の値を割り振っていく。すると、それぞれの葉(シンボル)に対して、一意にビット列が与えられる。この記号とビット列の関係をもとに、もとのデータの記号をビット列に変換していくことで符号化が行われる。

#### 7-1-2 シンボル集中

ハフマン符号化は、発生確率に偏りのあるデータ列を符号化した場合に符号化効率を向上させることができる。そのため、符号化対象となるデータ列に発生するシンボルそれぞれの発生確率に著しい偏りをもたせることが重要となる。例えば画像では、DCTを用いて小ブロック単位の画像を周波数成分に変換することで、低周波数成分にエネルギーを集中させることや、動きベクトルの符号化では周囲のブロックのもつ動きベクトルとの相関が高いことを利用し、周囲の動きベクトルから生成される動き補償予測ベクトルとの差分値を符号化することでベクトル値を0付近に集中させることなどを行っている。

#### 7-1-3 ランレングス符号化

データ列を可逆圧縮する符号化の一手法で、連続する同一のシンボルの列を、その連なり(Run)の長さ(Length)を示す数字に置き換えて表現するものをランレングス符号化と呼ぶ。同じ値が続く場合には有効な手段である。例えばFAXのように、スキャンされたデータ列は白または黒の値が連続で並ぶ確率が高いため、非常に高い符号化効率を得られる。

ランレングス符号化方式は、画像の符号化においては、例えばジグザグスキャンされたDCT係数を表現する際に、ゼロ係数の連続回数をこのランで表現している。更に、このランやシンボル値をハフマン符号により表現することもできる。その際にもランとシンボル値を別々にハフマン符号化する考え方や、ランとシンボル値の組み合わせを一つの事象としてと

らえてハフマン符号化する考え方も存在する。

#### 7-1-4 Golomb-Rice (ゴロム・ライス) 符号

幾何分布に従って出現する整数の符号化に最適な整数符号化手法である。ゴロム符号と類似の手法にライス符号があるが、ゴロム符号の特殊な場合がライス符号となるため、ライス符号の事をゴロム・ライス符号と呼ぶことが多い。特にライス符号は符号化・復号の演算量が少ないことが特徴となっている。符号化効率は、シンボルが幾何分布のときはハフマン符号と同一となり、それ以外の場合ではハフマン符号に劣る。

#### 7-1-5 CAVLC (Context-based Adaptive Variable Length Coding)

日本語ではコンテキスト適応型可変長符号化と呼ぶ。符号化効率としては CABAC (7-2-2) よりも劣るとされているが、ハフマン符号化をベースとした手法であり、具体的には指数ゴロム符号 (Exponential Golomb Coding) を用いている。CABAC と比較すると非常に高速な処理が可能である。例えば CAVLC を用いた DCT 係数の表現では、隣接マクロブロックの DCT 係数の状態に依存して現在の対象マクロブロックの符号化に使用する符号化テーブルを切り替える。このように切替えを行うことがコンテキスト適応型の名の由来となっており、対象画像のテクスチャなどの特徴に応じた符号化テーブルが使用でき、より短い符号への圧縮が期待できる。

## ■2 群 - 5 編 - 7 章

### 7-2 算術符号化

(執筆者：米山暁夫) [2013年3月 受領]

算術符号化は、シンボルの出現確率を計算しながら符号化テーブルを作成し、符号割当をする動的な方式のため、多大な計算処理を必要とするが、画像データの特徴に応じた最適な符号割当ができる。このため、最近の計算処理の高速化にともない、固定した符号化テーブルを使用するハフマン符号化などより符号化効率の優れ、シンボルの平均符号長の下限值(エントロピー値)に近い圧縮効果が得られることから注目されている。

算術符号を動画画像符号化方式に組み合わせて使用する方法が開発されており、DCT 係数の量子化出力を算術符号で符号割当てする手法や、ウェーブレット変換の変換係数を算術符号で符号割当てする手法、オブジェクト符号化に算術符号を組合せた手法などが開発されている。

#### 7-2-1 算術符号化の原理

ハフマン符号化では、ひとつのシンボルに対する可変長符号の割当てから、複数のシンボルの組合せに対してひとつの可変長符号を割り当てる手法により更に効率的な圧縮を実現しているが、この考えを更に進めると、対象データ列をひとまとまりのシンボルの組合せと考え、ひとつの可変長符号で表現するという手法が 1960 年頃に MIT の P. Elias によって提案され、IBM の G. G. Langdon, J. J. Rissanen により完成された<sup>2)</sup>。

この手法は、符号化と復号化の過程で算術演算を利用するため、算出符号と呼ばれている。算出符号の考え方は、元データを 0 から 1 までの実数区間の一つの点として表現するというものである。

算術符号化はハフマン符号化と比較して高い符号化効率を得るとされているが、以下の特徴をもつ。

- (1) 復号処理の終了点が判別できないため、データ列の最後端にデータの終わりを表す特殊な記号「End-Of-File: EOF」を付加したり、あるいはあらかじめ総シンボル数を伝えたりする必要がある。
- (2) 符号値を適当な有効桁数に留める手法と浮動小数点演算を行う際の丸め誤差の影響を取り除く手法が必要となる。

#### 7-2-2 CABAC (Context-based Adaptive Binary Arithmetic Coding)

CABAC は、算術符号化とコンテキスト適応符号化を組み合わせたことで、ほぼ理論限界まで理想的なエントロピー符号化を可能とした手法である。

まず、符号化すべき多値のシンタックスデータエレメントを、可変長の 2 値化シンボル系列に変換する。次に、符号化すべきデータの周囲の情報や既に符号化された 2 値化シンボル値に基づいて、2 値化シンボル 1 ビットごとにコンテキスト計算を行う。そして、計算されたコンテキスト情報に応じて、算術符号化の確率テーブルを選択し算術符号化処理を行うとともに、確率テーブルの更新を行う。このように、CABAC 符号化では 2 値化シンボル 1 ビットごとに複雑な処理が必要となり、また現ビットの符号化が終わらないと次のビットの符号化に進めないため、並列化処理が困難である。

## ■2 群 - 5 編 - 7 章

### 7-3 DCT 係数符号化

(執筆者：米山曉夫) [2013年3月 受領]

DCT は、画素情報を周波数情報に変換することで発生情報の偏りをもたらずため、符号化効率の向上の観点から重要な技術となる。2 次元 DCT を施した後の出力である 2 次元 DCT 係数は、DC 係数と AC 係数を別々に符号化することが一般的である。

- (1) DC 係数 (0, 0) : 直前のブロックの値との差をカテゴリーの符号と値 (+極性) を表すビット列で表現する。また、標準の符号化テーブルは決まっている。
- (2) AC 係数 : 2 次元データ低周波数から高周波数へと進むジグザグスキャンにより 1 次元化したうえで、Run-Level の 2 次元可変長符号化を施す。以下に説明する。

#### 7-3-1 2 次元可変長符号化

例えば、 $8 \times 8$  DCT 係数の場合は一つの DC 成分と 64 個の AC 成分が出力される。この係数を低周波数から高周波数に向けてジグザグにスキャンした 1 次元シンボル列は、一般的に低周波数領域においては非ゼロ係数が、高周波数領域ではゼロ係数が多数発生する。そこで、このジグザグスキャン順に並んだ DCT 係数について、その非ゼロ係数に対して「先行するゼロ係数の個数 (Run)」と、「非ゼロ係数値 (Level)」の組合せ一つのシンボルとして可変長符号を割り当てる手法を 2 次元可変長符号化と呼ぶ。ただし、ジグザグスキャン化されたデータ列の後半はゼロ係数が連続することが多いため、データ列の最後の非ゼロ係数を明示的に示す EOB (End-Of-Block) という専用の符号を用いて符号化効率を上げている。

#### 7-3-2 3 次元可変長符号化

上記 2 次元可変長符号化から更に拡張し、「先行するゼロ係数の個数」、「非ゼロ係数値」、「最後に出現する非ゼロ係数か」の三つの情報の組合せを符号化する手法を 3 次元可変長符号化と呼ぶ。これにより、ジグザグスキャン後の DCT 係数列の後半はほぼゼロ係数となる特性を効率的に表現することができる。

## ■2群 - 5編 - 7章

### 7-4 動きベクトル符号化

(執筆者：米山曉夫) [2013年3月 受領]

MPEG-2 や H.264 などの動画像符号化では、動き補償予測符号化が行われる。これは符号化対象フレームを矩形ブロック（マクロブロック）単位に分割し、そのブロックごとに以前のフレームで最も類似する領域（参照領域）の位置と、その参照領域との画像の差分を表現することにより符号化効率を向上させるものである。この参照領域の位置と符号化対象ブロックとの位置関係を動きベクトルにより表現する。また、動きベクトルの精度（最小単位）は動画像符号化規格により異なっている。新しい規格ほど詳細な表現が可能となっている。

#### 7-4-1 ベクトル予測

ブロック単位で動き補償を行う際に発生する動きベクトル情報は、周囲のブロックがもつ動きベクトルと高い相関をもつため、周囲のブロックから対象ブロックの動きベクトルの予測を行い、対象ブロックの動きベクトルそのものを符号化する代わりに、この動き補償予測ベクトルと動きベクトルとの差分のみを符号化することで動きベクトル情報の符号量削減を行っている。

動き補償予測は、対象ブロックと周囲のブロックのサイズや位置関係により様々なパターンに分類される。基本的な手法は、対象ブロックの左や左上、上に隣接するブロックの動きベクトル値から、垂直成分と水平成分について別々に中央値を算出し、それを動き補償予測ベクトルとし、この動き補償予測ベクトルとの差分量についても、垂直成分と水平成分を別々にハフマン符号で符号化する。更に、垂直成分と水平成分の相関性を利用して片方からもう一方の成分を予測する手法も存在する。

## ■2群 - 5編 - 7章

### 7-5 エラー耐性

(執筆者：米山暁夫) [2013年3月 受領]

本章では、符号化効率を向上させる符号割当手法を述べてきたが、様々な効率的な符号化方式を利用することで圧縮率を向上させたとしても、すべての符号化データが正しく復号処理に渡らなければ適切な符号化ができない。例えば、符号化映像データの伝送時にデータの一部が欠損してしまったり、蓄積された符号化映像データを読み出した際にデータ誤りが混入してしまったりした場合などが想定される。このような場合にも映像の復号再生に与える影響を抑える工夫を総称してエラー耐性技術と呼ぶ。以下に代表的なものを説明する。

#### 7-5-1 同期ワード (再同期マーカ)

可変長符号化により符号化されたデータ列は、データ列の先頭から正しく復号して行く場合には正しい情報の復元が可能だが、復号中にひとたびエラーを生じると、そこから以降の復号は不可能となる。これは可変長符号化を利用した場合には避けられない現象である。

そこで、符号化データ列内に再同期マーカを適切に配置し、復号処理中にエラーを生じた際には、その後最初に出てくる再同期マーカまでの処理をスキップし、その再同期マーカ位置から復号処理を再開することを可能とする。なお、再同期マーカの直後にはマクロブロックの画面内の位置情報や量子化パラメータ情報など復号再開時に必要な情報を挿入する。

#### 7-5-2 データパーティショニング

符号化された動画のデータ配置は、画像1枚単位の情報を格納するヘッダ部分に続き、動きベクトル情報とDCT係数情報がマクロブロック単位に繰り返し並ぶことになるが、データパーティショニングを用いた場合には、再同期マーカの直後に各マクロブロックの動きベクトル情報だけを先に並べ、その後ろにDCT係数情報を並べる形態をとる。動画の主観品質に対しては動きベクトル情報の与える影響の方がDCT係数と比較して大きいことから、この動きベクトル情報の正しい復号を行うことができる確率を高めるために、再同期マーカの直後に動きベクトルをまとめて配置させたものである。

#### 7-5-3 リバーシブルVLC

リバーシブル可変長符号による符号割当技術は、ハフマン符号にビットを付加することで、ビット列を順方向で復号するだけでなく、逆方向でも正しい復号を行うことを可能とする技術である。例えば1枚の画像の途中でデータの欠損があった場合、それ以降の次の再同期マーカまでのデータの復号は不可能となってしまうが、リバーシブルVLCを利用して符号化した場合には、次の再同期マーカを検出した点から遡って復号していくことが可能となり、復号エラー領域を最小化することができる。

#### ■参考文献

- 1) D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. IRE, 40, pp.1098-1101, Sep. 1952.
- 2) G. G. Langdon, "An Introduction to arithmetic coding," IBM Journal of Research and Development, 28, 2, pp.135-149, Mar. 1984.