

講座：基礎からの画像符号化 [全12回] 開講にあたって

編集幹事 山田悦久

画像符号化技術を本学会誌の講座に前回取り上げたのは、1989年6月号から1990年6月号にかけて連載した「画像情報圧縮の手ほどき」である。JPEGやH.261、MPEG-1などの国際標準化作業が進められていた頃であり、講座をまとめた書籍「先端技術の手ほどきシリーズ『画像情報圧縮』」は多くの版を重ねた。

その後、静止画ではJPEG2000など、動画ではMPEG-2 | H.262やMPEG-4 AVC | H.264などの国際標準方式やデファクトスタンダード方式が世の中に登場した。現在、画像符号化技術はテレビ放送、ビデオレコーダ、静止画・動画カメラ、PC、携帯電話など、われわれの生活において不可欠な要素技術の一つとなっている。

この20数年の間、画像符号化技術の進展として、静止画符号化方式には先の書籍には記載されていない新しい技術としてWavelet変換が使われているが、動画画像符号化方式の基本構成は当時のままである。今年、2013年の春頃に成立する見込みの新しい動画画像符号化方式「HEVC (High Efficiency Video Coding)」も基本構成は同じであるが、複雑な演算処理や適応処理を取り入れることによって、MPEG-2ではSDTV解像度を対象としていた4～5Mbps程度のビットレートでも、HDTVを符号化できる性能にまで達しつつある。

今回の講座では、このHEVCの規格成立に合わせて、主に若手ならびに専門外の読者を対象に、あらためて画像符号化技術の基礎から解説する連載を試みるものである。本講座では、全12回を各号で2回ずつ掲載して半年間にわたって取り上げる。

なお、本講座は、甲藤二郎前編集理事、吉田俊之編集理事、上倉一人論文部門委員長、伊藤典男、山下幸彦両編集企画幹事と私山田が担当する。

予定目次 (全12回)

1月号	(第1回) 画像信号の基礎 (第2回) 可変長符号化	吉田俊之 (福井大学) 長谷川まどか (宇都宮大学)
2月号	(第3回) 画像符号化のための線形変換 (第4回) 変換係数の量子化	山下幸彦 (東京工業大学) 稲積泰宏 (富山大学)
3月号	(第5回) フレーム内予測 (第6回) 映像符号化におけるループ内フィルタ	松田一朗 (東京理科大学) 松尾翔平 (NTT)
4月号	(第7回) フレーム間予測 (第8回) 視点間予測・視点捕間	児玉 明 (広島大学) 石川彰夫 (情報通信研究機構)
5月号	(第9回) 符号化処理の階層構造 (第10回) 符号化装置の制御技術	市ヶ谷敦郎 (NHK) 坂手寛治 (三菱電機)
6月号	(第11回) 静止画像符号化に関わる標準規格 (第12回) 動画符号化に関わる標準規格	石川孝明 (早稲田大学) 数井君彦 (富士通研究所)

画像信号の基礎

正会員 吉田 俊之†

1. ま え が き

(A) 2013年春に、「HEVC」と呼ばれる新しい画像符号化方式が成立する予定で、それを受けて、本誌ではHEVCの技術動向を取り上げる特集を予定している。この「基礎からの画像符号化講座」では、「HEVCの技術動向を取り上げる特集」を理解するための基礎知識として、画像符号化の要素技術と基礎をやさしく解説することを目的とする。「基礎からの画像符号化講座」は6回で構成され、各回2名の計12名の執筆者が分担して担当する。要素技術として、1回目に画像信号の基礎と可変長符号化の説明を行い、2回目は直交変換と量子化を解説、3回目と4回目は予測符号化の内容を扱う、5回目では画像符号化アルゴリズムの構成を概観した後、最後の6回目に、JPEG方式、JPEG2000方式、MPEG-1方式、MPEG-2方式、MPEG-4方式、AVC/H.264方式などの静止画像符号化方式と動画画像符号化方式を概説する。

冒頭から「何かたどたどしい」と感じられたかも知れない。(A)のパラグラフにはまわりくどい表現が散見され、本来は次のような簡潔な文章に推敲すべきであろう。

(B) 本誌は、「HEVC」と呼ばれる新しい画像符号化方式の特集を予定している。本講座では、「HEVC特集」に備えた基礎知識として、画像符号化の要素技術と基礎をやさしく解説する。今回からの6回を10名程度の執筆者が担当し、画像信号の基礎、可変長符号化、直交変換と量子化、予測符号化、および符号化アルゴリズムの構成について解説した後、最後にJPEG/JPEG2000、MPEG-1/-2/-4、AVC/H.264等の実際の静止画/動画の符号化方式を概観する。

これから解説する「画像符号化」は、この例のような「文章を推敲して簡潔にする過程」に非常に似通っている。「簡

† 福井大学

"Basic Technologies of Image Coding (1): Introduction to Image and Video Signals" by Toshiyuki Yoshida (Faculty of Engineering, University of Fukui, Fukui)

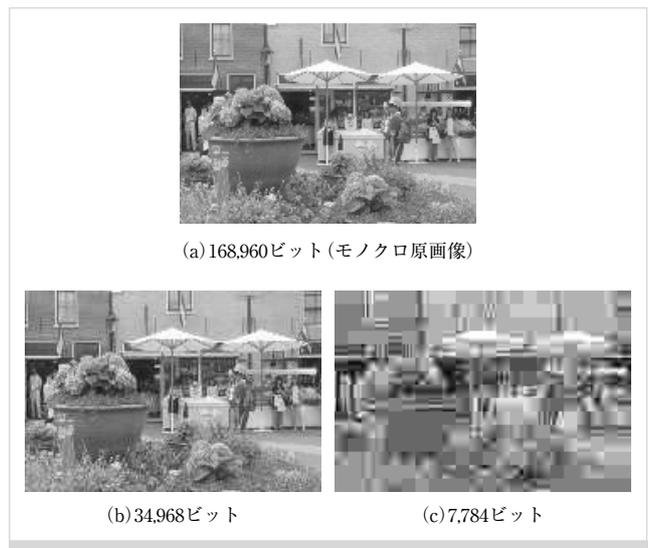


図1 原画像(a)を圧縮符号化した結果と符号量

潔に」という観点からは、(B)をさらに

(C) 本講座では、画像符号化の基礎を解説する。可変長符号化や直交変換等を解説した後、実際の静止画/動画の符号化方式を概説する。

程度まで短縮できるが、これでは本講座の意図や概要が伝わらず、「簡略化し過ぎ」であろう。

一方、図1は、原画像(a)を圧縮符号化・復号した画像と符号量の例である。図1(b)は、図1(a)の符号量の1/5程度に圧縮されているが、図1(a)とは見分けがつかず、図1(a)に対して充分忠実である。図1(c)は図1(a)の1/22程度まで圧縮されているが、劣化が激しく、これは「圧縮し過ぎ」であろう。

図1(a)～図1(c)の各画像は、上記(A)～(C)の各文章に対比でき、(A)と図1(a)が「オリジナル」、それを「適切に圧縮したものが(B)および図1(b)」、「やり過ぎ」が(C)と図1(c)である。文章を推敲する目的の一つに「なるべく少ない文字数で、なるべく忠実に伝えること」があるが、画像符号化の目的も同様で、「なるべく少ない符号量で、なるべく忠実に伝えること」と言える。しかも、「文章の推



敵」と「画像符号化」では、その基礎となる考え方や方針、あるいは「要素技術」は、ほぼ同じと言えるのである。実際、上記の文章(A)から(B)への推敲過程は、

- (1)「1回目に～の説明を行い」～「3回目と4回目は～を扱う」の箇所は、同じような表現の繰り返しを避ける。「MPEG-1方式, MPEG-2方式, . . .」の部分においても重複を避け、「MPEG-1/-2/-4」とする。
- (2)「基礎からの画像符号化講座」や「HEVCの技術動向を取り上げる特集」等の頻繁に現れる用語は、「講座」、「特集」といった略称により簡潔に表現する。
- (3)「2013年春」や「各回2名で計12名」の部分は、読者にとって必要以上に細かい情報のため、省略するか、概数を示すに留める。

等の方針に基づいている。「画像符号化の要素技術」は、以上の3つを数学的に実現したものと言え、各々、

- (1) 相関性の除去：同じような値の重複を避ける
- (2) エントロピー符号化：出現頻度が高い値には短い符号を与える
- (3) 量子化：必要以上に高い数値精度を下げる

処理に対応している。ここでは、本論に入る前準備として、「符号化の対象を知ること」と「用語の整理」を目的に、画像信号の基礎について解説する。

2. 画像信号の基礎¹⁾²⁾

対象物体から出た光をレンズを通して合焦させると、スクリーン上に結像する(図2)。この「像」をフィルムに焼き付けるとアナログ画像が、また撮像素子を用いて記録するとデジタル画像が生成される。「デジタル」とは「数値の」という意味で、デジタル画像は「数値として記録された画像」を指す。図2のスクリーン上の点 (x, y) における「像」の明るさ(輝度)を $f(x, y)$ と表そう。

図3(a)は、 $f(x, y)$ を波形として表した例である。注意すべきは、座標 x, y は任意の実数値を取り得ることで、したがって、スクリーン上の任意の点 (x, y) に対して輝度値 $f(x, y)$ が一つ定まる。これは、「像」が無数個の点の輝度で構成され、「像」を完全に記録するにはスクリーン上のすべての点(無限個の点)の輝度を記録する必要があることを示している。また、輝度値自体も実数であるため、ある一点に着目しても、その厳密な記録には無限の桁数が必要となる。このような「無限」を扱う困難を回避するため、デジタル画像を生成する際は、次に示す空間サンプリングと量子化が適用される。

2.1 空間サンプリング

われわれの目の解像力には限界があり、「像」に存在する非常に細かいパターンをすべて記録しても、実際には見ることはできない。空間サンプリングは、この性質を利用して、対象となる「像」を有限個の離散点上で記録する操作。すなわち、 i, j を整数として、離散点



図2 画像生成の原理

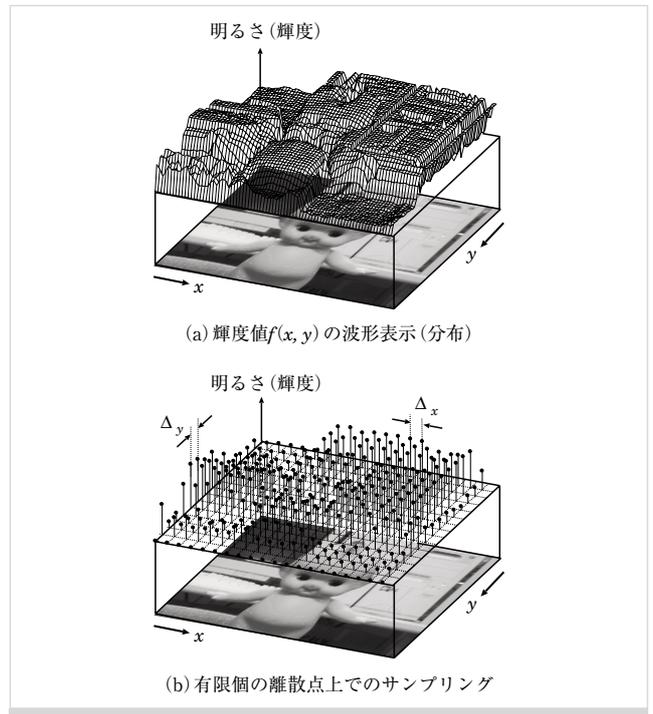


図3 輝度 $f(x, y)$ の波形表示とサンプリング操作

$$(x, y) = (i\Delta_x, j\Delta_y) \tag{1}$$

上で輝度値 f を記録する操作を指す(図3(b))。ここで、 Δ_x (Δ_y) は x (y) 方向のサンプリング間隔である。

空間サンプリングによって生成されるサンプル値配列は、元の「像」 $f(x, y)$ の情報の一部に過ぎないが、次のサンプリング定理が示す条件(2)が満たされれば、サンプル値配列から原波形 $f(x, y)$ が完全に復元できる。つまり、サンプル値配列と原波形は対等と言えることになる。

$f(x, y)$ のフーリエ変換を $F(f_x, f_y)$ とする。条件

$$F(f_x, f_y) = 0 \quad \left(\left| f_x \right| > \frac{1}{2\Delta_x}, \left| f_y \right| > \frac{1}{2\Delta_y} \right) \tag{2}$$

が成立すれば、サンプル値配列 $f(i\Delta_x, j\Delta_y)$ から、原波形 $f(x, y)$ が完全に復元できる。

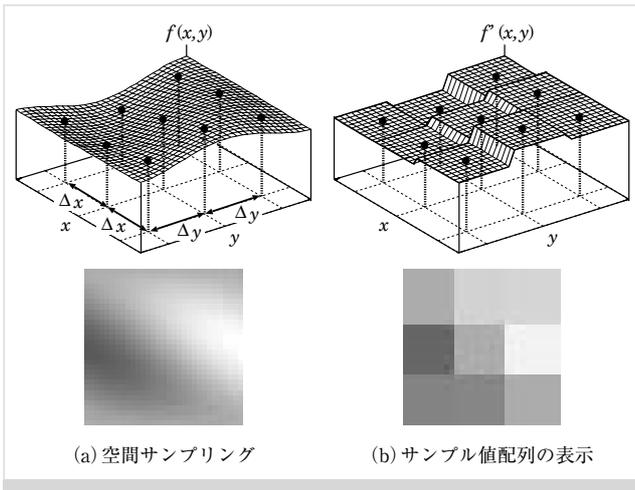


図4 空間サンプリングと画像表示

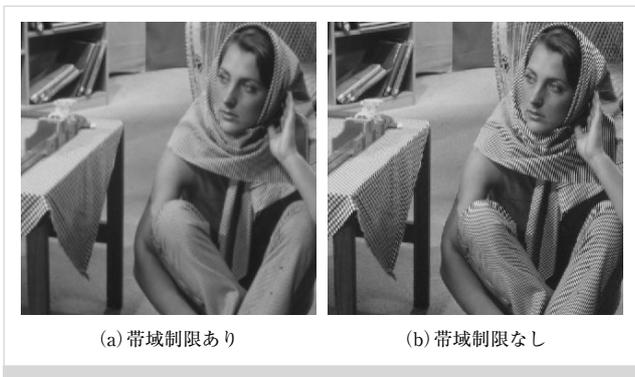


図5 サンプリングにおける帯域制限あり/なしの影響

図4(a)は、条件(2)を満たす原波形 $f(x, y)$ とそれを Δ_x, Δ_y で空間サンプリングしたサンプル値配列の例で、この場合は、サンプリング定理が保証する通り、サンプル値配列から原波形 $f(x, y)$ が完全に復元される。一方、条件(2)が満たされない場合、すなわち、 $f(x, y)$ が“ $2\Delta_x, 2\Delta_y$ よりも細かいピッチのパターン(周波数成分)を含む場合”は、式(2)を満たすように細かいパターン(高周波成分)をカットした後にサンプリングする必要がある。この操作を帯域制限という。図5は、帯域制限あり/なしを比較した例で、帯域制限なし(図5(b))では、原画像には含まれないパターン(縞模様)が現れる妨害(折り返し歪みと呼ぶ)が生じる。サンプリングを行う際は、折り返し歪みが生じないように、きちんと帯域制限を施す必要がある。

以上は空間サンプリングに対する基本的解説であるが、実際の画像は、サンプリング定理に完全に基いて生成あるいは表示されている訳ではない。図4(a)の下段は、サンプル値配列から復元した原波形 $f(x, y)$ を画像として表示している。と、言いたいが、ここに矛盾が潜んでいることに気づかれるだろうか。復元した原波形 $f(x, y)$ の x, y は実数で、すでに指摘したように $f(x, y)$ は無数の点で構成さ

れるが、これを有限個のドットで表現(印刷)することは不可能である。同じことは、例えば、 $f(x, y)$ を液晶のようなディスプレイ上に表示する際にも言え、縦横有限個のドットで構成されるディスプレイ上で連続な波形を表示することはできない。

つまり、サンプリング定理に基づいて原波形を復元しても、それを表示することは不可能なため、実際には図4(b)に示すように、各サンプル値 $f(i\Delta_x, j\Delta_y)$ をそのまま一区画として表示しており、各サンプル点を「画素、ピクセル(pixel)」、または「ペル(pel)」と呼ぶ。われわれが実際に目にするデジタル画像は、図4(b)のような「カクカク」の画像で、「カクカク感」を低減するには十分な画素数が必要となる。撮像素子が画素数を追求してきたのはこのためである。

一方、撮像素子における実際の空間サンプリングは、CCDやCMOS等の撮像素子上に配置されたフォトダイオード(PD)(入射光量を電荷量に変換する素子)によって実現される。撮像素子の感度を上げるには各PDの面積をなるべく大きくする必要があるが、式(1)が要求するように、面積ゼロの1点でサンプリングするには無限小のPDが必要となり、ここにも定理と実際との乖離が存在する。実際の有限面積を持つPD上では、入射光はその面積上で積分され、これが“細かい縞模様”(高周波数成分)を減衰(コントラストを低下)させるため、補正が必要となる³⁾。ただし、画像符号化の準備の目的からは、その詳細にまで立ち入る必要はなく、ここでは空間サンプリングの基礎とともに、“実際の空間サンプリングはサンプリング定理のみで完結しない”という事実を知れば充分である。

2.2 量子化

空間サンプリング後の画素値は実数で、そのまま記録することは非現実的である。例えば、画素値123.45に対し、小数2桁目の5を切り捨てても、われわれの目ではまったく検知できない。1桁目を四捨五入して120としてすら影響は少ないだろう。1章の(3)で触れたように、量子化とは、数値精度を下げることでデータ量を削減する操作を指し、空間サンプリング後の画素値は、有限のビット長で表現可能なように量子化される。

最も基本的な量子化処理を図6に示す。入力値 x (実数)の取り得る範囲を $n=0, 1, \dots, N-1$ の N 個の区間に分割し、各々に量子化代表値 x_n (通常は区間の中央値)を割当てて。量子化の際は、入力 x が属する区間のインデックス n を出力とする。一般に n は2進数で表現されるため、区間数 N は適当な整数 L を取って $N=2^L$ と設定することが多く、これを「 L ビット量子化(器)」と呼ぶ。以上により、入力の実数値 x は L ビットの2進数に対応付けられる(2進 L ビットで表現される)ため、記録に要するデータ量が削減される。

画素値を量子化する際は、撮像素子から出力される「最も暗い値」と「最も明るい値」の間を等間隔に 2^L [段、階調]に分割し、 L ビット量子化を行う。 L が小さいと階調数が

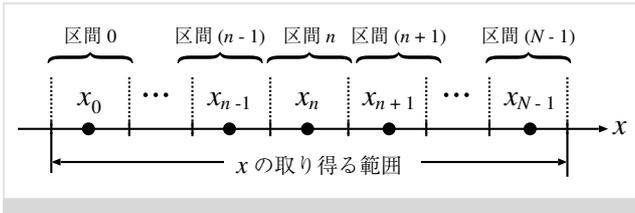


図6 量子化処理の例

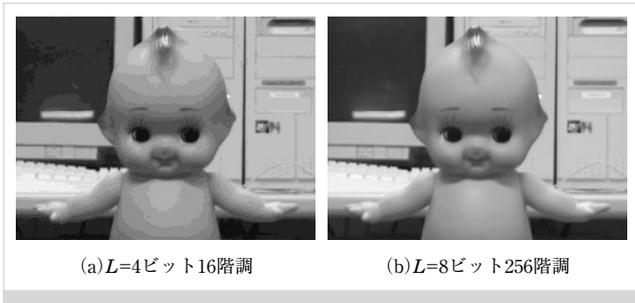


図7 量子化ビット数による影響

不足し、例えば図7(a)のおでこやお腹に見られる「偽輪郭」が生じる原因となるため、実際には相応に大きく選ぶ必要がある。通常は $L=8$ ビット256階調が用いられる(図7(b))。高精細画像では、8ビットを越えるビット深度が用いられることもある。

なお、図6における各区間幅を「量子化ステップ幅 Δ 」と呼び、各区間で Δ が一定の場合を「一様量子化」、 Δ が区間に依存する場合を「非一様量子化」という。後者は、符号化アルゴリズム内で変換係数を量子化する際等に用いられる。

2.3 静止画像と動画像

これまで、時間軸については固定して考えてきたが、各点の輝度 $f(x, y)$ は時間とともに変化するため、本来は $f(x, y, t)$ と表すべきである。静止画像とは、時刻 t を固定し、その瞬間に生成される単一のデジタル画像を指す。図1(a)は、8ビット/画素、 176×120 画素の静止画像で、全ビット量を計算し確認されたい。

一方の動画像は、一定の時間間隔 Δ_t で連続的に撮像した画像の集合を指し、各単一画像を「フレーム」、 Δ_t を「フレーム間隔」、その逆数を「フレームレート」と呼ぶ(図8(a))。ここで、「一定間隔で連続的に撮像する」とは、 $f(x, y, t)$ を時間方向に間隔 Δ_t でサンプリング(時間サンプリング)することに相当する点に注意されたい。

2.4 順次走査と飛び越し走査

テレビ放送がアナログ方式であった時代は、走査と呼ばれる処理を利用し、輝度分布を1次元波形に変換してアナログ伝送していた。走査とは、各フレームの輝度分布 $f(x, y)$ を垂直 y 方向に走査線に分解する操作を指すが、これは本質的に垂直方向のサンプリングに他ならず、実際には垂直方向のサンプリング方式と考えてよい。

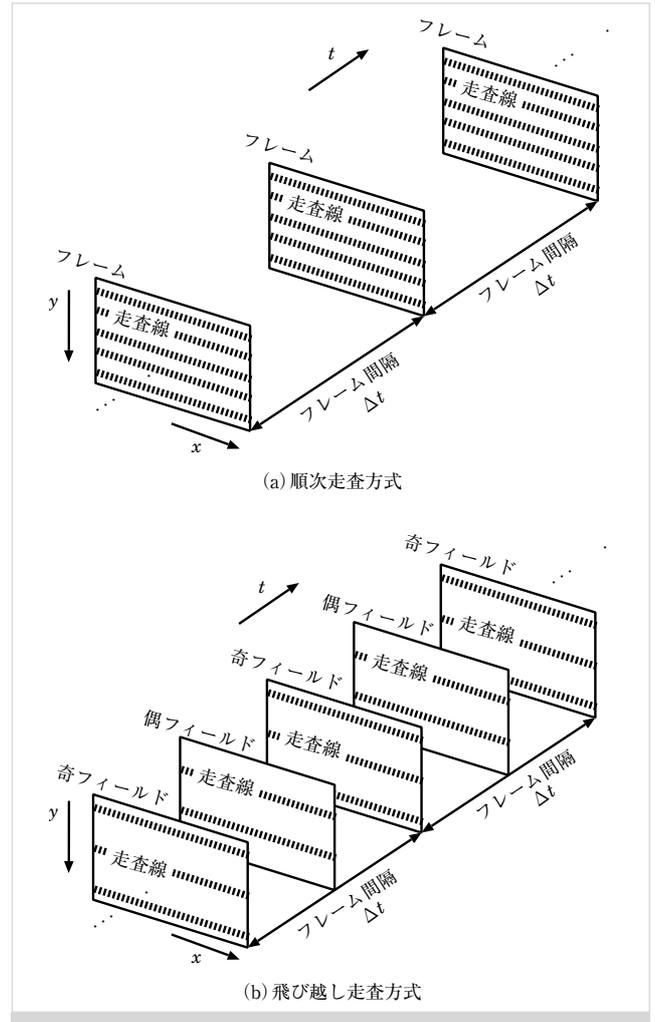


図8 順次 (progressive) 走査と飛び越し (interlace) 走査

一般に、動画像内の動きは、フレーム間隔 Δ_t を小さく選ぶことによって滑らかに表示されるが、これではフレーム数が増加し、記録・伝送すべき情報が増大してしまう。情報量を抑えつつ動きを滑らかに表示する手法として考え出されたのが、飛び越し (Interlace) 走査方式である。

図8(a)は、輝度 $f(x, y, t)$ を間隔 Δ_t で時間サンプリングしてフレームを生成する方式で、これを「順次 (Progressive) 走査方式」と呼ぶ。一方の図8(b)は、走査線を奇数番目と偶数番目に分け、各々(「奇フィールド、偶フィールド」と呼ぶ)を時間軸に沿って交互にサンプリングする方式で、「飛び越し (Interlace) 走査方式」と呼ぶ。

順次走査(a)と飛び越し走査(b)を比較すると、両者は単位時間あたりに生成される走査線数は同じである一方、後者は見かけ上、時間方向のサンプリング間隔が半分になっている。すなわち、飛び越し走査は、順次走査と同じ情報量で、より滑らかな動きを再生できるため有効な走査方式と考えられ、歴史的経緯から、現在のデジタル放送においても利用されている。例えば、現状の1080iと呼ばれる動画像フォーマットは、走査線数1,080本の飛び越し走査



図9 カラー画像におけるRGB各成分の例

方式を指す。

一方で、図8から判るように、飛び越し走査の走査線(サンプリング)構造は複雑なため、後の回に随時説明されるように、その符号化アルゴリズムが複雑化することは避けられない。また、飛び越し走査は時間 t -垂直 y 平面内での「五の目状サンプリング」に相当し、サンプリング理論に基づく厳密な帯域制限を施すことは困難であるため、折り返し歪みによる画質の低下も指摘されている⁴⁾。

2.5 色の表現と画像フォーマット

本節では、カラー画像と色の表現等について触れるが、ここでは紙面の制約上、“カラー画像の各画素はR(赤)、G(緑)、B(青)の3成分によって表される”ことを出発点とする(詳細は文献3)等を参照されたい)。

図9は、カラー画像のRGB各成分をモノクロ画像として表示した例で、各成分間には高い類似性(相関性)があることがわかる。1章の(1)で触れたように、圧縮符号化の観点からは、「類似したもの、相関があるもの」を重複して扱うことはムダが多く、回避する必要がある。RGB成分間の相関性については、後の回で扱う直交変換等を用いて除去し圧縮効率を上げることも可能であるが、実際には人間の視覚特性に基づく巧みな手法を利用している。

われわれは、色を見る際に「色合い」だけでなく、その「明るさ(輝度)」も感じ取っている。各画素の輝度値を Y で表すと、 Y はRGB成分から次式によって求められる³⁾。

$$Y=0.299R+0.587G+0.114B \quad (3)$$

さらに、色差成分と呼ばれる

$$C_b=0.564(B-Y), C_r=0.713(R-Y) \quad (4)$$

を作ると、 C_b は「青み成分」、 C_r は「赤み成分」に対応する。

一方、われわれの視覚特性には“輝度に対する感度は高く、色に対する感度は低い”という性質がある。これを利用すると、画質を維持しながら、輝度 Y に対して色差 C_b ,

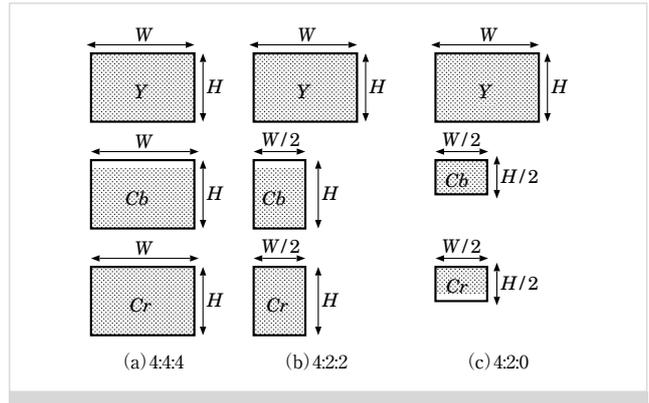


図10 カラー画像のフォーマット

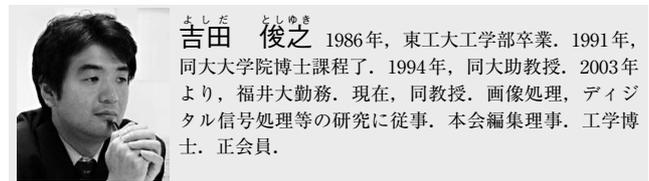
C_r の解像度(画素数)を落すことができる。図10に示すように、色差成分 C_b 、 C_r の画素数を、水平方向にのみ1/2に削減したものを4:2:2形式、水平垂直両方向とも1/2に削減したものを4:2:0形式と呼んでいる⁵⁾。画像符号化では、対象となるRGBカラー画像を、まず4:2:0または4:2:2形式に変換した後に、 Y 、 C_b 、 C_r 各成分を独立に符号化することで、上記のムダを除去している。

3. むすび

本稿では、対象を知ることと用語の整理を目的に、画像信号の基礎について解説した。以降では、1章の(1)~(3)各項目を要素技術として具体的に解説し、画像符号化アルゴリズム全体の説明へと進んでいく。(2012年10月1日受付)

〔文献〕

- 1) 酒井善則, 吉田俊之: “映像情報符号化”, オーム社(2001年)
- 2) 貴家仁志: “画像情報符号化”, コロナ社(2008年)
- 3) 日下秀夫: “カラー画像工学”, オーム社(1997年)
- 4) 吹抜敬彦: “テレビ信号の多次元信号処理”, 日刊工業新聞社(1988年)
- 5) 山岸秀一: “知っておきたいキーワード4:4:4と4:2:0”, 映情学誌, 62, 10, pp.1542-1546 (Oct. 2008)



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⁾。

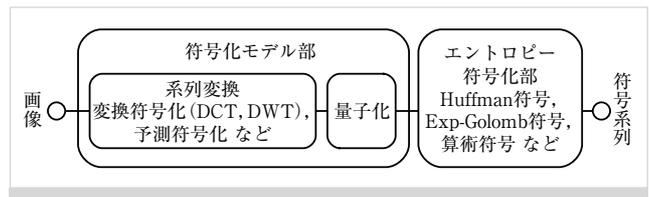


図1 画像符号化のブロック図

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
...	...

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

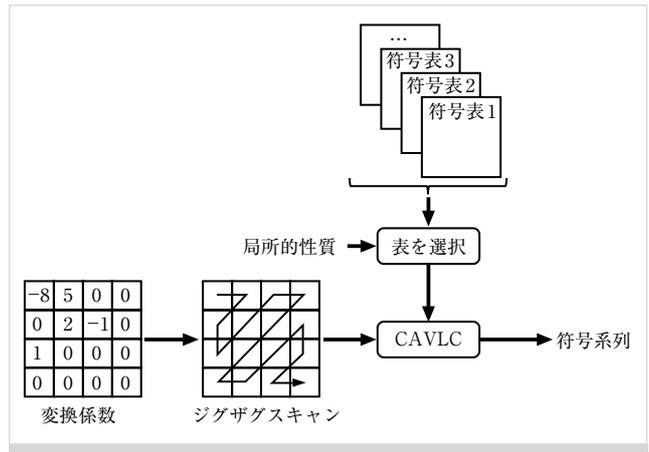


図2 CAVLCエンコーダの処理の概要

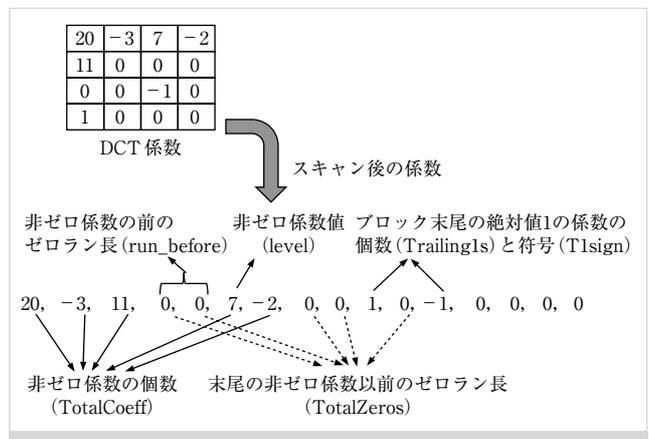


図3 CAVLCにおける情報の種類

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, 先頭に小数点が付せられた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,

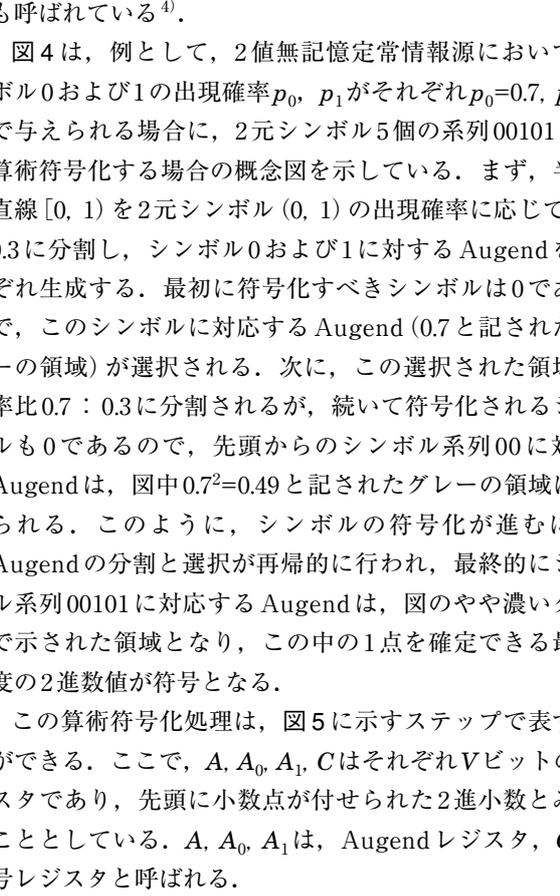


図4 算術符号化の原理

- (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, 演算を有限精度で打ち切ること, 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可能性がある。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、

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

と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⁾に見ることが出来る。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

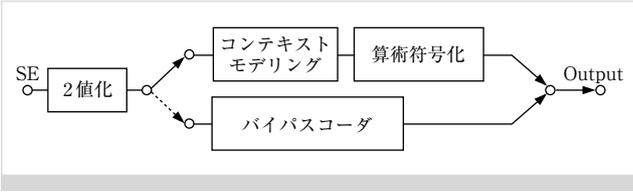


図6 CABACのブロック図

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)



はせがわ 長谷川まどか 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。



画像符号化のための線形変換

正会員 山下幸彦†

1. ま え が き

静止画や動画などの画像データはテキストデータなどと比べて、データ量が非常に多い。したがって、画像符号化では、画像のある一定の画質に保ちながら、保存や伝送のためのデータ量が少ない、すなわち、符号化効率が高い方式を開発することが重要である。そのため画像符号化では、数値をビット列に変換するときにデータ量を削減するエントロピー符号化(ハフマン符号化や算術符号化)の他に、予測や、画像の輝度信号や色差信号または予測誤差の集まりに対して、線形変換を行うことによってデータ量を削減する。画像符号化のための線形変換としては、変換の重要性を広く知らしめた楨本らによるスラント変換¹⁾、現在の標準の変換である離散コサイン変換(Discrete Cosine Transform: DCT)²⁾、ブロック歪みが原理的に生じない離散ウェーブレット変換(Discrete Wavelet Transform: DWT)^{3) 4)}などが知られている。DCTは、画像符号化の標準方式であるJPEG, MPEG-1, MPEG-2, MPEG-4, H.264/MPEG-4AVCに採用されており、現在の画像符号化の中心的な変換となっている。DWTは、静止画像を最も高い効率で符号化できるJPEG2000で採用されている。本稿では、第2章で線形変換によってデータ量を削減できる理由とKarhunen-Loève変換を説明し、第3章で離散フーリエ変換(DFT)からDCTを導出するとともにDCTを説明し、第4章でDWTを解説する。そして、第5章で本稿をまとめる。

2. 線形変換によるデータ量の削減

2.1 平均情報量の削減

一つの連続的な値 x を符号化することを考える。ここでは、 x は平均 μ 、分散 σ^2 の正規分布に従うものとする。確率密度関数は、

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

で与えられる。連続的な値は無限の情報量を持っているため、符号化することはできない。そこで、本講座第4回で解説する量子化を行う。ここでは最も単純な量子化である、量子化ステップ q に対して、 x が区間

$$I_k = \left(\left(k - \frac{1}{2}\right)q + \mu, \left(k + \frac{1}{2}\right)q + \mu \right) \\ (k = 0, \pm 1, \pm 2, \dots)$$

に含まれている場合、量子化値を k とするものを考える。そして、この k をビット列に変換する。復号器で、ビット列から量子化値 k を得ることはできるが、元の値 x を厳密に得ることはできない。ここでは、 x の復号値を $qk + \mu$ とする。この操作を逆量子化と呼ぶ。このとき、量子化・逆量子化による誤差は、 $\pm \frac{q}{2}$ 程度であるので、 q を充分小さくすれば、実用的に x を符号化することができる。

いま、量子化値が k となる確率を $P(k)$ とすれば、

$$P(k) = \int_{I_k} p(x) dx$$

となる。 q が充分小さい場合、 $x \in I_k$ ならば $p(x)$ を $p(qk + \mu)$ で一定と近似できるため、

$$P(k) \simeq qp(qk + \mu) \quad (2)$$

となる。次に、このビット列の平均的な長さ、すなわち、平均的なデータ量を最小にすることを考える。そのための条件は、シャノンの情報源符号化定理によって与えられており、量子化値 k に対して、 $-\log_2 P(k)$ 程度の長さのビット列を割当てればよい。したがって、量子化値の平均的なデータ量(ビット)は、

$$H = - \sum_{k=-\infty}^{\infty} P(k) \log_2 P(k) \quad (3)$$

となる。この H は、エントロピー(平均情報量)と呼ばれる。ここでは、量子化値に対する最適な符号化が行われるものとし、 H を符号化によって出力されるデータ量と考える。式(2)より、

$$H \simeq - \sum_{k=-\infty}^{\infty} qp(qk + \mu) \log_2 (qp(qk + \mu)) \quad (4)$$

が成立する。 q が充分小さいとして、この和を積分で近似する。 $y=qk$ とおき、式(4)の最初の q を dy と考えることができるので、 H は次式で近似できる。

† 東京工業大学 大学院理工学研究科

"Basic Technologies of Image Coding (3): Arithmetic Coder" by Yukihiko Yamashita (Graduate School of Science and Engineering, Tokyo Institute of Technology, Tokyo)



$$\begin{aligned}
 H &\simeq -\int_{-\infty}^{\infty} p(y+\mu) \log_2(qp(y+\mu)) dy \\
 &= \int_{-\infty}^{\infty} p(y+\mu) \left(\frac{y^2}{2\sigma^2} - \log_2 q + \log_2 \sigma \right. \\
 &\quad \left. + \frac{1}{2} \log_2(2\pi) \right) dy \\
 &= \frac{1}{2} \log_2 \sigma^2 - \log_2 q + \frac{1}{2} (\log_2(2\pi) + 1)
 \end{aligned} \tag{5}$$

すなわち、一つの連続値を符号化したデータ量は、分散の対数の1/2から量子化ステップの対数を引き、ある定数を加算したものになる。画像は、画素における明るさなどの値(画素値)が複数集まったものである。カメラの方向は任意であるため、画素の位置によって、画素値の確率的な構造は変わらないと仮定できる。画素値をそのまま並べて符号化すれば、データ量は式(5)に画素数をかけたものになる。画像を見ると、隣接する二つの画素値はあまり変わらないことが多いことがわかる。図1は、この状況を図示したものである。s₁とs₂が二つの画素値を表し、図1の黒点が1組のデータ(s₁, s₂)を表している。そして、図1の黒点の分布は、s₁とs₂が近い値をとることが多いことを示している。このとき、(s₁, s₂)-座標系を使って表す場合と、それを回転させた(t₁, t₂)-座標系を使って表す場合とでは、データ量はどちらが少なくなるであろうか。以下では、その優劣に関して数学的に説明する。

画像の扱いを容易にするために、画像のN個の画素値を一列に並べて、N次元列ベクトルfで表す。そして、fを変換することを考える。変換として、ここでは線形変換だけを考える。線形変換とは、あるベクトルに定数をかけたものの変換結果が、もとのベクトルの変換結果に同じ定数をかけたものになり、二つのベクトルの和の変換結果がそれぞれの変換結果の和になるものである。図1の座標回転も線形変換によって表すことができる。ベクトルの要素の2乗や異なる要素の乗算をするような変換は、非線形変換と呼ばれる。一般に、非線形変換を使えば線形変換に限るよりもデータ量を少なくすることができるが、その最適な非線形変換を求めることも実装することも難しいため、本稿では扱わない。

線形変換は、fに行列Uをかけることによって実現できる。変換後のベクトルの各成分を量子化するときの量子化

ステップはすべて同じ値qとする。量子化ステップの値を変えることは、変換後のベクトルの各成分にそれぞれ定数をかけることと同じであり、一つの線形変換で表すことができるため、このように仮定しても一般性を失わない。画像を復号するときは、逆量子化で得たベクトルを逆変換して画像を得る。このとき、逆量子化で得たベクトルの各要素には、原画像を変換したベクトルの要素に対して0から±q/2の誤差が生じる。ここでは、簡単のため、すべての逆量子化で得たベクトルが一定の誤差を持っていると仮定する。画像符号化の目的は、復号画像の原画像に対する誤差を抑えながら、データ量を削減することである。そのためには、誤差が原画像によって変化しない方が良い。したがって、原画像によって復号画像の誤差が変わらない線形変換を考える。

実際の画像符号化では、下に述べるように、人間の視覚特性を考慮するため、原画像によって誤差は異なる。しかしながら例えば、前処理を施すことによって許容される誤差が原画像によって変わらないようにできるため、原画像によって誤差が変わらない変換に限ることができる。

誤差が原画像によって変わらないようにするためには、Ufのノルムがfのノルムの定数倍である必要がある。この条件を満たすUは、直交(実ユニタリ)行列の定数倍となるが、定数倍は量子化係数を変えること等価であるので無視して、変換行列Uは直交行列であるものとする。単位行列をIで表すとき、直交行列Uは、UU^T=U^TU=Iを満たす行列として定義され、ベクトルの回転(画像の回転ではなく)や対称操作を表す。

fに関する期待値操作をE_fで表す。fの平均E_f[f]をf̄で表す。また、ベクトルや行列に対する転置を・^Tで表す。fとf̄の第i成分を、それぞれf_iとf̄_i(i=1, 2, ..., n)で表せば、fの共分散行列Rの第(i, j)-成分R_{ij}は、

$$R_{ij} = E_f(f_i - \bar{f}_i)(f_j - \bar{f}_j) \tag{6}$$

で定義される。R_{ii}はf_iの分散であり、R_{ij}(i≠j)はf_iとf_jの共分散である。共分散R_{ij}はf_iとf_jが一緒に大きくなったり小さくなるとき正の大きな値をとり、f_iとf_jがばらばらに変動するとき0に近い値をとる。したがって、画像上で二つの画素が近い場合、それらの画素値の共分散は正の大きな値をとり、離れると0に近い値になる。

分散R_{ii}を使えば、ベクトルfを符号化したデータ量は、それぞれ要素の分散に対する式(5)のN個の和

$$\frac{1}{2} \sum_{i=1}^N \log_2 R_{ii} - N \log_2 q + \frac{N}{2} (\log_2(2\pi) + 1)$$

となる。行列Uで変換したベクトルの共分散行列は、R'=URU^Tとなる。したがって、変換しない場合を基準にした変換を行ったことによるデータ量の変化は、

$$\frac{1}{2} \sum_{i=1}^N \log_2 (R')_{ii} - \frac{1}{2} \sum_{i=1}^N \log_2 R_{ii} \tag{7}$$

となる。したがって、この値が小さくなるようにUを選べば良い。また、trでトレース(対角和)を表せば、トレース

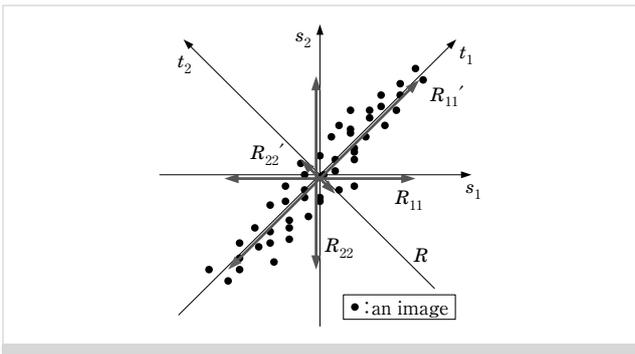


図1 平均情報量の削減



は可換であるから

$$\sum_{i=1}^N (R')_{ii} = \text{tr}[URU^T] = \text{tr}[RU^T U] = \text{tr}[R] = \sum_{i=1}^N R_{ii} \quad (8)$$

となり、分散の和は直交変換によって変わらない。式(8)の条件のもとに、式(7)を小さくするためには、 \log_2 関数は上に凸な関数であるから、変換 U によって得られるベクトルの各成分の分散が均一であるよりも、「できるだけ少数の成分の分散を大きくし、残りの成分の分散を小さくした方がよい。」すなわち、図1の場合は、 (t_1, t_2) で表す方が、 (s_1, s_2) で表すよりもデータ量を小さくすることができる。

直交行列 U の中で、式(7)を最小にするものは、Karhunen-Loève変換で与えられる。これは、 U の列ベクトルが、共分散行列 R の固有ベクトルとなっている場合である。ここで、Karhunen-Loève変換を使った場合のデータ量の変化を計算する。先に述べたように、画像の近隣の画素間の共分散は大きい。ここでは、1次元信号を考え、共分散行列が

$$R_{ij} = \rho^{|i-j|} \quad (9)$$

の形で与えられるとし、 $\rho=0.95$ とする。これは、隣の画素値との共分散が0.95、その隣の画素値とは 0.95^2 で、遠くなるほどべき乗で共分散が小さくなることを意味している。 $N=64$ とすれば、式(7)の値は、 -50.4 ビットとなり、データ量を1画素あたりの平均で $50.4/64=0.79$ ビット程度削減することができる。

2.2 視覚特性上重要でない情報の削減

線形変換を使えば、人間視覚に対して重要な部分と重要でない部分を切り分けることができる。一般に、画像におけるゆっくりとした変化を表す画像の低周波成分の誤差は、すぐに知覚されるが、激しい変化を表す高周波成分の誤差は知覚されにくい。また、量子化係数を小さくすれば誤差が小さくなるが、量子化後の値やその分散が大きくなり、式(5)よりデータ量が増加する。逆に、量子化係数を大きくすれば誤差は大きくなるが、データ量は減少する。したがって、低周波成分に対して量子化係数を小さく、高周波成分に対して量子化係数を大きくすれば、データ量を保ちながらも人間が見たときの画質を向上させることが可能になり、符号化効率が向上する。

分散の偏りと視覚特性とは別のものであり、そのための線形変換も異なるものであるべきであるが、現在の画像符号化では、空間周波数をもとにした同じ変換を用いている。そのため、先に述べた前処理による許容誤差を原画像によらなくすることと、量子化ステップを変えることが等価になる。

3. 離散コサイン変換 (DCT)

3.1 離散フーリエ変換 (DFT)

N 個の値からなる1次元信号 $x(m)$ ($m=0, 1, \dots, N-1$)を考える。この信号をDFTした結果を $X(u)$ ($u=0, 1, \dots, N-1$)とおけば、DFTと逆DFTは、

$$X(u) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} x(m) e^{-\frac{2\pi i}{N} mu} \quad (10)$$

$$x(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} X(u) e^{\frac{2\pi i}{N} mu} \quad (11)$$

と表される。オイラーの公式 $e^{\theta} = \cos\theta + i \sin\theta$ より、DFTは原信号を周期が N の整数分の1のsin関数とcos関数で分解していることがわかる。

上の式で、 u が小さい/大きい場合の $X(u)$ は、信号の低/高周波成分の大きさを表している。一般に、画像の低周波成分は大きく、その分散も大きい。逆に、高周波成分は小さく、その分散も小さい。したがって、DFTによって、信号の分散を低周波成分に集中することができるため、もとの画素値を符号化するよりもデータ量を小さくすることができる。

DFTは信号を周波数成分に分解することができるが、画像符号化の線形変換に使われることはほとんどない。その理由を説明する。信号の性質が画像のどの部分でも同じ、すなわち、ある点の値の、その周囲の点の値に対する条件付き確率密度が一定、あるいは、信号が定常である場合、変換する信号の長さ N はできるだけ大きくした方が符号化効率が高くなる。しかしながら、一般に画像は定常ではなく、また N を大きくすると計算量が大きくなるという問題のため、画像符号化では N を比較的にな小さな値に設定する。例えば、 N として4や8という値が使われる。この場合長い信号から長さ N の部分で切り出して、線型変換を施すことになる。例えば、図2(a)から切り出した、図2(b)の信号を変換していることになる。このとき、DFTは図2(c)の周期信号を変換していると考えることができる。この信号には、元の信号には存在しなかった、大きな段差が生じている。この原因は、切り出した信号の右端と左端における値が違うためである。したがって、DFTを用いると切り出しによって実際には存在しない高周波成分が生じ、低周波分に信号の分散が集中しなくなるため、符号化効率が低下する。

3.2 DFT から DCT へ

図2(d)のように、切り出した $x(m)$ を線対称にしたものを隣に置いた、長さ $2N$ の信号(周期 $2N$ の周期信号)を考える。

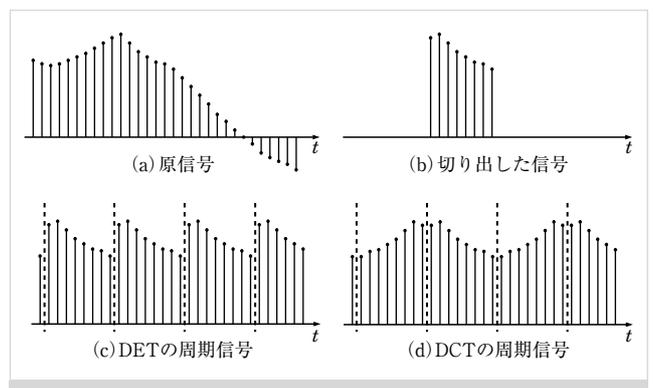


図2 DFTとDCT



$$y(m) = \begin{cases} x(m) & (0 \leq m \leq N-1) \\ x(2N-1-m) & (N \leq m \leq 2N-1) \end{cases} \quad (12)$$

$y(m)$ から長さ $2N$ の DFT によって得られる $Y(u)$ は、

$$Y(u) = \frac{1}{\sqrt{2N}} \sum_{m=0}^{2N-1} y(m) e^{-\frac{2\pi i}{2N} mu} \quad (13)$$

$$y(m) = \frac{1}{\sqrt{2N}} \sum_{u=0}^{2N-1} Y(u) e^{\frac{2\pi i}{2N} mu} \quad (14)$$

となる。ここで、 $C(l) (l=0, 1, \dots, N-1)$ を、

$$C(l) = \begin{cases} 1 & (l=0) \\ \sqrt{2} & (l \neq 0) \end{cases} \quad (15)$$

とおき、

$$Y'(u) = Y(u) \frac{C(u)}{\sqrt{2}} e^{-\frac{\pi i}{2N} u}$$

とおけば、

$$Y'(u) = \frac{C(u)}{2\sqrt{N}} \sum_{m=0}^{2N-1} y(m) e^{-[\frac{\pi i(2m+1)u}{2N}]} \quad (16)$$

$$y(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{2N-1} \frac{Y'(u)}{C(u)} e^{[\frac{\pi i(2m+1)u}{2N}]} \quad (17)$$

という変換と逆変換を考えることができる。ここで、式 (16) で加算される項の m に $2N-1-m$ を代入すれば

$$\begin{aligned} y(2N-1-m) e^{-[\frac{\pi i\{2(2N-1-m)+1\}u}{2N}]} \\ = y(m) e^{[\frac{\pi i(2m+1)u}{2N}]} \end{aligned} \quad (18)$$

となり、式 (16) の和を 0 から $N-1$ と N から $2N-1$ に分け、式 (18) の対称性を後者の和に用いれば、 0 から $N-1$ の和になるので、

$$Y'(u) = \frac{C(u)}{\sqrt{N}} \sum_{m=0}^{N-1} x(m) \cos \left[\frac{\pi(2m+1)u}{2N} \right] \quad (19)$$

が成立する。式 (19) より、 $Y'(u)$ はすべて実数である。

式 (17) で、 $y(m)$ も $Y'(u)$ も実数であるため、複素数の指数関数項の \sin 関数の項は消えるはずであるから、

$$y(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{2N-1} \frac{Y'(u)}{C(u)} \cos \left[\frac{\pi(2m+1)u}{2N} \right] \quad (20)$$

が成立する。式 (19) より、 $Y'(2N-u) = Y'(u)$ と $Y'(N) = 0$ が成立するから、式 (20) の和を、 $m=0, 1$ から $N-1, N, N+1$ から $2N-1$ の四つの部分に分け、まとめれば、

$$x(m) = y(m) = \frac{1}{\sqrt{N}} \sum_{u=0}^{N-1} C(u) Y'(u) \cos \left[\frac{\pi(2m+1)u}{2N} \right] \quad (21)$$

が成立する。式 (19) と (21) が、DCT とその逆変換を表している。もとの信号を対称におく方法が複数通りあるため、DCT も複数通り存在するが、画像符号化では一般に上に示した DCT が使われる。

3.3 2次元 DCT

今までは、値が一列に並んでいる 1次元信号の変換を論じてきたが、画像は値が縦横に並んでいる 2次元信号である。画像符号化では、一般に画像の縦と横に対して可分な変換を用いる。すなわち、すべての行に対して 1次元の変

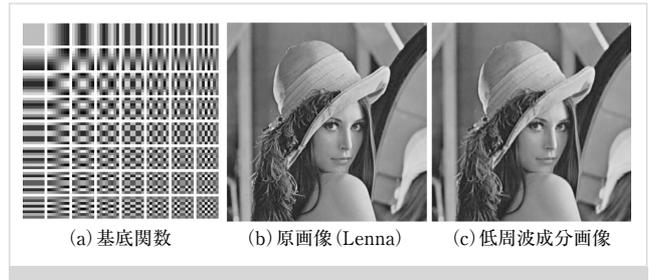


図3 2次元 DCT

換を行った後、その結果のすべての列に対して 1次元の変換を行って得られたものを、2次元の変換の結果とするものである。DCT の場合に、この処理を一つの式で表す。2次元信号 $x(m, n) (m, n=0, 1, \dots, N-1)$ から 2次元 DCT (2D DCT-II) によって得られる結果を $X(u, v) (u, v=0, 1, \dots, N-1)$ とすれば、

$$\begin{aligned} X(u, v) = \frac{C(u)C(v)}{N} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m, n) \\ \cos \left[\frac{\pi(2m+1)u}{2N} \right] \cos \left[\frac{\pi(2n+1)v}{2N} \right] \end{aligned} \quad (22)$$

となり、2次元逆 DCT (2D IDCT-II) は、

$$\begin{aligned} x(m, n) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v) X(u, v) \\ \cos \left[\frac{\pi(2m+1)u}{2N} \right] \cos \left[\frac{\pi(2n+1)v}{2N} \right] \end{aligned} \quad (23)$$

で表される。

式 (22) を見ると、 N^2 個の変換の結果 $X(u, v)$ は、入力信号と \cos 関数の積で構成される N^2 種類の基底関数との内積を計算して得られる形になっている。図 3 (a) は、 $N=8$ の場合の、この 64 個の基底関数を図示したものである。

DCT の画像符号化における有効性を見るために、図 3 (b) の画像を、 8×8 画素のブロックに分割し、それぞれのブロックを DCT で変換し、低周波成分を表す 8 個の変換の結果 $X(u, v) ((u, v) = (0, 0), (0, 1), (1, 0), (2, 0), (1, 1), (0, 2), (3, 0), (2, 1))$ はそのままにし、残りの 56 個の変換係数を 0 した後に逆変換して得られた画像が、図 3 (c) である。この図より、少数の変換係数で画像を十分な精度で表していることがわかる。

4. 離散ウェーブレット変換 (DWT)

DCT ではブロックごとに線形変換を行っており、データ量を小さくするために量子化ステップを大きくすると、復号画像のブロック境界に不連続な線が見える。その問題を解決できるブロックによらない変換が、JPEG2000 などで行われている DWT である。

DWT では、入力信号 $x(m)$ に対して、ローパスフィルタ ($h(n)$) とハイパスフィルタ ($g(n)$) を施し、低周波成分と高周波成分の二つの信号に分解する。そして、それぞれの成分から値の一つおきに取出すというダウンサンプリングを

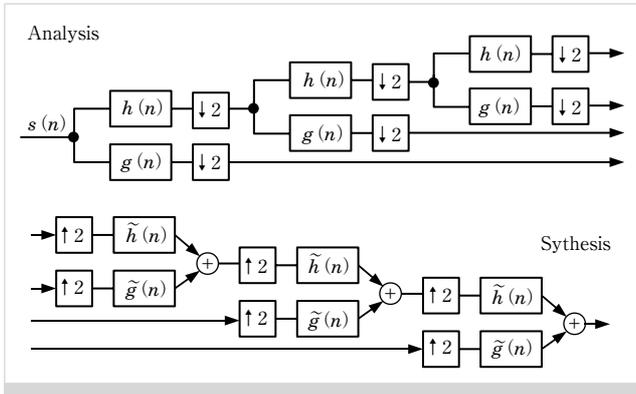


図4 離散ウェーブレット変換の階層化



図5 離散ウェーブレット変換

行う。ダウンサンプリングすることによって、二つの成分のデータ数と原信号のデータ数が同じになる。

逆DWTによって、二つの成分から元の信号に戻すときには、まず、各成分の値の間に0を挿入することによって2倍の長さとするアップサンプリングを行う。そして、一般には分解時とは別のローパスフィルタ $\tilde{h}(n)$ とハイパスフィルタ $\tilde{g}(n)$ を施し、両者を加算すればよい。

そしてDWTでは、この1回の分解によって得られた高周波成分はそのまま利用するが、低周波成分はさらにその低周波成分と高周波成分に分解していく(図4)。画像符号化では、この分解を5から6回程度行うことが多い。

2次元信号の場合、DCTと同様に横と縦に対してそれぞれ変換する。すなわち、すべての行に対して分解した結果のすべての列に対して分解を行う。

図5(a)はDWTして得られた結果であり、画像の左上が縦横とも低周波成分、左下が横が低周波成分で縦が高周波成分、右上が横が高周波成分で縦が低周波成分、右下が縦横とも高周波成分を表している。左上は両方とも低周波成分であるため、さらに分解している。この図は3回の分解を行った結果である。DWTを使った画像符号化では、基本的には図5(a)の各成分の値を量子化して、ビット列に変換する。

DWTによって、データ量が削減できる理由を説明する。DCTと同様に周波数で分割するため、低周波成分が大きく、高周波成分は小さくなる。このことは、図5(a)において、高周波成分は暗いことからわかる。したがって、少数の成分に分散を集中させることができるため、データ量を小さくすることができる。図5(b)は、横に関しては1段、縦に関しては2段分解した時の低周波成分はそのままにし、他の成分の値は0にした後に逆変換して得られた画像である。DCTの場合の図3(c)に対応して、全体の1/8の値だけを残している。DWTの低周波成分でも、原画像を十分な精度で表していることがわかる。

さらに、DWTでは階層表現により、DWTの変換結果に残る空間的冗長性を利用して、データ量を削減することも可能である。DCTを用いる符号化では、 $(u, v) = (0, 0)$ の成分に関してだけ、隣接ブロックとの差分を取ることで、データ量を削減しているが、他の成分は完全に別々

に値を符号化してしまっている。一方、図5(a)を見ると、DWTの高周波成分の値が小さい(暗い)領域はかたまっていること、縮尺が異なるが、高周波成分は分解の段数が異なる高周波成分と、その領域が対応していることがわかる。すなわち、分解の段数が多い高周波成分の量子化値が0ならば、分解の段数が少ない高周波成分の対応する位置の量子化値も0である可能性が高い。したがって、段数が多い高周波成分の量子化値からビット列にしていくときに、段数が少ない高周波成分の量子化値に対する符号を適応的に変えることによって、データ量を削減することができる。

DWTのためのフィルタ $h(n)$ 、 $g(n)$ 、 $\tilde{h}(n)$ 、 $\tilde{g}(n)$ は、多数提案されている。JPEG2000では、フィルタ関数の点数(タップ数)が、7または9であるDaubechiesの9-7フィルタが用いられる。また、画像符号化にDWTを適用するためには、信号が有限長であるための端点処理やデータ数が奇数である場合の処理も必要となるが、いずれの場合も変換前後でデータ数を変えずに、DWTと逆DWTが可能である。

5. む す び

画像符号化で用いられる線型変換に関して、それを用いる理由と変換の概要に関して解説した。さらに詳しく知りたい場合は、参考文献を参照すると良い。

(2012年10月29日受付)

〔文 献〕

- 1) H. Enomoto and K. Shibata: "Orthogonal Transform Coding System for Television Signals", IEEE Trans. on, Electromagnetic Compatibility, 13, 3, pp.11-17 (Aug. 1971)
- 2) K.R. Rao and P. Yip, (訳) 安田浩, 藤原洋: "画像符号化技術 [DCTとその国際標準]", オーム社 (July 1992)
- 3) I. Daubechies: "Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics (June 1992)
- 4) S. Mallat: "A Wavelet Tour of Signal Processing, Third Edition: the Sparse Way", Academic Press (Dec. 2008)



やました ゆきひこ
山下 幸彦 1985年、東京工業大学大学院理工学研究科情報工学専攻修士課程修了。同年、日本原子力研究所入所。(株)アイザックを経て、現在、東京工業大学大学院理工学研究科国際開発工学専攻にて、パターン認識、画像処理、集合知利用に関する研究に従事。准教授、博士(工学)。正会員。

変換係数の量子化

正会員 稲積 泰宏†

1. ま え が き

画像符号化はロスレス符号化とロッキー符号化に大別される。ロスレス符号化は原画像を符号化・復号した際に劣化がまったくない方式である。これに対して、ロッキー符号化は原画像を符号化・復号した際に劣化が発生する方式である。多くの方々が普段用いている画像符号化方式であるJPEGやMPEG-2, AVC/H.264などはすべてロッキー符号化である。

なぜ、高速なインターネット環境が整備された現在でも画像を正確に表現できないロッキー符号化が広く用いられているだろうか？ それは、次の二つの理由がある。

- (1) 原画像を厳密に符号化（ロスレス符号化）するには膨大な符号量を必要とする。
- (2) 人は画像を見る際に画像のすべての画素が完全に復号されているかどうかはわからない。

一般に、量子化とは「連続値を離散値で近似的に表現すること」である。ほとんどのロッキー符号化では、空間領域ではなく、周波数領域の「変換係数」に対して値の精度を落とすことで画像圧縮を実現している。これが、本稿で対象とする「変換係数の量子化」である。

本稿では、画像符号化における圧縮の要素技術である「変換係数の量子化」について解説する。特にJPEGやMPEG等、広く用いられている離散コサイン変換 (Discrete Cosine Transform: DCT) 係数の量子化を対象とする。また、変換係数の量子化により画質劣化を伴うため、ロッキー符号化器の設計や性能評価には、画質評価が必要不可欠である。そこで、画質評価の指標である、Peak Signal to Noise Ratio (PSNR)、主観評価、客観評価について概説する。

2. DCT 係数の量子化

「DCT 係数の量子化」とは、入力画像 ($N \times N$ のブロック単位, JPEG の場合 $N=8$) に対して、DCT を施して得られ

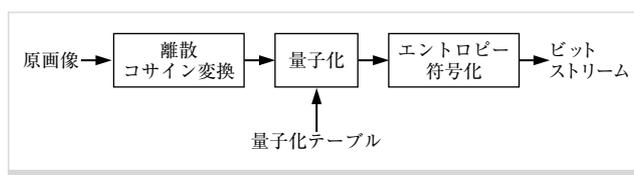


図1 DCT符号化器の構成

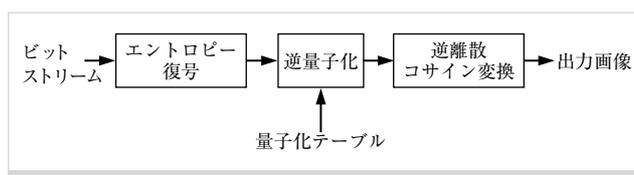


図2 DCT復号器の構成

る実数の変換係数を整数で近似する処理である。

DCT符号化器の構成を図1に示す。具体的な手順は以下の通りである。

- (1) 原画像をブロック単位でDCTを施す。
- (2) 得られた変換係数に対して「量子化」を行う。
- (3) 量子化された変換係数に対して、エン트로ピー符号化を行うことでビットストリームを生成する。

DCT復号器の構成を図2に示す。同図のように、DCT符号化器の逆の手順で出力画像を生成する。

- (1) ビットストリームをエン트로ピー復号する。
- (2) 復号された変換係数に対して「逆量子化」を行う。
- (3) 得られた変換係数に対して、逆離散コサイン変換 (逆DCT) を施し、出力画像を得る。

本稿では、DCT符号化器および復号器における「量子化」および「逆量子化」に着目する。図1および図2の「量子化」および「逆量子化」の入力に、「量子化テーブル」があるが、これは「量子化」および「逆量子化」で共通のものが用いられる。量子化テーブルとは、変換係数を量子化するために用いられるテーブルであり、 (8×8) ブロックDCT係数に対する量子化の場合は (8×8) の要素からなる行列であり、変換係数の各要素に対応している。すなわち、量子化テーブルの左上から右下に向かうほど高い周波数に対応している。

† 富山大学大学院理工学研究部 (工学)

"Basic Technologies of Image Coding (4): A Method for Quantization of Transform Coefficients" by Yasuhiro Inazumi (Graduate School of Science and Engineering for Research, University of Toyama, Toyama)

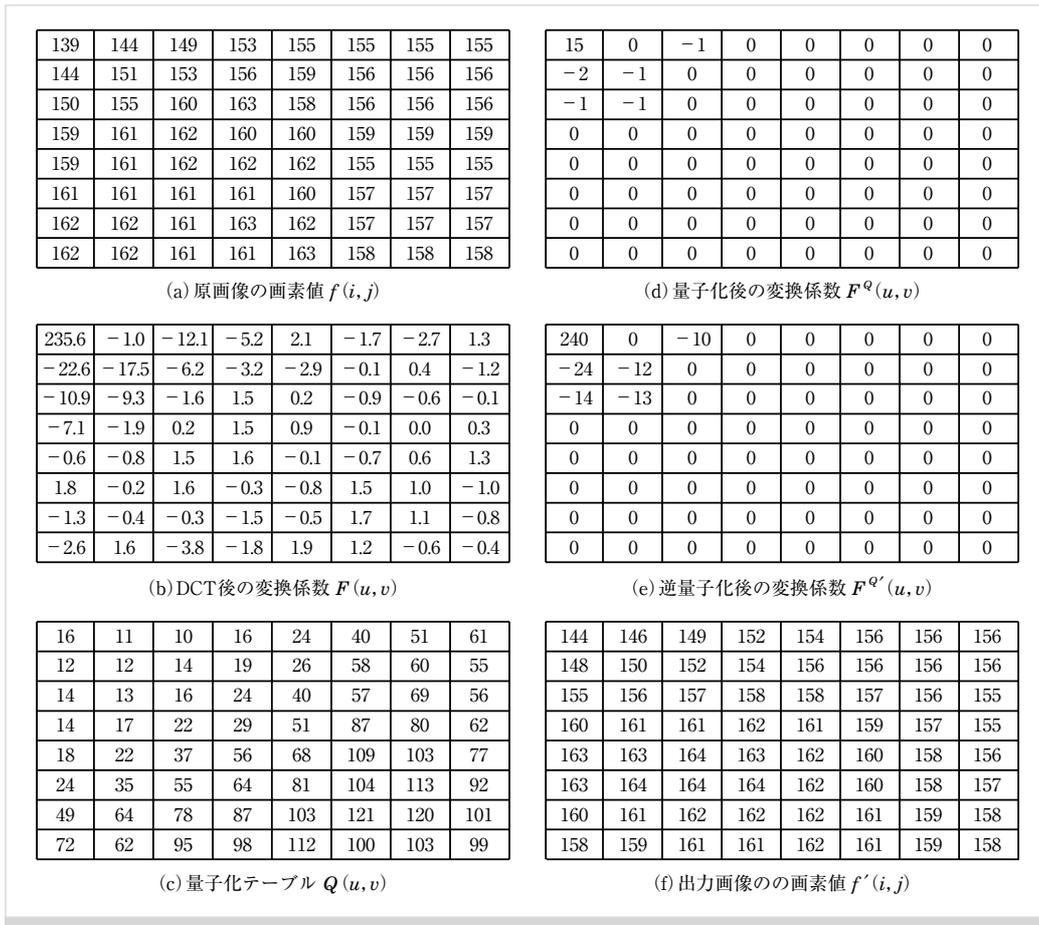


図3 量子化と逆量子化[文献1]から引用]

(8 × 8) ブロック DCT 係数の量子化の具体例を図3に示す¹⁾。図3 (a) は入力画像の画素値である。これに対して DCT を施した変換係数が図3 (b) である。図3 (c) は、JPEG で用いられている量子化テーブルである。図3 (d) は、図3 (b) の変換係数に対して図3 (c) の量子化テーブルに対応する要素の商を整数で丸めたものであり、これが DCT 係数の「量子化」である。これに対して、図3 (e) は、図3 (d) と量子化テーブル (図3 (c)) の積であり、これが DCT 係数の「逆量子化」である (なお、実際の画素値を直接 DCT するのではなく、「画素値-128」を入力としていることに注意されたい)。図3 (e) は逆量子化後の変換係数の値である。図3 (b) と図3 (e) を比較すると、変換係数の値が両者で類似しているが、量子化によって図3 (d) のように多くの値が0となるため、送信情報が削減できる。

さらに、図3 (e) を逆 DCT して得られる出力画像の画素値が図3 (f) である。原画像である図3 (a) と出力画像となる図3 (f) を比較すると、変換係数 (図3 (d)) では多くが丸められているにもかかわらず、画素値が類似している。これを画像として表現したものが図4 (a) および図4 (b) である。図4 (a) が原画像であり、図4 (b) が出力画像である。図4 (a) と図4 (b) を比較すると、両者の違いは一見わからないことが視覚的に確認できる。

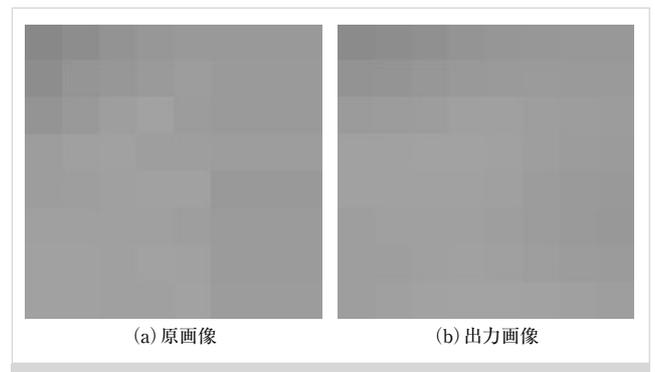


図4 図3 (a) と (f) を画像で比較した場合 (8 × 8 ピクセルを拡大)

一般的な画像処理アプリケーションを用いて JPEG で符号化する場合、どの程度の品質で符号化するかパラメータ (例えば、0 から 100 などアプリケーションに依存する) を指定することができる。このパラメータは量子化テーブルの倍率に対応しており、粗く量子化すると、前述の変換係数の0の数が増えるため高い圧縮率となるが、値の精度がより落ちるため画質がより劣化してしまう。逆に、細かく量子化を行うと値の精度が高まるため画質劣化が軽減されるが、変換係数の0の数が減るため低い圧縮率となってしまふ。したがって、画質と圧縮率はトレードオフの関係に



(a) 原画像BRIDGE



(c) 符号化画像BRIDGE



(b) 原画像BOAT



(d) 符号化画像BOAT

図5 コンテンツ間の主観品質の違い(PSNR=30dB)

あると言える。

ところで、なぜ量子化テーブルが図3(c)のように高周波成分を低周波成分に比べて優先的に削減するように設計されているのであろうか？これは、次の二つの傾向があるためである。

- (1) 一般に、自然画像を8×8ピクセル単位でDCTすると、その変換係数は直流成分および交流成分に分布するが、実際の電力(値)は直流成分および低周波成分に集中する傾向がある。
 - (2) 人の視覚特性として、高周波成分になるほど鈍感になり、低周波成分になるほど敏感になる傾向がある。
- 以上が、量子化によって画像圧縮される仕組みである。しかしながら、量子化により変換係数を近似しているため、図3(b)と同図(e)の関係のように、類似した値にはなるが、正確な値を保持していない。これが、画質劣化に影響するため、画質評価が必要となる。

3. 画 質 評 価

3.1 Peak Signal to Noise Ratio (PSNR)

符号化器の設計や性能評価に古くから使われている指標としてPSNRがある。256階調、画像サイズ $X \times Y$ の原画

像 O と対象画像 T のPSNRは、

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad [\text{dB}]$$

で定義される。式の255は画像のピーク信号である。ここで、

$$MSE = \frac{1}{XY} \sum_{j=0}^{Y-1} \sum_{i=0}^{X-1} [O(i,j) - T(i,j)]^2$$

である。

PSNRは値が高いほどノイズが少ないという尺度である。PSNRの原画像と主観の対応関係は、40dBの場合は見分けが不可能であり、30dBの場合は一見すると区別がつかず、20dBの場合は非常に劣化している傾向にある。しかしながら、上記はあくまでも目安であり、実際にはコンテンツに大きく依存するため、必ずしも数値と主観が一致するとは限らない。

PSNRは、同一の符号化器かつ同一のコンテンツでの比較では有効であるが、異なるコンテンツ間や異なる符号化器間の比較では役に立たない。これは、同一のノイズ量でも、符号化アルゴリズムやコンテンツの複雑さ等によって、人にとって見え方が大きく異なってしまうからである。その画像例を図5に示す。図5(a)のBRIDGEは画像全体が複雑なのに対して、図5(b)のBOATは空のように平坦な



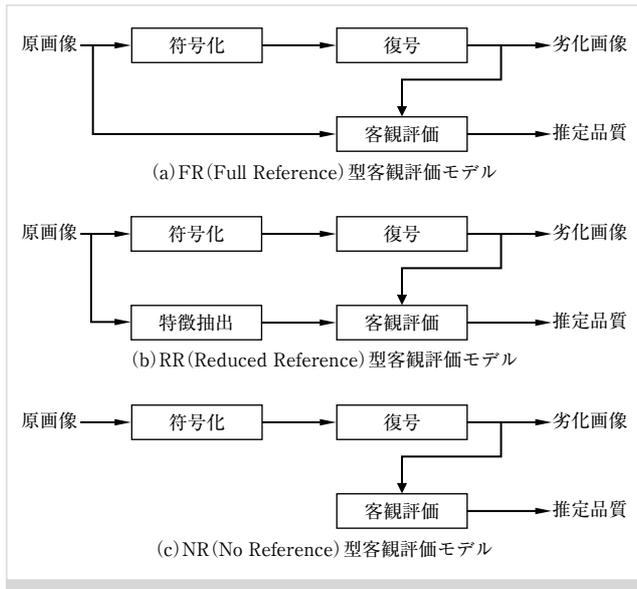


図6 客観評価モデルの種類

部分が多い。これらをそれぞれPSNRが30dBとなるようにJPEG符号化した画像が図5(c)および図5(d)である。図5(c)および図5(d)の両者とも、原画像と比べると劣化しているが、その程度は明らかに図5(d)のBOATの方が目立つ。しかしながら、実際のノイズの量はPSNRを揃えているため同一である。以上より、異なるコンテンツ間では、PSNRが同一でも、主観品質が近いという保証はまったくないと言える。

3.2 主観評価

画質の最終的な判断は人であるため、人が実際に画像を見て評点をつける主観評価が、人にとって有効な画質評価とされている。広く用いられる主観評価として、次の三つがある。

- (1) DSCQS (Double Stimulus Continuous Quality Scale) 法²⁾
- (2) DSIS (Double Stimulus Impairment Scale) 法²⁾
- (3) ACR (Absolute Category Rating) 法³⁾

DSCQS法は、原画像および対象画像を交互に2回提示し、0から100の連続値で評価する手法である。なお、原画像と対象画像の提示順はランダムである。

これに対して、DSIS法は、原画像の次に対象画像が提示され、それらの劣化の程度を5段階で評価する手法である。

一方、ACR法は、対象画像のみを5段階で評価する手法であり、評価に要する時間がDSCQS法およびDSIS法に比べ大幅に削減できるという利点がある。

3.3 客観評価

主観評価は人による評価であるため、信頼性が高いと考えられるが、多大な労力と時間を要するため、現実的ではない。そこで、主観評価で得られる評価値を推定する各種の客観評価が提案されており、近年ではVQEG (Video Quality Experts Group) で標準化作業が進められている⁴⁾⁵⁾。客観評

価モデルは図6(a)～図6(c)に示す以下の三種類に大別される。

- (1) FR (Full Reference) 型客観評価モデル
- (2) RR (Reduced Reference) 型客観評価モデル
- (3) NR (No Reference) 型客観評価モデル

図6の各図の客観評価の入力に着目すると明らかのように、FR型は原画像および劣化画像が必要であり、RR型は原画像の画像特徴量が必要ではあるが原画像自体は不要であり、NR型は原画像の情報をまったく必要としない。

実際に提案されている手法としてFR型⁶⁾、RR型⁷⁾、NR型⁸⁾などがあり、それぞれ図6の客観評価モデルに対応している。

FR型、RR型、NR型の順に必要な情報が減少するが、それに伴い主観評価値の推定精度が低下する傾向にある。一方で、原画像の情報を必要としない、もしくは特徴量のみしか必要としなければ、さまざまな用途で客観評価が利用可能である。そのため、RR型およびNR型での推定精度向上が期待される。

4. む す び

本稿では、ロッキー符号化のための主要技術となる変換係数の量子化についてDCTを対象として解説した。また、ロッキー符号化のための評価指標であるPSNR、主観評価、客観評価についても概説した。なお、詳細については参考文献を参照されたい。

(2012年10月9日受付)

[文 献]

- 1) G.K. Wallace: "The JPEG Still Picture Compression Standard", IEEE Trans. On Consumer Electronics, 38, 1 (Feb.1992)
- 2) ITU-R Recommendation BT.500-11: "Methodology for the subjective assessment of the quality of television pictures" (2002)
- 3) ITU-T Recommendation P.910: "Subjective video quality assessment methods for multimedia applications" (1999)
- 4) Video Quality Experts Group (VQEG), <http://www.its.bidrdoc.gov/vqeg-home.aspx>
- 5) 杉本修, 川田亮一: "VQEG (Video Quality Experts Group) の動向と関連技術", 信学Fundamentals Review, 1, 3, pp.27-36 (Jan.2008)
- 6) Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli: "Image Quality Assessment: from Error Visibility to Structural Similarity", IEEE Transactions on Image Processing, 13, 4, pp.600-612 (Apr. 2004)
- 7) J.A. Redi, P. Gastaldo, I. Heynderickx and R. Zunino: "Color Distribution Information for the Reduced-Reference Assessment of Perceived Image Quality", IEEE Transactions on Circuits and Systems for Video Technology, 20, 12, pp.1757-1769 (Dec.2010)
- 8) M.A. Saad, A.C. Bovik and C. Charrier: "Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain", IEEE Transactions on Image Processing, 21, 8, pp.3339-3352 (Aug.2012)



いとう 泰宏 1998年、富山大学工学部卒業。2000年、同大学院理工学研究科博士前期課程修了。2003年、東京工業大学大学院理工学研究科博士課程修了。同年、神奈川大学工学部助手。2007年より、富山大学大学院理工学研究部講師。画像処理、画像符号化の研究に従事。博士(工学)。正会員。

フレーム内予測

正会員 松田 一朗†

1. ま え が き

一般にデジタル画像信号は、水平・垂直方向に等間隔に配置した標本点の値(画素値)からなる2次元(動画の場合は時間方向も含めた3次元)の配列とみなすことができる。しかし、ランダムな数値が並んだ配列とは異なり、われわれが通常扱う画像には「距離が近い画素値同士は互いに似通っている」といった性質があり、その配列には明らかに冗長(無駄)な情報が含まれている。元の信号からこのような冗長性を取り除き、より少ないビット数で表現可能な情報に変換することは、効率的な画像符号化を実現するために不可欠な操作である。予測は直交変換と同様、画像信号の冗長性を削減する手段の一つであるが、比較的シンプルな処理であるため、画素値そのものだけでなく、直交変換を適用後の変換係数や、動き補償に用いる動ベクトルなど、画像符号化に関連するさまざまな情報の符号化にも広く利用されている。

フレーム内予測(画面内予測)は、動画画像符号化において、他のフレームを参照しないフレーム内符号化(イントラフレーム符号化)に用いられる予測方式の呼称であるが、原理的に静止画像符号化と共通の技術であり、画素値の空間方向の相関を利用していることから空間予測とも呼ばれる。本稿では、静止画像符号化方式に採用されているものも含め、各種の空間予測手法について解説する。

2. 予測符号化の基礎

デジタル信号の予測符号化は、古くから音声信号を対象として研究が行われ、比較的早い段階で画像符号化への応用が始まった¹⁾²⁾。これは、当時の撮像管などの画像取得装置が図1のように走査線に沿ったアナログ信号を出力していたため、これを標準化した画素値は、音声信号と同様に1次元の信号系列として扱えたためである。

さて、図1において画素 x_0, x_1, x_2, \dots を順に走査し、 x_{n-1}

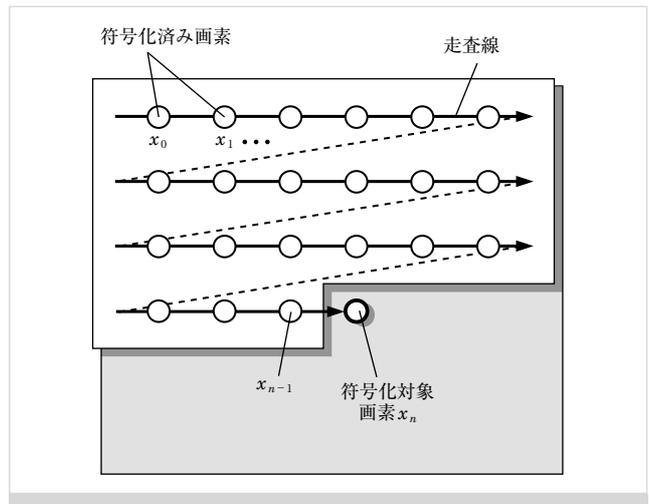


図1 画像信号の1次元化

まで符号化が終了したものとする。次に、符号化する画素値 x_n に対する最も単純な予測方式は、直前に符号化した値 x_{n-1} をそのまま予測値とする前値予測である。では、元の画像信号 x_n を直接符号化するよりも、予測値との差分である予測誤差 $e_n = x_n - x_{n-1}$ を符号化した方が効率的と言えるのはどのような場合であろうか? 厳密には符号化対象信号への符号割当て、すなわち、エントロピー符号化の方法を含めて議論する必要があるが、一般には符号化対象信号の平均電力(分散)が小さい方が符号化レートを小さくできることが知られている³⁾。言うまでもなく、隣接画素同士が互によく似ているならば、前値予測の予測誤差はゼロに近い値となり、その分散も充分小さくなることが期待できる。ここで、画像信号の平均値、分散および自己共分散(距離が離れた画素同士の共分散)が画素の位置 n に依存しないと仮定する(このような性質を弱定常性という)。

$$E[x_n] = \bar{x} \tag{1}$$

$$E[(x_n - \bar{x})^2] = C_{xx}(0) = \sigma_x^2 \tag{2}$$

$$E[(x_n - \bar{x})(x_{n-\tau} - \bar{x})] = C_{xx}(\tau) \tag{3}$$

隣り合った画素同士の統計的な類似度は、自己相関係数 $\rho = C_{xx}(1)/C_{xx}(0)$ で表せ、その値が1に近いほど互いに似ていることを意味する。予測誤差信号の平均電力は

† 東京理科大学 理工学部 電気電子情報工学科
"Basic Technologies of Image Coding (5): Intra Frame Prediction" by Ichiro Matsuda (Department of Electrical Engineering, Faculty of Science and Technology, Tokyo University of Science, Noda)



$$\begin{aligned} \sigma_e^2 &= E[e_n^2] = E[(x_n - x_{n-1})^2] \\ &= E\{[(x_n - \bar{x}) - (x_{n-1} - \bar{x})]^2\} \\ &= 2C_{xx}(0) - 2C_{xx}(1) = 2(1 - \rho) \cdot \sigma_x^2 \end{aligned} \quad (4)$$

となり、 $\rho > 0.5$ であれば元の信号電力 σ_x^2 より小さくなる。一般的な画像信号の相関係数は0.75~0.99であることが知られており¹⁾、前値予測によって信号電力を大幅に削減できることがわかる。逆に、 $\rho < 0.5$ の場合は信号電力が増加し、符号化効率の低下を招く恐れがある。では、 x_{n-1} に適当な係数 a をかけたものを予測値とした場合はどうか？ 簡単のため、画像信号から予め直流分を差し引くなどして $\bar{x}=0$ とみなせるものとする、予測誤差信号の平均電力は次式で表せる。

$$\sigma_e^2 = E[(x_n - a \cdot x_{n-1})^2] = (1 - 2\rho a + a^2) \cdot \sigma_x^2 \quad (5)$$

この式を最小とする係数は $a=\rho$ であり、これを最適予測係数という。最適予測係数の下では $\sigma_e^2=(1-\rho^2)\cdot\sigma_x^2$ となり、相関係数のとり得る値の範囲($-1 \leq \rho \leq 1$)で常に $\sigma_e^2 \leq \sigma_x^2$ が成り立つ⁴⁾。このことは、適切な予測を行う($\rho=0$ のときは予測を行わない)ことにより、符号化効率の低下を回避可能であることを示している。

以上の議論は、各画素における予測誤差 e_n を誤りなく伝送することで、再生画像が元の画像と完全に一致する可逆(ロスレス)符号化を前提としていた。一般に高い圧縮率を実現するためには、再生画像に歪を許容する非可逆(ロッキー)符号化が用いられる。予測符号化においては、予測誤差 e_n を粗く量子化することで非可逆符号化を実現できるが、このとき復号側の再生値 x'_{n-1} は元の信号 x_{n-1} に量子化雑音を加わったものであることに注意する必要がある。例えば前値予測において、復号側で受け取る予測誤差 e_1, e_2, \dots, e_n に誤りがなければ、先頭画素値 x_0 (x_1 の予測値)より画素値 x_n を正しく復号できる。

$$\begin{aligned} x_n &= x_{n-1} + e_n = x_{n-2} + e_{n-1} + e_n = \dots \\ &= x_0 + \sum_{k=1}^n e_k \end{aligned} \quad (6)$$

しかし、非可逆符号化では予測誤差 e_n に量子化雑音 d_n が加わった $e'_n=e_n+d_n$ を受け取るため、同様な手順を実行すると再生値に量子化雑音の和が重畳されてしまう。

$$x'_n = x_0 + \sum_{k=1}^n e'_k = x_0 + \sum_{k=1}^n (e_k + d_k) = x_n + \sum_{k=1}^n d_k \quad (7)$$

これを防ぐには、符号化側においても復号側と同一の再生値 x'_{n-1} を用いて予測誤差 $e_n=x_n-x'_{n-1}$ を求めればよい。このため、非可逆符号化に対応した符号化器は、図2のように、復号処理の一部を実行する局所復号器と、そこで得られた再生値を蓄えるフレームメモリーを備えているのが一般的である。

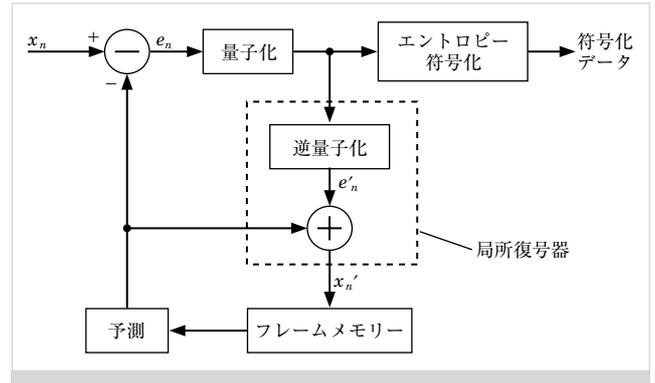


図2 符号化器の基本構成

3. 画素単位の予測

前値予測に基づく符号化手法は、初期の映像符号化方式(当時は帯域圧縮と呼ばれていた)に盛んに用いられていた⁵⁾。画像信号は、水平方向だけでなく垂直方向にも強い相関を示すことから、その後のフレームメモリーの低コスト化に伴って、現在では前値に加えて上方の走査線に存在する再生値も利用する2次元予測が一般的となっている。このように、画素単位で予測値との差分を符号化する方式を、DPCM (Differential Pulse Code Modulation) ともいう。

3.1 固定予測 (JPEG ロスレスモード)

静止画像符号化の国際標準方式であるJPEGは、離散コサイン変換(DCT)を用いた非可逆符号化方式として知られるが、DPCMに基づいたロスレス符号化モードも規格化されている⁶⁾。この符号化モードでは、図3に示すように、符号化対象画素(x)に対してすでに符号化が終了している左(x_a)、上(x_b)、左上(x_c)の再生値を用いて予測を行う。予測値 \tilde{x} の計算方法(予測式)は表1に示す8通りが定義されており、画像の性質に応じていずれかを選択することができる。ただし、符号化の途中でこれらの予測式を切替えることは許されていない。このように、画面全体で共通の予測式を用いる方式を固定予測という。表1のNo.1, 4, 7の予測式は、それぞれ前値予測、平面予測、行列予測と呼ばれ、固定予測方式として代表的なものである。

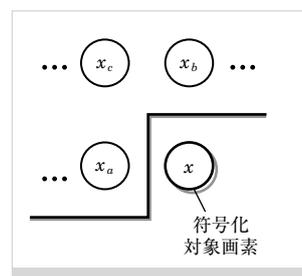


図3 画素配置

表1 JPEG ロスレスモードの予測式

No.	予測式
0	$\tilde{x} = 0$ (no prediction)
1	$\tilde{x} = x_a$
2	$\tilde{x} = x_b$
3	$\tilde{x} = x_c$
4	$\tilde{x} = x_a + x_b - x_c$
5	$\tilde{x} = x_a + (x_b - x_c)/2$
6	$\tilde{x} = x_b + (x_a - x_c)/2$
7	$\tilde{x} = (x_a + x_b)/2$



3.2 MED予測 (JPEG-LS)

上述のJPEGロスレスモードは、DCTベースの符号化アルゴリズムでは可逆符号化の実現が困難であることから、便宜的にJPEG規格に追加されたものであり、その符号化効率は必ずしも高くはなかった。そこで、可逆符号化に特化したJPEGの後続規格として開発されたのがJPEG-LSである⁷⁾。JPEG-LSは、Hewlett Packard Labsが提案したLOCO-I方式⁸⁾をベースとして開発されたものであり、基本的なアルゴリズムはJPEGロスレスモードと同様にDPCMを踏襲しているが、図3に示す画素値の大小関係に基づいて、次のように3種類の予測式を切替えているのが特徴である。

$$\tilde{x} = \begin{cases} \min(x_a, x_b) & (x_c \geq \max(x_a, x_b)) \\ \max(x_a, x_b) & (x_c \leq \min(x_a, x_b)) \\ x_a + x_b - x_c & (\text{otherwise}) \end{cases} \quad (8)$$

この予測方式は、3つの予測値 $\{x_a, x_b, x_a+x_b-x_c\}$ の中央値 (Median) を求めることと等価であり、MED予測またはMAP (Median Adaptive Prediction) と呼ばれている⁹⁾。このように、画素単位 (またはブロック単位) で予測式や予測係数を切替える手法を適応予測という。JPEG-LSでは、図2と同様に、予測誤差に対して一様量子化を施すことにより、再生誤差を指定された範囲 (例えば ± 1) に制限する準可逆符号化にも対応することが可能である。

4. ブロック単位の予測

DCTに代表される直交変換符号化は、一般に低ビットレートにおいてDPCMより優れた性能を示すといわれており¹⁰⁾、非可逆符号化方式の基本アルゴリズムとして広く用いられている。しかし、直交変換はブロック単位の独立した処理であるため、ブロック内の信号の冗長度削減には効果を発揮するものの、ブロック間の相関は原理的に利用できないという欠点がある。このため、近年は直交変換とブロック単位のフレーム内予測を併用する方式が注目を集めている。

4.1 DC/AC予測 (MPEG-4)

JPEGや動画像符号化標準であるMPEG-1/2のフレーム内符号化では、 8×8 画素のブロック単位で求めたDCT係数の直流成分 (DC係数) についてのみ、隣接ブロックとの相関を利用した固定予測 (直前に符号化されたブロックとの単純差分) が適用されていた。これに対し、H.263 version 2 (通称H.263+) と呼ばれる動画像符号化標準では、一部の交流成分 (AC係数) についても上または左のブロックを参照するフレーム内予測方式が導入され¹¹⁾、ほぼ同等の技術 (DC/AC予測) が後続規格のMPEG-4¹²⁾にも採用されている。

図4において、 $F(i, j)$ ($i, j=0, 1, 2, \dots, 7$) は当該ブロックのDCT係数が格納された 8×8 の配列を表している。MPEG-4のDC/AC予測では、まず当該ブロックの左、左上、上に位置する符号化済みブロックのDC係数 $F_A(0, 0)$ 、

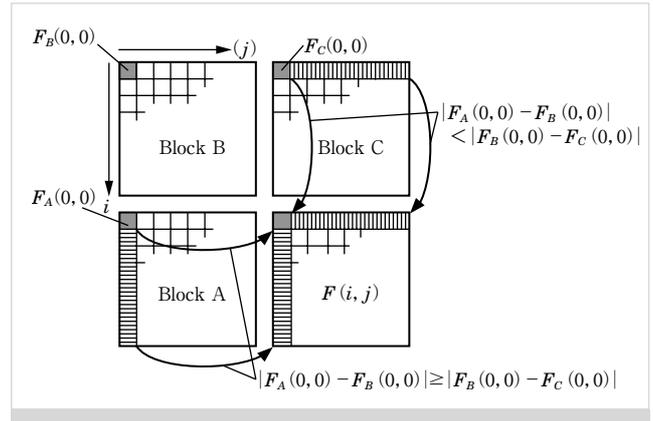


図4 DC/AC予測

$F_B(0, 0)$ 、 $F_C(0, 0)$ を参照し、水平・垂直いずれの方向で予測を行うのかを判定する。具体的にはDC係数の差分の絶対値 $|F_A(0, 0) - F_B(0, 0)|$ と $|F_B(0, 0) - F_C(0, 0)|$ の大小を比較し、前者が小さい場合は垂直方向、それ以外は水平方向の予測を適用するものとする。ただし、垂直方向の予測と判定された場合は、予測対象を当該ブロックの最上列に位置するDCT係数 $F(0, j)$ ($j=0, 1, 2, \dots, 7$)に限定し、上方向ブロック (Block C) の同一位置のDCT係数に、量子化精度の差を補正するスケーリングを施したものを予測値とする。これは、垂直方向の輝度変化が緩やかな領域では、上方向のブロックとの間で垂直方向の直流成分に相当するDCT係数同士の相関が高くなると期待できるためである。同様に、水平方向の予測と判定された場合は、当該ブロックの最左行に位置するDCT係数 $F(i, 0)$ ($i=0, 1, 2, \dots, 7$)を予測対象とし、左方向のブロック (Block A) の同一位置のDCT係数より予測値を算出する。しかし、DC係数 $F(0, 0)$ 以外では、実際にこれらの予測値との相関が高いとは限らないため、残りの7個のAC係数 ($F(0, 1), \dots, F(0, 7)$) または $F(1, 0), \dots, F(7, 0)$ については、予測を適用するか否かをマクロブロック (16×16 画素) 単位で指定できるようになっている。

4.2 方向性イントラ予測 (H.264/AVC)

MPEG-4のDC/AC予測は、DCT係数同士で予測を行うという制約のため、水平または垂直方向以外の相関を利用することが原理的に難しく、予測対象となるDCT係数の数も限られていた。このため、高効率な動画像符号化標準を目指して新たに策定されたH.264/AVC¹³⁾では、直交変換が適用される前のブロック内すべての画素をフレーム内予測 (イントラ予測) の対象とするように改められ、さまざまな方向のエッジやテクスチャにも対応できるよう、複数の予測モードが用意されることとなった。

H.264/AVCでは、直交変換の基本サイズが 4×4 に縮小されており、イントラ予測を施すブロックのサイズは、 4×4 画素、 8×8 画素、 16×16 画素の3通りが規定されて

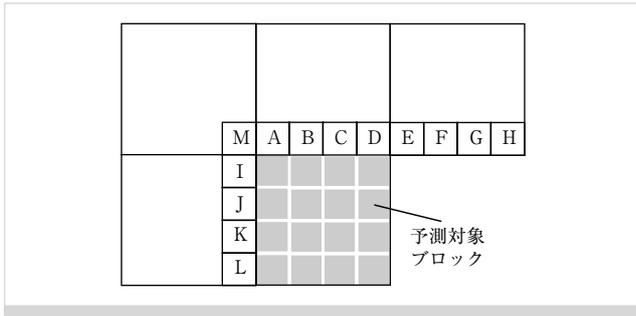


図5 H.264/AVC 4×4イントラ予測に使用する画素の配置

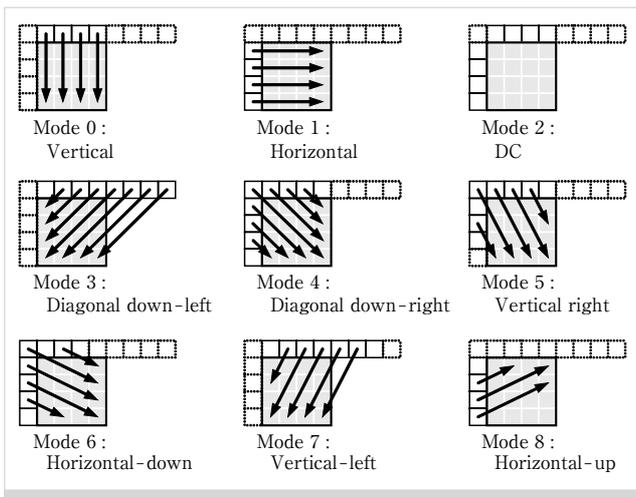


図6 予測モード(4×4イントラ予測)

いる。このうち最小サイズのブロックを対象とした4×4イントラ予測では、図5のように、周囲の符号化済みブロックの境界に位置するA～Mの計13画素を参照して予測値を算出する。予測値の算出方法を定めた予測モードは、図6のように9通りが用意され、ブロック毎に適切なモードを適応選択することができる。例えば、Mode 2のDC予測は、図5に示す画素A～D、I～Lの8画素の平均を求め、これを4×4画素それぞれの予測値として与えるものである。ブロック内が均一の値で予測されるため、画像の平坦部に適した予測モードといえる。それ以外の予測モードは、図6内の矢印の方向に参照画素の値を用いた予測を行うものであり、それぞれの方向にエッジなど方向性の強いテクスチャが存在する場合に有効となる。どの予測モードを選択するかについては、ブロックごとの付加情報として符号化する必要があるが、隣接ブロックと同一の予測モードを選択した場合は僅かな符号量で済むように、工夫がなされている。

5. む す び

フレーム内予測は比較的古い技術であり、特に非可逆符号化の分野では、直交変換やウェーブレット変換といった

新しい技術に主役の座を奪われたと思われる時期もあった。しかし、近年は直交変換の欠点を補う存在として再び注目を集めるようになっており、4.2節で紹介した方向性イントラ予測は、H.264/AVCの符号化効率向上に大きく貢献したといわれている。この流れを受け、次世代の動画画像符号化標準であるHEVCでは、最大33通りの角度に対応した方向性イントラ予測(Angular予測)が採用されるなど、フレーム内予測に関して大幅な改良が加えられる予定である¹⁴⁾。予測処理のシンプルな構成は他の符号化技術との相性も良く、今後も更なる発展が望める技術といえる。符号化効率の水準をさらに一段高めるような、画期的なフレーム内予測技術が登場することに期待したい。

(2012年11月29日受付)

〔文 献〕

- 1) E.R. Kretzmer: "Statistics of Television Signals", Bell Syst. Tech. J., 31, pp.751-763 (July 1952)
- 2) C.W. Harrison: "Experiments with Linear Prediction in Television", Bell Syst. Tech. J., 31, pp.764-783 (July 1952)
- 3) N.S. Jayant and P. Noll: "Digital Coding of Waveforms -Principles and Applications to Speech and Video", Prentice Hall (1984)
- 4) 伊東晋: "画像情報処理の基礎", 東京理科大学出版会 (Oct. 1986)
- 5) 石黒辰雄: "テレビジョン信号のデジタル伝送", テレビ誌, 27, 1, pp.2-9 (Jan. 1973)
- 6) ITU-T Rec. T.81 | ISO/IEC 10918-1: "Information Technology - Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines" (1994)
- 7) ITU-T Rec. T.87 | ISO/IEC 14495-1: "Information Technology - Lossless and Near-lossless Compression of Continuous-Tone Still Images: Baseline" (1999)
- 8) M.J. Weinberger, G. Seroussi and G. Sapiro: "The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS", IEEE Trans. on Image Processing, 9, 8, pp.1309-1324 (Aug. 2000)
- 9) S. A. Martucci: "Reversible Compression of HDTV Images Using Median Adaptive Prediction and Arithmetic Coding", Proc. of IEEE Int. Symp. on Circuits and Systems, pp.1310-1313 (May 1990)
- 10) A. Habibi: "Hybrid Coding of Pictorial Data", IEEE Trans. on Communications, 22, 5, pp.614-624 (May 1974)
- 11) G. Côté, B. Erol, M. Gallant and F. Kossentini: "H.263+: Video Coding at Low Bit Rates", IEEE Trans. on Circuits and Systems for Video Technology, 8, 7, pp.849-866 (Nov. 1998)
- 12) ISO/IEC 14496-2:2001 (E) : "Information Technology Coding of Audio-Visual Objects Part 2: Visual", 2nd edition (Dec. 2001)
- 13) ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC: "Advanced Video Coding for Generic Audiovisual Services" (2003)
- 14) J. Lainema and K. Ugur: "Angular Intra Prediction in High Efficiency Video Coding (HEVC)", Proc. of IEEE International Workshop on Multimedia Signal Processing (MMSP 2011), pp.1-5 (Oct. 2011)



まつだ いちろう
松田 一朗 1991年、東京理科大学理工学部電気工学科卒業。1996年、同大学大学院博士後期課程修了。同年より、東京理科大学理工学部電気工学科に勤務し、現在、同大学電気電子情報工学科准教授。画像符号化等の研究に従事。博士(工学)。正会員。

映像符号化におけるループ内フィルタ

正会員 松尾 翔平†

1. ま え が き

国際標準化機関 International Telecommunication Union Telecommunication Standardization Sector (ITU-T) および International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) によって標準化された映像符号化規格 (MPEG-1, H.262/MPEG-2, H.263, H.264/AVC など) は、映像情報の有する冗長性を効率良く削減するために、予測と変換というフレームワークを採用している。例えば、同一の画面内ですでに符号化済みの画素を復号して予測参照信号として予測信号を生成する画面内予測 (イントラ予測, Intra-frame Prediction) と、符号化済みのフレームを復号して予測参照信号として予測信号を生成する画面間予測 (インタ予測, Inter-frame Prediction) といった予測方式がある。また、原信号と上記予測信号との差分信号を、より圧縮しやすい情報に変換するために、離散コサイン変換 (Discrete Cosine Transform: DCT) といった変換が用いられる。

一般的な映像符号化の処理構造を図1に示す。予測信号として一度復号した信号をフレームメモリに格納して参照するループ構造 (図1内点線部) を有する。このループ内部で用いられるフィルタを、特にループ内フィルタ (図1内二重線部) と呼ぶ。このループ構造に入る前に適用されるフィルタをプレフィルタと呼び、ループ構造の後に適用されるフィルタをポストフィルタと呼ぶ。本稿では、ループ内フィルタを紹介する。

2. 次世代高能率映像符号化規格 HEVC¹⁾

映像符号化に用いられるループ内フィルタを説明する前に、規格化に向けて現在標準化が進められている次世代高能率映像符号化規格 HEVC を概説する。図1に示されるように、H.264/AVC と同様、HEVC も予測と変換からなる従来の映像符号化のフレームワークを利用する点は変わらない。H.264/AVC と比較して大きく異なっており、符号化効

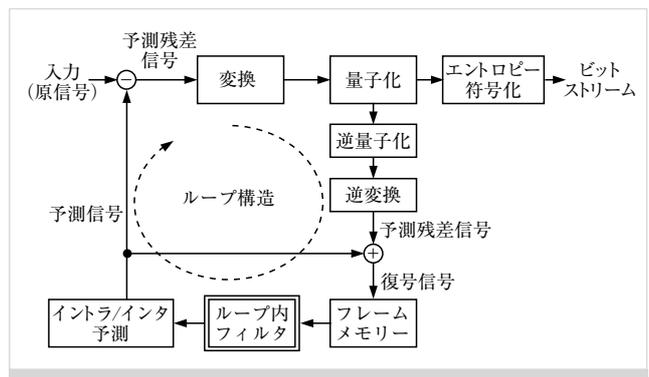


図1 映像符号化のループ構造とループ内フィルタの位置

率改善に貢献している点は、主に下記2点と考えられる。

(1) 柔軟な処理構造

H.264/AVCでは、符号化の処理の単位として 16×16 のマクロブロック (Macro Block: MB) が定義されていた。その中でイントラ予測であれば 16×16 , 8×8 , 4×4 の3種類のブロックサイズ、そしてインタ予測であれば 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4 の7種類のブロックサイズが予測に利用されていた。変換は 8×8 と 4×4 の整数変換が用いられる。

これに対し、HEVCでは、符号化/予測/変換を担うブロックをそれぞれ符号化/予測/変換ユニット (Coding/Prediction/Transform Unit: CU, PU, TU) と呼び、CU内部でPUとTUが独立に定義される。HEVCの参照ソフトウェアであるHEVC Test Model (HM) では、従来のMBに相当する最大のCUを Largest Coding Unit (LCU) と呼び、そのサイズを 64×64 としている。LCUは四分木ベースでCUに分割され、各CUは独立のPUとTUに分割される。PUとTUも四分木ベースで分割されていくが、PUに関しては Asymmetric Motion Partition (AMP) と呼ばれる $n \times n/4$ と $n \times 3n/4$ (32×8 と 32×24 や 16×4 と 16×12 などの長方形) のサイズを許容できる。HEVCの処理体系の一例を図2に示す。

以上のことから、 64×64 といった、 16×16 よりも大きな単位で処理が可能で、および予測と変換を独立に定義できる点がH.264/AVCとは異なり、特に高解像度での符号

† NTTメディアインテリジェンス研究所
"Basic Technologies of Image Coding (6): In-loop Filter of Video Coding"
by Shohei Matsuo (NTT Media Intelligence Laboratories, Yokosuka)

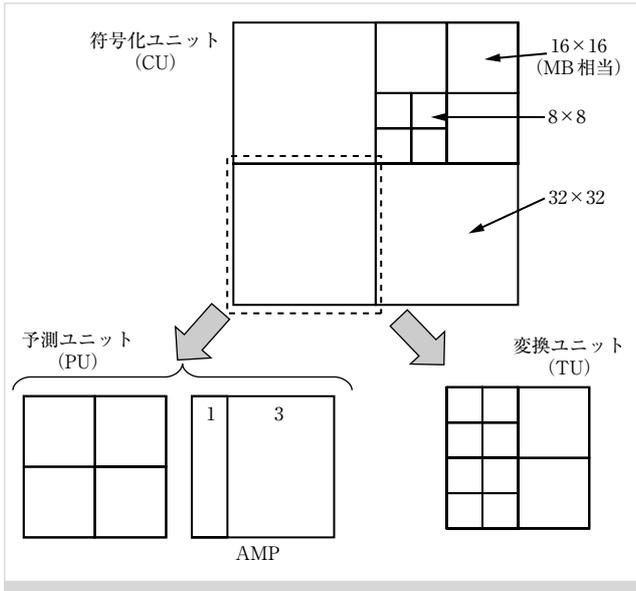


図2 HEVCの処理体系(CU, PU, TU)



図3 HEVCのループ内フィルタの構成 (Main Profile)

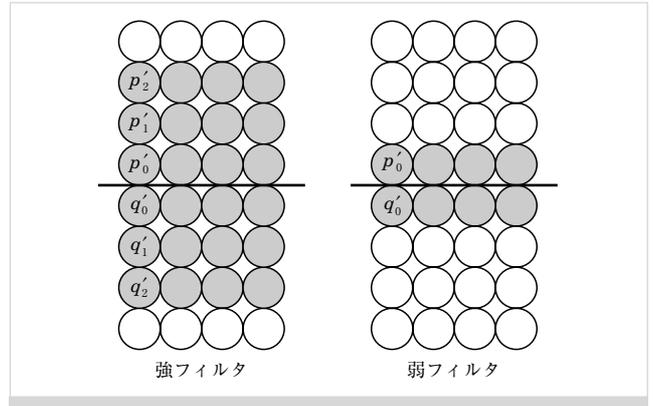


図4 デブロッキングフィルタの例

化効率向上に貢献していると考えられる。

(2) 新しい符号化ツールの導入

H.264/AVCには存在しない新しい符号化ツールがHEVCには導入されており、それらが符号化効率や主観画質改善に貢献している。具体的には、本稿で紹介する新しいループフィルタであるサンプルアダプティブオフセット、DCTベース補間フィルタなどである。詳細は次章にて解説する。

以上、2点の改善点に加えて、既存の符号化ツールも改良が施されて、全体として符号化効率がH.264/AVCと比べて大きく改善している。HEVCは、同一主観画質の下でH.264/AVCの1/2から1/3のビットレートに収まることや、客観性能の面では、約22% (イントラ符号化) から約35% (インタ符号化) ほどの符号化効率改善が報告されている²⁾。

3. HEVCに採用されたループ内フィルタ

本章では、HEVCに採用されたループ内フィルタを紹介する。2012年10月のJCT-VC上海会合時点で確定しているHEVCのMain Profileでは、図3に示される通り、3種類のフィルタが導入されており、それぞれ異なる機能を有している。HEVCの国際規格原案の定義では、補間フィルタはループ内フィルタではないが、本稿ではループ内で用いられるフィルタということで、本章にて紹介する。

3.1 デブロッキングフィルタ (DF)¹⁾

H.264/AVCにも採用されているループ内フィルタである。HEVCもH.264/AVCと同様にブロック単位で予測、変換、量子化を行うため、復号画像にはブロック状の歪みが発生する。前記歪みはブロック歪みと呼ばれ、特に低ビットレートにて顕著であり、主観画質の低下を招く。さらにインタ符号化の場合、その画質が劣化した画像を次の予測で用いるため、後続の符号化性能にも影響を及ぼすこと

になる。ブロック歪みを効率的に削除し、主観画質および客観性能を向上させる目的で、HEVCにもデブロッキングフィルタが導入されている。

HEVCのデブロッキングフィルタは、H.264/AVCと同じく、ブロックの境界線に対して、左右ないしは上下に隣接する画素を用いてフィルタ処理を施す。フィルタ処理は常にブロック境界に対して施すのではなく、符号化情報を用いて、境界強度 (Boundary Strength) を決定し、その状況に応じてフィルタのon/offや強度を切替えて適応的に処理する仕組みとなっている。

垂直方向のフィルタ処理の具体例を図4に示す。H.264/AVCと同様に強フィルタと弱フィルタが定義されており、ブロック境界 (直線) を中心に、強フィルタの場合は上下3画素ずつ、弱フィルタの場合は上下1画素ずつを変更する。修正前の画素を p_i, q_i ($i=0..3$)、修正後の画素を p'_i, q'_i ($i=0..3$) としたとき、強フィルタ処理は

$$p'_0 = (p_2 + 2p_1 + 2p_0 + 2q_0 + q_1 + 4) \gg 3 \tag{1}$$

$$p'_1 = (p_2 + p_1 + p_0 + q_0 + 2) \gg 2 \tag{2}$$

$$p'_2 = (2p_3 + 3p_2 + p_1 + q_0 + q_0 + 4) \gg 3 \tag{3}$$

となる。 q'_0, q'_1, q'_2 は上記式 (1), (2), (3) の係数を反転させて適用する。また、実際には $p_i - 2t_c \leq p'_i \leq p_i + 2t_c$ (t_c は量子化パラメータなどで決定される定数) の範囲に収まるようにクリッピング処理される。弱フィルタ処理は

$$\Delta = (9 * (q_0 - p_0) - 3 * (q_1 - p_1) + 8) \gg 4 \tag{4}$$

を算出し、 Δ の絶対値が $10t_c$ よりも小さい場合、 $-t_c \leq \Delta \leq t_c$ となるようにクリッピング処理した後、

$$p'_0 = p_0 + \Delta \tag{5}$$

$$q'_0 = q_0 - \Delta \tag{6}$$

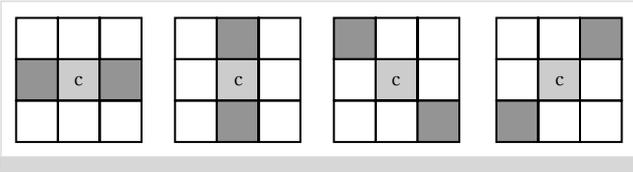


図5 エッジオフセット(EO)のパターン例

表1 EOのカテゴリ一例

カテゴリ	該当条件
1	$c < \text{二つの画素}$
2	" $c < \text{一方の画素}$ "かつ" $c = \text{もう一方の画素}$ "
3	" $c > \text{一方の画素}$ "かつ" $c = \text{もう一方の画素}$ "
4	$c > \text{二つの画素}$
0	上記1~4以外

とする。最終的には8ビットの場合は0から255に収まるようにクリッピング処理を適用した値が p'_0 と q'_0 になる。

弱フィルタの場合、量子化パラメータとその値から定義される変数に応じて、 p'_1 と q'_1 も処理される場合があるが、詳細は文献1)を確認していただきたい。

以上のフィルタリング処理により、ブロック境界上の歪みを低減して、主観画質の改善および後続フレームに対する予測効率の向上を実現する。

3.2 サンプルアダプティブオフセット (SAO) ¹⁾³⁾

デブロッキングフィルタが適用された後に用いられるループ内フィルタがSample Adaptive Offset (SAO)である。SAOは画素単位で適応的に画素値のオフセットを与える。エンコーダにて、原画像とDF後の復号画像の二乗誤差エネルギーを最小化するようにオフセットが計算され、そのオフセットをオーバーヘッドとして伝送する。2種類のオフセットモードが定義されており、それぞれBand Offset (BO) およびEdge Offset (EO) と呼ぶ。

BOは、CU内の各画素値を32のグループ(バンド0から31)に分割する。画素値が8ビットと仮定すると、画素値が0~7の時がバンド0、8~15の時がバンド1、…、248~255の時がバンド31、という形で分割する。その後、エンコーダは最も二乗誤差エネルギーが低減できる四つの連続するバンドを探し、各バンドに対するオフセットを計算して、バンドの位置情報と各オフセットをデコーダへ伝送する。

EOは、図5に示されるようなEO画素分類パターンについて、各画素値(c)に対してその隣接周辺画素(濃灰色)とのエッジ成分を計算して、その成分に応じて各画素を複数カテゴリへ分類する。表1に示すようなカテゴリごとにオフセットが計算される。選択された画素分類パターンと対応するオフセットがデコーダに伝送される。

SAOはエンコーダおよびデコーダの計算量の増加を抑えつつ、効果的に符号化歪みを削減して、符号化効率および主観画質に対する寄与が大きいため、HEVCのMain

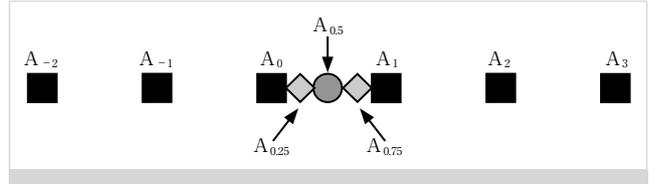


図6 H.264/AVCの補間フィルタ

Profileに採用されている。

3.3 DCT ベース補間フィルタ (DCT-IF) ¹⁾

デブロッキングフィルタ、サンプルアダプティブオフセットのフィルタを施した後、動き補償のために用いられるフィルタが本フィルタである。

H.264/AVCと同じように、HEVCでも小数画素精度の動き補償を採用している。小数画素位置の参照信号を生成するための補間フィルタは、予測効率を向上させる上で重要なポイントであり、HEVC標準化でも数多くの技術議論が展開された領域である。図6に示されるように、H.264/AVCの輝度信号補間では、1/2画素精度の位置($A_{0.5}$)を1次元6タップフィルタ $[1, -5, 20, 20, -5, 1]/32$ で補間した後、1/4画素精度の位置($A_{0.25}, A_{0.75}$)は平均値フィルタ $[1, 1]/2$ で補間を行っていた。色差に関しては、補間対象の位置の周辺4整数位置画素を用いて双線形フィルタによって補間される。

これに対し、HEVCでは新しくDCTベース補間フィルタ(DCT-based Interpolation Filter: DCT-IF)と呼ばれる補間フィルタが導入されている。DCT-IFは、補間フィルタのタップ長と補間点の位置が定まると、DCTの定義式からフィルタ係数が一意に定まる性質を有する。フィルタタップ長を $2M$ ($M > 0$)、整数精度位置にある画素を $p(i)$ (i は座標: $-M+1 \leq i \leq M$)、 $p(0)$ と $p(1)$ の間に存在する小数精度位置の画素を $p(\alpha)$ ($0 < \alpha < 1$)、フィルタ係数を C_k とすると、DCTの定義式により、

$$C_k = \frac{1}{M} \sum_{i=-M+1}^M p(i) \cos\left(\frac{(2i-1+2M)k\pi}{4M}\right) \quad (7)$$

$$p(\alpha) = \frac{C_0}{2} + \sum_{k=1}^{2M-1} C_k \cos\left(\frac{\pi(2\alpha-1+2M)k}{4M}\right) \quad (8)$$

が導かれる。DCT-IFが提案された当初は、輝度信号には12タップ ($M=6$)、色差信号には8タップ ($M=4$)として8ビット精度で丸められた係数が利用されていた。長い議論と検討の結果、位相シフトなども考慮しつつ、現在は輝度信号8タップ、色差信号4タップの6ビット精度の係数が用いられている。表2および表3が実際のフィルタ係数群になる。

4. HEVC 標準化に提案されたループ内フィルタ

本章では、HEVCのMain Profileには採用されていないが、HEVC標準化にて提案されており、符号化効率や主観



表2 輝度信号用8タップDCT-IF係数

小数位置	フィルタ係数
1/4	$[-1.4, -10.58, 17, -5.1, 0]/64$
1/2	$[-1.4, -11, 40, 40, -11, 4, -1]/64$
3/4	$[0, 1, -5, 17, 58, -10, 4, -1]/64$

表3 色差信号用4タップDCT-IF係数

小数位置	フィルタ係数
1/8	$[-2, 58, 10, -2]/64$
1/4	$[-4, 54, 16, -2]/64$
3/8	$[-6, 46, 28, -4]/64$
1/2	$[-4, 36, 36, -4]/64$
5/8	$[-4, 28, 46, -6]/64$
3/4	$[-2, 16, 54, -4]/64$
7/8	$[-2, 10, 58, -2]/64$

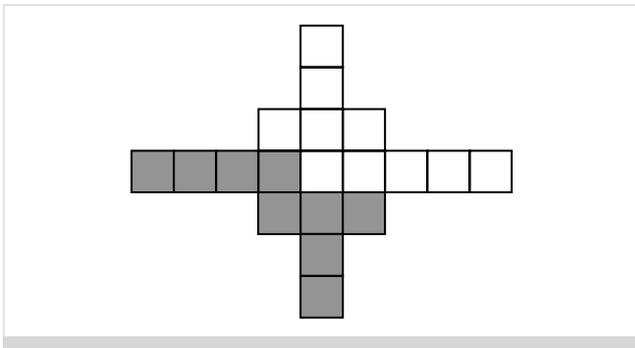


図7 ALFのフィルタ形状の例

画質の改善効果が大きいループ内フィルタを紹介する。

4.1 アダプティブループフィルタ (ALF)⁴⁾

SAOと同様に符号化歪みを除去する目的で提案されたループ内フィルタである。フレーム、LCUで構成される領域、4×4ブロックなどの単位で、エンコーダにて予測誤差エネルギーを最小化するようにウィナーフィルタの係数を設計し、デコーダにフィルタ係数を伝送するフィルタである。主観画質への寄与が小さく、かつ復号処理量の増加が大きいという観点からMain Profileには導入されていないが、主に視覚的に目立たない部分の符号化歪みを効果的に削減でき、符号化効率の改善率はループ内フィルタの中では最も高いので、今後の拡張プロファイルへの導入が期待されるフィルタである。HEVC標準化にて提案されていたALFのフィルタ形状の一例を図7に示す。図7の中心にある画素を修正するために、19画素をフィルタ入力対象とし、対称性を用いた構成で、実際に伝送するフィルタ係数の数は図7の白画素で示される10係数となる。

4.2 ノンローカルミーンズベースループフィルタ (NLM)⁵⁾

画像の歪みを効果的に除去するフィルタとしてNon-Local Meansフィルタが提案されており、当該フィルタの

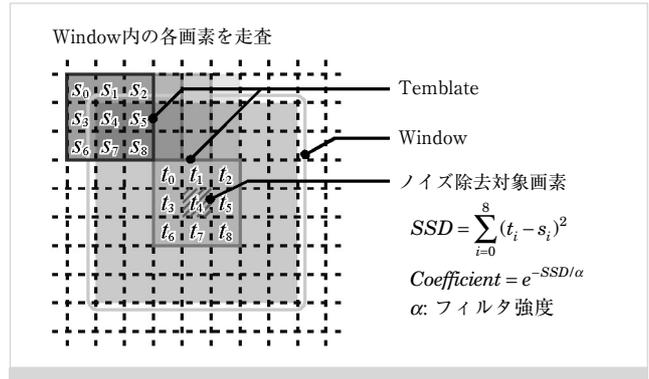


図8 NLMのフィルタ係数算出処理例

概念を映像符号化に応用したループ内フィルタが、本フィルタである。ALFが直接フィルタ係数を伝送するのに対し、NLMはフィルタ強度のパラメータのみ伝送し、デコーダでフィルタ係数を復号画素値から算出する点が異なる。

具体的なフィルタ形状とフィルタ処理対象範囲の事例を図8に示す。Windowで定義される範囲をTemplate形状で走査し、二乗誤差SSDに応じて対象画素の係数を決定する。デコーダでの演算量が大きく増加するので、HEVCのMain Profileには採用されていないが、符号化効率の改善、特に主観画質改善への寄与が大きいフィルタである。

5. む す び

本稿では、映像符号化に利用されているループ内フィルタの定義とその事例について紹介した。符号化ツールの中でも、ループ内フィルタは符号化効率改善や主観画質向上の観点で寄与率が高く、HEVCを含めた今後の映像符号化規格でも重要な位置付けであり、さらなる発展が期待される領域である。

(2012年12月21日受付)

〔文 献〕

- 1) B. Bross, et al.: "High Efficiency Video Coding (HEVC) text specification draft 9 (SoDIS)", JCTVC-K1003, 11th JCT-VC meeting (Oct.2012)
- 2) B. Li, et al.: "Comparison of Compression Performance of HEVC Draft 7 with AVC High Profile", JCTVC-J0236, 10th JCT-VC meeting (July 2012)
- 3) C.-M. Fu, et al.: "CE8 Subset3: Picture Quadtree Adaptive Offset", JCTVC-D122, 4th JCT-VC meeting (Jan.2011)
- 4) T. Yamakage, et al.: "CE8: Summary report of Core Experiment on non-deblocking loop filtering", JCTVC-H0038, 8th JCT-VC meeting (Jan.2012)
- 5) M. Matsumura, et al.: "CE8h: CU-based ALF with non-local means filter", JCTVC-G235, 7th JCT-VC meeting (Nov.2011)

まつお しょうへい
松尾 翔平 2006年、早稲田大学大学院国際情報通信研究科修士課程修了。同年、日本電信電話(株)入社。現在、NTTメディアインテリジェンス研究所にて映像符号化の研究開発および標準化活動に従事。2010年、本会鈴木記念奨励賞受賞。正会員。

フレーム間予測技術

児玉 明†

1. ま え が き

フレーム間予測技術は、動画像圧縮に欠かせない技術の一つである。本稿では、動画像の動き予測に関する技術として、動き補償フレーム間予測、動きベクトルの検出方法、予測精度などについて概説する。

2. フレーム間予測

はじめに、フレーム間予測方法として、動き補償フレーム間予測、動きベクトルの検出法について述べる。次に、ワーピング予測、オーバーラップ動き予測、グローバル動き予測、アフィン変換を利用した動き予測について紹介し、おわりに、ベクトル探索法、予測精度について説明する。

2.1 概要

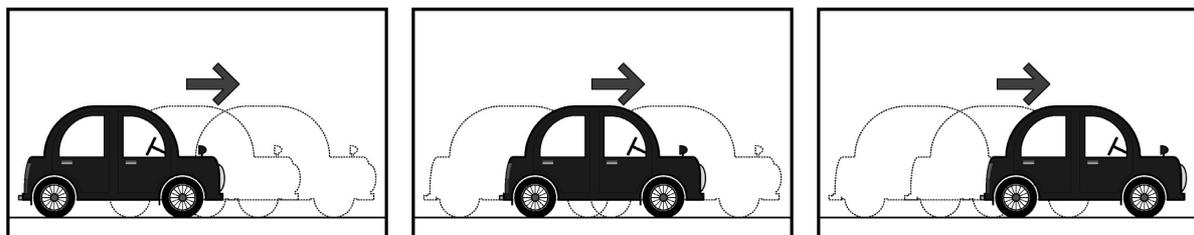
一般的にテレビで利用する動画像は1秒間に約30枚のフレームから構成され、隣接するフレーム間の相関が高い場合がある。動画像圧縮において、このような時間冗長性として、時間方向に連続するフレーム間の類似性を考える。ここで理解しやすくするために、固定カメラにおいて被写体を撮影している場合を考える。また、背景の変化はないとする。

参照フレーム(過去フレーム)と符号化対象フレーム(現フレーム)を比較する。被写体の動きがない場合は、フレ

ーム間で差分を取ると残差信号がゼロとなる。この処理をフレーム間差分という。一方、被写体の動きがある場合は、同様にフレーム間差分を行うと、現フレームでは過去フレームの被写体の位置と、現フレームの位置で差分データが発生し、その他はゼロとなる。このままでは、被写体形状領域2つ分の差分データが発生する。さらに差分情報量を減らすために、次の処理を考える。被写体の含まれる画像領域を抽出し、その被写体領域の動き量を算出する。参照フレーム(過去フレーム)における被写体から、動き量分位置を移動した被写体画像を生成する。この処理を動き補償フレーム間予測(単に、フレーム間予測)と呼び、この生成した被写体画像を動き補償画像(予測画像、予測フレーム)と呼ぶ。

フレーム間予測の方法として、一方向の予測を利用する片方向予測と両方向(双方向)予測がある。時間的に過去のフレームから現フレームに対する予測を順方向(前方)予測、未来フレームから現フレームに対する予測を逆方向(後方向)予測と呼ぶ。その両方を利用する場合を双方向(両方向)予測といい、片方向予測と比較して予測精度が向上する。フレーム間予測の例を図1に示す。例えば、(a)から(b)への予測が順方向予測、(c)から(b)への予測が逆方向予測、その両方を利用するのが、双方向予測である。

最終的に、現フレームと予測画像との間でフレーム間差



(a) 過去フレーム (参照フレームi)

(b) 現フレーム (符号化対象フレーム)

(c) 未来フレーム (参照フレームii)

図1 フレーム間予測の例

† 広島大学

"Basic Technologies of Image Coding (7): Inter-frame Prediction Technology" by Mei Kodama (Hiroshima University, Hiroshima)

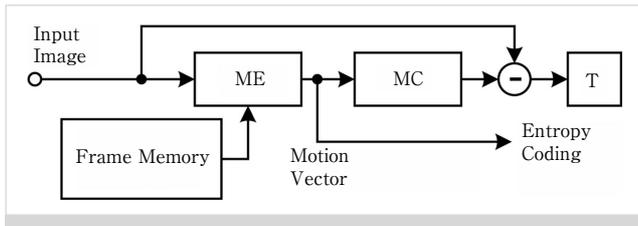


図2 動き予測処理

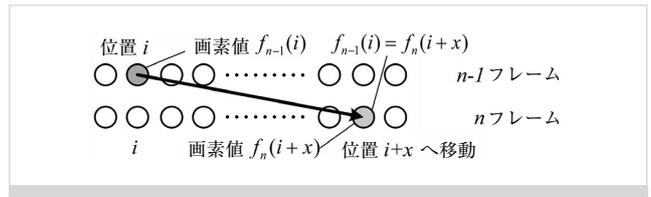


図3 勾配法の原理

分処理を行い、その予測誤差信号(残差信号)を符号化することで、現フレームを復元することができる。予測処理を利用することで、現フレームをイントラ符号化する場合よりも情報量を大幅に削減することができ、単に、フレーム間差分符号化する場合よりもさらに情報量を抑制できる。一般的に、矩形ブロック単位に抽出し、ブロック内で同一の動きベクトルを用いることで、画素あたりの動きベクトル情報の削減を図る。符号化効率に大きな影響を与える一つの要素として、動き補償の効率化が挙げられる。したがって、動き検出手法、動きベクトル(移動量)の単位、動きベクトルを用いた予測画像の生成方法などが重要である。

動き予測処理は、動き検出(ME: Motion Estimation)、動き補償画像の生成(MC: Motion Compensation)、動き補償画像と符号化対象画像との差分符号化の大きく3つの処理から構成される。例を図2に示す。ただし、図中のTは直交変換を示す。また、低遅延符号化時は、参照フレームの時間関係から順方向予測のみが利用される。一方、符号化処理・復号遅延を許容できる条件下では、未来フレームを先に符号化した後に、それにより過去のフレームを符号化する、逆方向、双方向予測を行うことで、予測効率の向上が期待できる。ただし、参照するフレームが多くなるのでメモリー量が増加する。

図2の処理は、単にMC、あるいは、ME+MCとして表現される場合がある。MEは符号化での処理であり、復号器ではMCと差分(予測誤差)情報との加算処理が行われる。したがって、本処理には過去フレーム、あるいは、未来フレーム、あるいはその両フレームを記憶するためのメモリーが必要である。

ここでは、被写体を中心とした動きを例としたが、他に、被写体などが動きに応じて見え隠れする部分を予測する背景予測にも利用できる。

2.2 動き検出方法

動き検出手法の基本として、はじめに勾配法とブロックマッチング法を示す¹⁾²⁾。

(1) 勾配法

画素単位に動きを検出する手法として、勾配法(Horn-Schunck method)³⁾がある。勾配法は、フレーム間の輝度勾配は微小区間で一定であると仮定し、フレーム内の輝度変化に着目する。次に、参照フレームと符号化対象フレ

ーム間で、輝度変化量が同じ部分を抽出し、動きを検出する方法である。

ここで、1次元系列で与えられる画像を考える。(n-1)フレームにおける画素位置*i*が、*n*フレームにおける画素位置(*i+x*)に移動したと仮定する(図3)。輝度勾配一定より、

$$\begin{aligned} f_n(i+1) - f_n(i) &= f_n(i+2) - f_n(i+1) = \dots \\ &= f_n(i+x) - f_n(i+x-1) = c \end{aligned} \quad (1)$$

となる。*c*は隣接画素間差分である。式(1)より $f_n(i+x) - f_n(i) = xc$ が成立する。また、(n-1)フレームの画素 $f_{n-1}(i)$ は*n*フレームの画素 $f_n(i+x)$ に対応する。したがって、隣接画素間の移動距離(1次元動きベクトル)*x*は

$$x = \frac{f_n(i+x) - f_n(i)}{c} = \frac{f_{n-1}(i) - f_n(i)}{f_n(i+1) - f_n(i)} \quad (2)$$

で表される。ここでは、1次元で示したが2次元へ拡張する場合、この水平方向の計算をさらに垂直方向へ適用することで算出できる。ただし、勾配法は、画素の輝度変化量を利用するので雑音に弱く、隣接画素値の差が小さい場合、利用が難しい。以上の、輝度変化量の連続性を仮定して算出した動きベクトルをオプティカルフロー(Optical Flow)と呼ぶ。

(2) ブロックマッチング法

一般的によく利用されるブロックマッチング法(Block Matching Method, 以下BMと略す)について概説する。本動き予測は、撮影した被写体の動きを剛体(変形のない被写体)の平行移動と仮定したモデルである。

まず符号化対象フレームを矩形ブロックに分割する。各矩形ブロックにおいて、同一サイズで参照フレームから類似したブロックを探索する。ここでは、参照フレームと同位置のブロックを基準としたある探索範囲内における全探索法を利用する。BMを図4に示す。

まず対象矩形ブロックに対して、符号化対象フレームの位置と対応する参照フレームの位置の矩形ブロック(テンプレート)を定める。テンプレートから、ある探索範囲($-X \leq x_0 \leq X, -Y \leq y_0 \leq Y$, 図中の点線領域)を全探索し、誤差が最小となるブロックを参照ブロックと定める。誤差評価関数として、差分絶対値和(SAD: Sum of Absolute Difference), 平均二乗誤差値(MSE: Mean Square Error), 平均絶対差

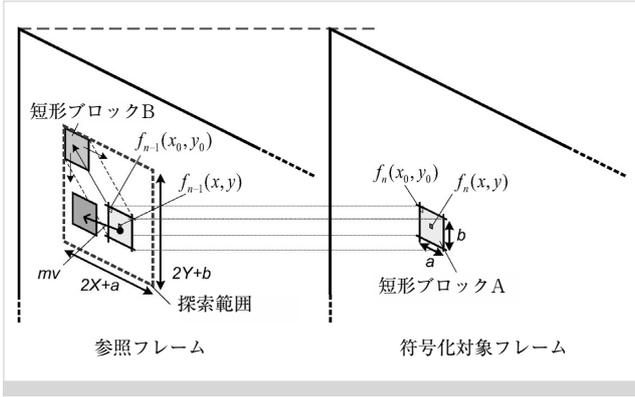


図4 ブロックマッチング法

(MAD: Mean Absolute Diffenece)などが利用される。式(3)にSADを利用した評価関数 J_{SAD} の例を示す。

$$J_{SAD} = \sum_{Block A} |C_i - R_i| \quad (3)$$

$(C_i \in Block A, R_i \in Block B)$

SAD最小 ($min SAD$) となる参照ブロックを用いて、最終的に、参照ブロックとテンプレート間で、動きベクトル ($\vec{mv} = (v_x, v_y)$) を算出する。ただし、誤差最小が移動位置の正確性を示すとは限らない場合があり、符号化効率の観点とは別に、真のベクトル検出に関する研究もある。

このモデルで近似できるシーンに対しては符号化効率が向上する。被写体を追従しカメラがパニングなどしている場合においても、モデルに近い動きについては、同様に符号化効率向上が期待される。

次に、動き量を算出する領域単位(矩形ブロックの大きさ)と領域形状について考える。予測ブロック形状を可変にする方法⁴⁾⁵⁾が検討されている。扱う領域単位を小領域にすることで、予測誤差の低減が期待される。しかし、領域数が増加すると、符号化する動きベクトルの情報量が増加する。予測誤差低減とベクトル情報量増加との間でトレードオフの関係がある。一般的に、 16×16 , 8×16 , 16×8 , 8×8 , 4×8 , 8×4 , 4×4 などの矩形ブロックが利用される。HEVCでは、 64×64 まで扱うことが検討されている。

(3) ワーピング予測, オーバラップ動き予測

予測効率の向上のための方法として、ワーピング予測とオーバラップ動き予測がある。

ワーピング予測は、フレームを三角形や四角形のパッチに分割し、予測単位となるパッチの頂点(格子点)を代表点とする。代表点の動きベクトルから、パッチ内部の画素の動きベクトルをベクトル内挿により補間する手法⁶⁾⁷⁾が検討されている。代表点の動きベクトルは、代表点を中心としたBMより算出する。ワーピング予測は、ベクトル内挿により、平行移動に加えて、物体の伸縮、回転にも対応することが可能である。しかし、異なる動きを持つ物体の境

界に跨るパッチに利用した場合、境界付近に歪みが発生する場合がある。

隣接ブロックの動きベクトルを用いた複数の参照画素の加重平均により予測画像を作成するオーバラップ動き補償がある^{8)~11)}。BMのブロック境界における動きの不連続性の課題に対して、ブロックを重ね合わせることで予測誤差を低減する方法である。

ここまで、剛体の平行移動を想定した動き予測を中心に話してきた。次に、その他の移動モデルに基づいた予測方法として、グローバル動き予測、アフィン変換による動き予測について概説する。

(4) グローバル動き予測

グローバル動き予測は、カメラの動きにより生じるパン、チルト、ズームなどの動きパラメータを予測するものである。カメラ操作により生じる、パン、チルト、ズームなどは画面全体の動きとして現れる。BMにおいてもある程度予測できるが、画面全体を一組の動きパラメータで記述できれば符号化効率が向上する。加えて、平行移動モデルを想定したBMとは異なり、ズームを含めて表現できるところが特徴である。グローバル動き検出(GME: Global Motion Estimation)/グローバル動き補償(GMC: Global Motion Compensation)の処理例として、上倉¹³⁾の方法について示す。

まず、GMEにより、例えば3種類のカメラの動き、パニング量 H 、チルト量 V 、ズーム率 Z を求める。これらの3つのパラメータと参照画像から、グローバル動き補償予測フレームを生成する。

各パラメータの推定手順を次に示す。フレームの中心座標を原点として、原点对称となる2つのブロックの平行移動ベクトルの組みをBMにより求める。次に、算出したベクトルから、グローバル動きパラメータ H, V, Z を求める。

$$Z = \frac{\{i_m(w_x - w'_x) + j_n(w_y - w'_y)\}}{\sqrt{2(i_m^2 + j_n^2)}} \quad (4)$$

$$H = (w_x + w'_x)/2, \quad V = (w_y + w'_y)/2$$

ここで、 (i_m, j_n) はブロック $B_{m,n}$ の中心座標で、原点に対応する座標で表し、また、対称な位置にあるブロックをブロック $B_{m,n}$ 、ブロック $B_{-m,-n}$ とすると、その動きベクトルをそれぞれ (w_x, w_y) 、 (w'_x, w'_y) としている。フレーム全体に対して、パラメータ Z, H, V のヒストグラムをとり、最大頻度となる値をグローバル動き補償パラメータとして算出する方法が提案されている¹³⁾。

画素 (i, j) におけるベクトル $(\mathbf{v}(i, j) = (v_x(i, j), v_y(i, j)))$ は、画面内の中心座標からの変換を考慮して、式(5)で表現される。ただし、 H_0, V_0 はそれぞれ画面の横サイズ/2、縦サイズ/2である。

$$\begin{pmatrix} v_x(i, j) \\ v_y(i, j) \end{pmatrix} = \begin{pmatrix} Z & 0 \\ 0 & Z \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} + \begin{pmatrix} H - H_0 \\ V - V_0 \end{pmatrix} \quad (5)$$



画素 (i, j) の予測値は、参照画像の座標 $(i+v_x(i, j), j+v_y(i, j))$ として求められる。各画素のグローバル動きベクトルは、1/2画素精度で利用される。別の方法として、グローバル動き補償後の予測値を、4画素の内挿補間を用いて算出する方法もある¹⁴⁾。また、GME/GMCでは、フレーム全体で生じる輝度変化や、ノイズ、照明変化などに対して課題を有する。グローバルな動きが正確に検出されなかった場合、画面全体の予測効率が低下する可能性があるため、局所的な利用方法が提案されている^{15) 16)}。

(5) アフィン変換による動き予測

近似的なアフィン変換を利用した動き予測^{17)~19)}は、物体の平行移動、回転、拡大・縮小を予測するものである。アフィン変換では、現フレーム内での画素位置 (x, y) を入力とし、式(6)にしたがい参照フレーム内での対応する画素位置 (x', y') を予測する。 (x', y') が整数座標でない場合は、周囲の4画素を用いて内挿補間する。

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} C_x & 0 \\ 0 & C_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \quad (6)$$

平行移動量 (t_x, t_y) 、回転角 θ 、スケール C_x, C_y の各パラメータにより、上記移動量を表現することができる。

アフィンパラメータの探索例を示す¹⁷⁾。まず平行移動パラメータを、探索範囲内を1画素単位で全探索し、候補点を求める。次に候補点に対する近傍範囲を半画素刻みごとに、 $\theta, (C_x, C_y)$ をある定めた範囲内で変動して、予測誤差評価を行い、最適なアフィンパラメータを決定する。しかし、この予測誤差最小となるパラメータ算出のため、BMと比較して膨大な演算量を要する。これらの課題を解決するため、演算量を抑えたアフィン変換パラメータ推定法¹⁸⁾が検討されている。

2.3 ベクトル探索方法

BMにおけるこの全探索方法は、ハードウェア化が容易である。しかし、大きな動きがあるシーンでは、動き予測の探索範囲を広域にする必要があり、演算量において課題を有する。演算量の軽減、および、精度の向上を狙った探索法として、テレスコピック探索(テレスコピック、テレスコピックサーチ)法、階層探索法がある。

テレスコピック探索法は、隣接フレーム間におけるオブジェクトの動きに相関があるというモデルに基づいた手法である。オブジェクトの動きが激しい場合、参照フレームと時間的に離れたフレーム間では、動き検出するために広域な探索範囲が必要となる。そこで、あるオブジェクトに対応する動きベクトルをフレーム間で順次追跡して、参照ブロック(テンプレート)位置を予測し、探索範囲の1次探索に利用する方法である。2次探索は、領域内の全探索を利用する。参照領域に対する時間的なフレームの相互関係

が、望遠鏡を覗いている状況に近いことから名付けられた方法である。さらに本手法において動きベクトル探索範囲を抑制する方法²⁰⁾が検討されている。

階層探索法は、大局的な動きと局所的な動きを考慮し、元画像に対してあらかじめ画素を間引いてダウンサンプルした画像を利用して、大局的な動きベクトルを求め、一方、元画像(高解像度画像)で局所的な動きを予測する方法である。画素を間引いた画像を利用することで、探索範囲を大きくすることが可能である。探索範囲を絞るだけでなく、異なる解像度で動き予測を行うことで、動き精度の向上を図る方法²¹⁾である。

その他の手法として、符号化済み近傍マクロブロックの動きベクトルや、過去の動き領域を考慮して動的領域を探索する方法、ゼロベクトルから探索範囲を絞り込む方法などがある。

2.4 予測精度

BMにおける矩形領域(例えば、マクロブロック(16×16)画素)で用いる動きベクトルは、一般的に1/2画素(ハーフペル)精度で表される。さらに高精度な動き補償を実現するために、1/4画素精度の動きベクトル予測が利用される。例えば、8タップフィルタを利用して1/2画素精度の画素値を計算し、さらに線形補間により1/4画素精度の画素値を求める。画像の内挿補間により、画素単位で行う処理を1/2や1/4精度で適用することにより、動き抽出の精度を向上させることができる。しかし、ベクトル精度向上による計算量増加、および、その情報量増加と動き予測精度向上との関係を考慮する必要がある。

3. む す び

本稿では、画像符号化アルゴリズムの基礎技術として、動画像圧縮効率に大きな影響を与えるフレーム間予測について概説した。

動画像シーケンスの先頭やシーンチェンジなどでフレーム間の予測が不可能な場合、本予測を用いないでイントラ符号化を行う。一方で、動きを有するシーケンスでは、本予測は動画像圧縮に不可欠な技術である。

フレーム間動き予測を利用すると、符号化対象フレームは、キーフレーム(参照フレーム、あるいはイントラフレーム)との差分信号で表すことができ、情報量を大幅に削減可能である。さらに、差分信号は、直交変換、エントロピー符号化を利用して符号化される。予測に利用する領域の大きさ、予測方法などにより、動きベクトルの精度が向上する。一方で、予測誤差信号の符号量を低減できるが、動きベクトル自身の符号量が多くなる課題がある。目標符号量、動き検出処理/動き補償処理などを鑑みて、利用方法を決定する必要がある。

(2013年1月17日受付)





〔文 献〕

- 1) 高木幹雄, 下田陽久監修, 新編画像解析ハンドブック, 東京大学出版会 (2004)
- 2) 画像電子学会編, 画像電子情報ハンドブック, 東京電機大学出版局 (2008)
- 3) B.K.P. Horn and B.G. Schunk: "Determining Optical Flow", Artificial Intelligence, 17, pp.185-203 (1993)
- 4) 木村, 上野, 松田, 伊東: "可変サイズ可変形状ブロックに基づいた動き補償方式", 信学論, J80-D-II, 2, pp.434-443 (1997)
- 5) 木村, 長谷山, 北島, 白川, 小川: "動画像符号化のための動きパラメータを用いた可変ブロックサイズ動き補償", 信学論, J77-D-II, 7, pp.1273-1281 (1994)
- 6) J. Nieweglowski, T.G. Campbel, P. Haavisto: "A Novel Video Coding Scheme Based on Temporal Prediction Using Digital Image Warping", IEEE Trans. on Consumer Electronics, 39, 3, pp.141-150 (1993)
- 7) 横山, 宮本, 太田: "輪郭適合パッチを用いた適応代表点選択ワーピング動き補償予測", 信学論, J79-D-II, 1, pp.1273-1281 (1996)
- 8) M. Ohta, S. Nogaki: "Hybrid Picture Coding with Wavelet Transform and Overlapped Motion-Compensated Interframe Prediction Coding", IEEE Trans. on Signal Processing, 41, 12, pp.3416-3424 (1993)
- 9) M.T. Orchard and G.J. Sullican: "Overlapped Block Motion Compensation: An Estimation Theoretic Approach", IEEE Trans. on Image Processing, 3, 5, pp.693-699 (1994)
- 10) 甲藤: "オーバーラップ動き補償の特性解析", 信学技報, IE94-8, DSP94-8, pp.53-60 (1994)
- 11) R. Rajagopalan, E. Feig, M.T. Orchard: "Motion Optimization of Ordered Blocks for Overlapped Block Motion Compensation", IEEE Trans. on CSVT, 8, 2, pp.119-123 (1998)
- 12) M. Hotter: "Differential Estimation of the Global Motion Parameters Zoom and Pan", Signal Processing, 16, 3, pp.249-265 (1989)

- 13) 上倉, 渡辺: "動画像符号化におけるグローバル動き補償法", 信学論, J76-B-I, 12, pp.944-952 (1993)
- 14) 上倉, 渡辺, 小林, 一ノ瀬, 安田: "演算量低減を考慮したグローバル動き・輝度変化補償動画像符号化", 信学論, J82-B, 9, pp.1676-1688 (1999)
- 15) 如澤, 上倉, 渡辺, 小寺: "MPEG-4ビデオ符号化におけるグローバルMCの効果について", AVM研究会技術報告, 13-3, pp.15-22 (1996)
- 16) H. Jozawa, K. Kamikura, A. Sagata, H. Kotera and H. Watanabe: "Two-Stage Motion Compensation Using Adaptive Global MC and Local Affine MC", IEEE Trans. on CSV, 7, 1, pp.75-85 (1997)
- 17) 如澤: "アフィン変換を用いた動き補償予測に関する検討", 信学技報, IE94-36, pp.1-8 (1994)
- 18) 荒川, 安藤, 橋本: "アフィン変換を用いた動き補償予測の高速化", 映情学誌, 51, 7, pp.1114-1117 (1997)
- 19) 興梠, 村上: "グローバルなアフィン動きパラメータの実時間推定手法", 信学論, J82-D-II, 7, pp.1161-1170 (1999)
- 20) 南, 長沼: "テレスコピック探索に適した動きベクトル検出器構成法の提案", 信学論, J87-D-II, 11, pp.2007-2024 (2004)
- 21) 花村, 関口, 亀山, 富永: "解像度選択機能と互換性を有する階層的動画像符号化", 信学論, J76-B-I, 3, pp.299-310 (1993)



こだま 明 1992年早稲田大学理工学部電子通信学科卒業。1997年早稲田大学大学院理工学研究科後期博士課程了。1995年早稲田大学理工学部助手, 1998年広島大学地域共同研究センター助教授等を経て, 現在, 広島大学大学院総合科学研究科/情報メディア教育研究センター准教授。主に, 画像通信, 動画像符号化, スケーラブル符号化, 画像検索方式, マルチメディア情報構造化などマルチメディア通信サービスの研究に従事。工学博士。

講座：基礎からの画像符号化 予定目次 (全12回)

1月号	(第1回) 画像信号の基礎 (第2回) 可変長符号化	吉田俊之 (福井大学) 長谷川まどか (宇都宮大学)
2月号	(第3回) 画像符号化のための線形変換 (第4回) 変換係数の量子化	山下幸彦 (東京工業大学) 稲積泰宏 (富山大学)
3月号	(第5回) フレーム内予測 (第6回) 映像符号化におけるループ内フィルタ	松田一朗 (東京理科大学) 松尾翔平 (NTT)
4月号	(第7回) フレーム間予測 (第8回) 視点間予測・視点捕間	児玉 明 (広島大学) 石川彰夫 (情報通信研究機構)
5月号	(第9回) 符号化処理の階層構造 (第10回) 符号化装置の制御技術	市ヶ谷敦郎 (NHK) 坂手寛治 (三菱電機)
6月号	(第11回) 静止画像符号化方式 (第12回) 動画像符号化に関わる標準規格	石川孝明 (早稲田大学) 数井君彦 (富士通研究所)

視点間予測・視点補間

正会員 石川 彰夫[†]

1. ま え が き

通常、画像符号化と言えば、1視点からの画像、すなわち1台のカメラで撮影した画像を対象としていた。しかし、近年の3D映画の流行などにより、複数の視点からの映像を提示する立体映像は身近なものになり、現在ではゲーム機や家庭用テレビでも立体映像を表示できるものが珍しくなくなっている。その多くは、左目用と右目用の2視点の映像を提示するステレオ映像であるが、9視点などより多くの視点の映像を提示することで、見る位置による立体像の見え方の変化も再現できる多眼式ディスプレイも市販されている。

これらの立体映像の合成には、複数のカメラを用いて対象物体をさまざまな位置や向きからフレームを同期させて撮影した映像のまとめ（以下、多視点映像）が必要となる。例えば、ステレオ映像には2台のカメラで撮影した多視点映像が必要となり、9視点の立体映像には9台のカメラで撮影した多視点映像が必要となる。しかし、多視点映像では、映像がカメラの台数分だけ増えるので、通常の1台のカメラで撮影した映像に比べて情報量が多くなる。そこで、多視点映像を効率よく圧縮符号化する符号化方法が必要となる。

多視点映像を効率よく符号化するための国際標準の一つに、2009年に国際標準化機関MPEGとITU-Tの共同チームJVT (Joint Video Team)で標準化が行われたMVC (Multi-view Video Coding)¹⁾がある。MVCは、同じくJVTの国際標準H.264/AVC²⁾で多視点映像を扱えるように拡張した標準であり、Blu-ray 3D規格にも採用されるなど、利用が期待されている。

また、現在も符号化方式の規格化が進められている3DV (3D Video)³⁾がある。MVCでは、元となるH.264/AVCからの改造が大きくなならないよう、ブロック単位の予測符号化が採用されている。それに対して、3DVは、画像を合成して予測符号化することにより、大幅な圧縮率向上が見込

まれている。

以上、実際に標準化された技術の詳細については参考文献も参照されたい。

本稿では、多視点映像の符号化方式についての基本となる技術内容を概説する。まず、第2章で視差と奥行きについて説明し、第3章で視点間予測とそれを用いた視差補償予測符号化を説明し、第4章で画像を合成して予測をする視点補間予測を説明する。第5章で本稿をまとめる

2. 視差と奥行き

多視点映像は、異なる位置から同一の対象物体を撮影して得られる映像である。すなわち、対象物体表面の同じ領域が、位置を変えながら各視点の画像に写っている。したがって、これらの画像の間には高い相関がある。以下では、簡単のため、ステレオ画像を例として説明する。

図1のように、2台のカメラ（それぞれ左目カメラと右目カメラとする）が対象物体を撮影しているとき、左目カメラと右目カメラの位置の違いにより、対象物体を見る方向に差が生じる。これが「視差」である。

図2のように、左目カメラと右目カメラの光軸が平行で焦点距離も等しい場合を考える。また、いずれの光軸も両カメ

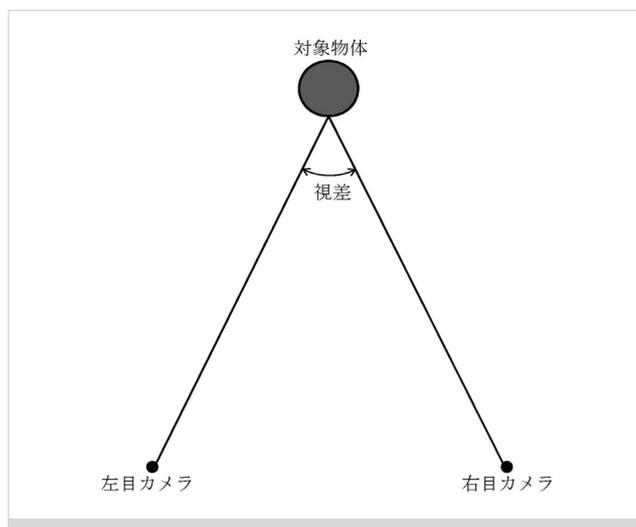


図1 視差

[†] 独立行政法人情報通信研究機構

"Basic Technologies of Image Coding (8): Inter-View Prediction and View Interpolation" by Akio Ishikawa (National Institute of Information and Communications Technology, Kyoto)

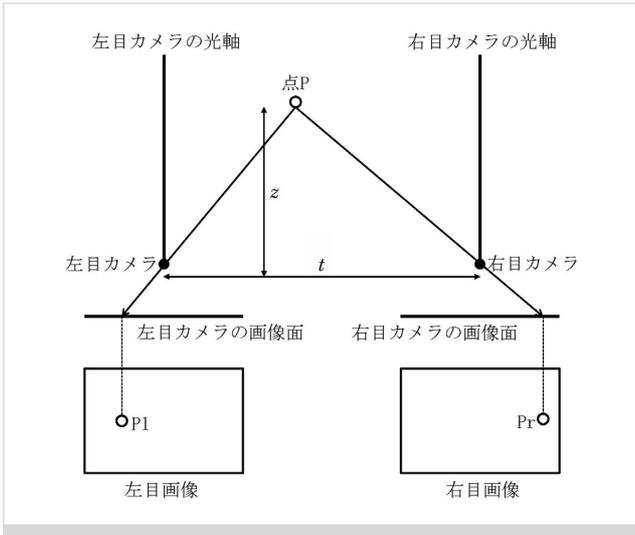


図2 ステレオ画像

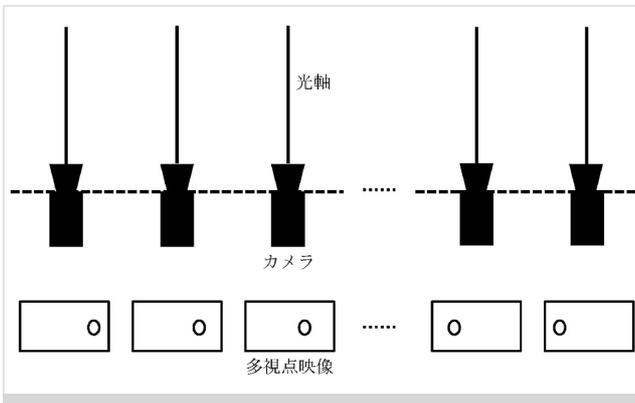


図3 多視点画像

ラが並んでいる面に対して直交しているとする。左目カメラで撮影した画像を左目画像とし、右目カメラで撮影した画像を右目画像とする。空間中の点Pから出た光が、左目カメラおよび右目カメラにそれぞれ入射し、左目画像上の点P_lと右目画像上の点P_rに写ったとする。この時、左目カメラと右目カメラの位置が異なるため、P_lとP_rの座標値が異なる。このP_lとP_rの座標値の差が視差に相当する。

左目カメラと右目カメラの間の距離をt、点Pと両カメラが並んでいる直線との距離(奥行き値)をz、両カメラの焦点距離をfとおくと、視差dは式(1)で表される。

$$d = f \frac{t}{z} \tag{1}$$

すなわち、視差は奥行き値に反比例する。したがって、視差を求めることは奥行き値を求めることと同義である。

同じ議論は、図3のように、多数のカメラが一直線上に並んでいる場合でも成り立つ。ここでも、すべてのカメラの焦点距離は等しく、光軸は平行であり、いずれもカメラが並んでいる直線に対して直交しているとする。

図3のような多視点映像においても、隣接するカメラの画像には、そのカメラ間の距離に応じて、対象物体表面の同じ領域が位置を変えながら写っている。したがって、多視点画像から対応点関係を推定し、対応点の画像中の位置の変化から視差情報を抽出することができる。また、式(1)を用いて、対応点の奥行き値を求めることもできる。

このように多視点画像から視差情報を抽出できるのであれば、ある視点の画像と視差情報とから別の視点の画像を予測することもできるはずである。これを視点間予測と呼び、視点間予測により多視点画像の情報圧縮を行う符号化方式を視差補償予測符号化(Disparity Compensated Predictive Coding)と呼ぶ。

3. 視点間予測

視点間予測とは、多視点画像において、参照視点の画像(以下、参照画像。通常、符号化済みの画像が用いられる)と視差情報を用いて符号化対象の視点の画像(以下、符号化対象画像)を合成し、予測に利用する手法である。

簡単のため、ステレオ画像を例として説明する。左目画像と右目画像からなるステレオ画像において、左目画像を参照画像とし、右目画像を符号化対象画像とする。左目画像と視差情報が与えられれば、左目画像中の対象物体の位置を視差に応じてずらす視点間予測を行うことにより、右目画像を予測合成することができる。予測合成した画像と実際の右目画像との差分、すなわち予測ができなかった部分の情報は、予測誤差情報として別途伝送すればよい。

前章で説明した通り、一般にステレオ画像の左目画像と右目画像の間には高い相関があるため、左目画像と視差情報および予測誤差情報を符号化した方が、左目画像と右目画像を独立に符号化するよりも、高い圧縮率を得ることができる。

ステレオ画像とは、カメラの数(眼数)が2台の多視点画像である。したがって、ステレオ画像を符号化する技術を2台より多い眼数の場合へと一般化して、視差補償予測符号化の方式を応用することにより、多眼画像符号化を実現することができる。多視点画像においても同様に視点間予測を適用し、多視点画像の持つ冗長性を利用して情報圧縮を行う符号化方式を利用することにより高い圧縮率を得ることができる。

ステレオ画像や多視点画像ではなく、ステレオ映像や多視点映像の場合には、動き補償予測(異なる時刻のフレームの画像を用いた予測)も併用することができる。

簡単のため、再びステレオ映像を例として説明する。図4のように、時刻t₂における右目画像は、時刻t₁における右目画像からの動き補償予測と、時刻t₂における左目画像からの視点間予測によって予測合成することができる。

第1章で紹介したMVCでは、視点間予測が利用されている。MVCで想定されている画像間の基本的な参照構造を

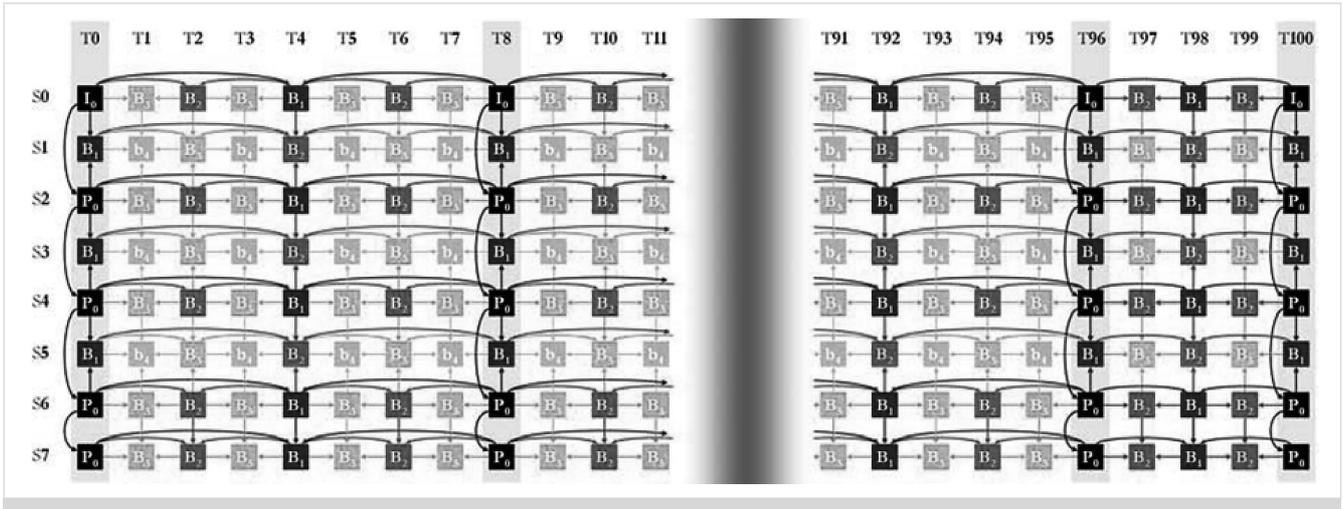


図5 MVCの参照構造

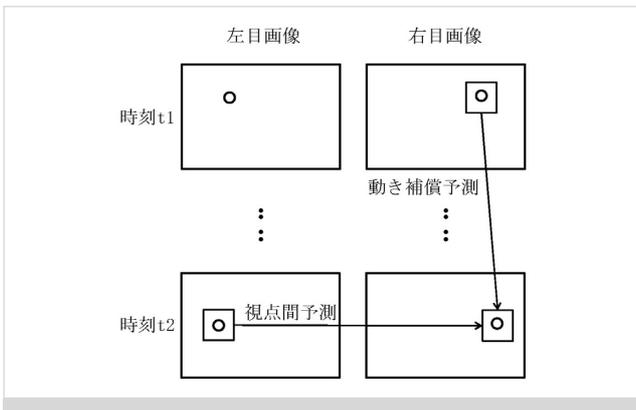


図4 動き補償予測と視点間予測

図5に示す。時間方向の動き補償予測だけでなく、視点間予測も併用する構造になっている。

4. 視点補間

前章の視点間予測で利用した視差情報は、同時に奥行き情報でもある。そこで、画像とは本来3次元空間にある物体の投影であるという考え方にに基づき、3次元の物体そのものを符号化する方法が考えられる。3次元の物体の情報を復号して各カメラの位置に投影することで、原理的には多視点画像のすべての画像が合成できることになる。このようにして合成した画像を予測に用いる方法を視点補間予測と呼ぶ。実際には、完全な合成はできず予測誤差が生じるが、3次元の物体の情報の精度を高めることで予測誤差は抑えられるので、特に視点数が多い多視点映像において圧縮効率の大幅な改善が期待できる。また、視点補間予測を応用すれば、復号側において任意視点からの画像(自由視点画像)の合成が可能となる。すなわち、実際にカメラで撮影した画像だけではなく、実際にはカメラが設置され

なかった視点からの画像を合成することも可能となり、単なるデータ圧縮に留まらない応用が可能である。

一般に、多視点画像などの入力映像から対象物体の構造情報(形状・表面の模様・材質・照明条件など)を抽出し、構造モデル(幾何モデル)情報として圧縮する符号化方式を構造抽出符号化と呼ぶ。構造抽出符号化では幾何モデルによる記述が行われるため、対象物体の表面の3次元的位置を求めることができる。したがって、参照画像と幾何モデル情報が与えられれば、参照画像と符号化対象画像の画素の対応が得られるので、参照画像から符号化対象画像を予測合成することができる。予測合成した画像と実際の符号化対象画像との差分、すなわち予測ができなかった部分の情報は、予測誤差情報として別途伝送すればよい。

幾何モデル情報の記述方法には、3Dポリゴンなどさまざまな方法があるが、近年、奥行きマップ(Depth Map)を伝送するDepth Image-Based Rendering(DIBR)方式⁴⁾が多く検討されている。幾何モデル情報として奥行きマップを用いる理由は、奥行きマップをグレースケール映像とみなすことにより、既存の映像符号化手法がその冗長性を利用して効率的な符号化を実現できるためである。そのため、既存の多視点映像符号化手法と互換性を取り易いので、奥行きマップを利用する符号化手法は3DVなどでも検討されている。

しかし、1視点の奥行きマップを伝送しただけでは、オクルージョンが発生し復元された映像に穴が生じる可能性がある。この問題を回避するため、多視点映像とその各視点に対応する奥行きマップのセットから構成されたMultiview Video plus Depth(以下、MVD)⁵⁾が提案されている。

MVDは、多視点映像とその各視点に対応する奥行きマップ(多視点奥行きマップ)のセットから構成されている。映像と奥行きマップのセットから予測画像を合成する手法

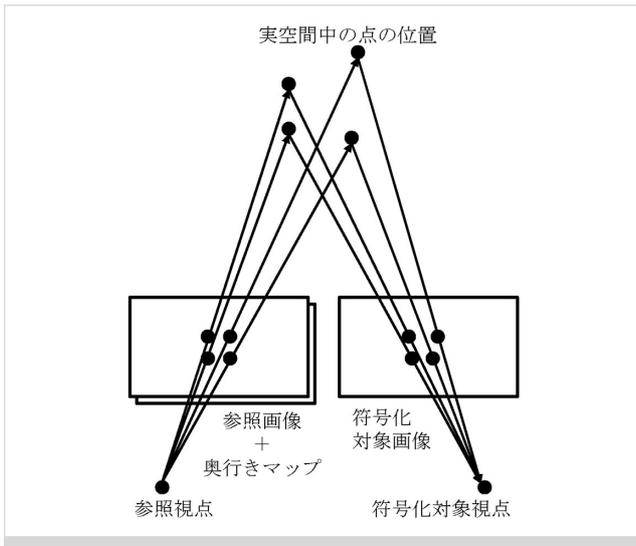


図6 3Dワーピング

奥行きマップから、奥行き値 Z が得られるため、式(2)を用いて、 $(u \ v)^T$ から X と Y を算出することが可能である。すなわち、各画素に対応する実空間中の点の位置を求めることができる。

以上が、3Dワーピングの概要である。ただし、3Dワーピングでは、実空間中の異なる点が復元された画像中の同一点に写ることがあり、必然的に実空間中のいずれの点も写らず情報が欠落する点あるいは領域が、復元された画像中に発生する。したがって、それらの点あるいは領域を補完する処理が必要となる。

5. むすび

本稿では、多視点映像符号化に必要な技術として、視点間予測と視点補間について概説した。詳細については、参考文献などを参照されたい。
(2013年2月20日受付)

〔文 献〕

- 1) ISO/IEC 14496-10:2009 Coding of audio-visual objects - Part 10: Advanced Video Coding, Annex H: Multiview Video Coding
- 2) ITU-T and ISO/IEC JTC 1: "Advanced video coding for generic audiovisual services", ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC) (2010)
- 3) "Call for Proposals on 3D Video Coding Technology", ISO/IEC JTC1/SC29/WG11 N12036 (2011)
- 4) C. Fehn: "Depth-image-based rendering (dibr), compression and transmission for a new approach on 3d-tv", in SPIE, USA, pp.93-104 (2004)
- 5) P. Merkle, K. Dix, K. Muller, A. Smolic, T. Wiegand and P. Kauff: "Intermediate view interpolation based on multi-view video plus depth for advanced 3d video systems", ICIP, pp.2448-2451 (2008)
- 6) Y. Mori, N. Fukushima, T. Fujii and M. Tanimoto: "View generation with 3D warping Using depth information for FTV", Image Communication., 24, 1-2 (2009)

の一つに、3Dワーピング⁶⁾がある。図6に3Dワーピングの概要を示す。

各視点の映像の各画素の奥行き値は、同じ視点の奥行きマップを参照すれば得られるので、逆射影変換によってその画素に対応する実空間中の点の位置を求めることができる。実空間中の位置が求められれば、通常の射影変換により符号化対象画像の予測画像を合成することができ、その予測画像と予測誤差情報とから多視点映像を復元することができる。

ここで、逆射影変換の概要を説明する。一般に、実空間中の点 $(X \ Y \ Z)^T$ が画像中の点 $(u \ v)^T$ に写っている場合、 $(X \ Y \ Z)^T$ と $(u \ v)^T$ の間には、カメラによって一意に決まる 3×4 の行列(射影行列) \mathbf{P} を用いて、式(2)の関係が成り立つ。

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{P} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (2)$$

ここで、 λ はスケールの不定性を表すスカラーである。



いしかわ あきお
石川 彰夫 1995年、東京大学工学部電子情報工学科卒業。1997年、同大学大学院工学系研究科修士課程修了。同年、国際電信電話(株)に入社。同社研究所(現、(株)KDDI研究所)に配属。2012年、(独)情報通信研究機構にて、主に、自由視点映像生成、多視点映像符号化、3次元映像処理の研究・開発に従事。正会員。

符号化処理の階層構造

正会員 市ヶ谷敦郎[†]

1. ま え が き

符号化処理により圧縮されたストリームには、復号に必要な情報が記述されている。これらの情報は、同じ処理を行う単位ごとにまとめて共通化することにより冗長とならないように設計されている。そのため、符号化処理は、映像信号全体から、細かい処理へと入れ子状の階層構造で進められる。本稿では、この符号化処理における階層構造について解説する。符号化処理としてデジタル放送で用いられているMPEG-2およびAVC/H.264を例に解説するが、用語の混同を生じる部分についてはMPEG-2をベースとする。

実際の符号化ビットストリームの構造をみると、階層構造をよく把握できる。図1に、具体的な例としてMPEG-2のビットストリームの構成を示す。図に示すように、シーケンス層、GOP (Group of Picture) 層、ピクチャ層、スライス層、マクロブロック層、ブロック層の6層に分類されている。各層には、ヘッダと呼ばれる共通の符号化制御情報が記述され、低位層から必要に応じて参照される。多少の違いはあるもののAVC/H.264もほぼ同様に分類される。

このような階層構造は、情報記述の冗長性を排除するには効率的であるが、一方で情報にエラーが生じた場合、関連する層全体に伝搬する。特に上位層における信号のエラーは符

号化処理全体に与える影響が大きくなることから、例えばMPEG-2の場合、スライス層以上のレイヤにはヘッダ部に32ビットの固定長のスタートコードが定義され、ビットストリームにエラーが生じても直近のスタートコードを検出することによりエラーから復帰することができる。このことから、スライスがエラー回復の最小単位となっている。次章よりこれらの各層の構成と役割について解説する。

2. 符号化処理の階層構造

2.1 シーケンス層

シーケンスヘッダと一つ以上のGOPで構成される層である。シーケンスヘッダは、シーケンスに共通の情報として映像信号の解像度やフレームレートなどのフォーマット情報、符号化で用いているツールセットを識別するプロファイルやレベルなどの情報が格納されている。原理上、これらの情報に変更がなければ、ビットストリームの先頭で一度伝送するのみで再送の必要がなく、ビットストリームの切替えや映像フォーマット、プロファイルやレベルに変更が生じた場合に新たなヘッダが送信される(図2)。

実用上は、エラー耐性の向上、放送におけるチャンネル切替え動作(ザッピング)など、受信機入力となる信号の切替えが頻繁に行われることが想定され、また、放送サービスではビットストリームの先頭から信号を受信することは極めて稀であるため、一定の間隔でシーケンスヘッダが送信される。デジタル放送では、運用規定においてシーケンスヘッダを0.5秒以内(ワンセグの場合は通常2秒、最大5秒以内)に一度送出することが定められている。

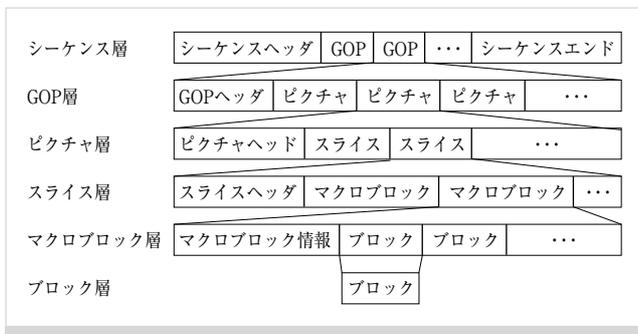


図1 MPEG-2ビットストリーム構成例

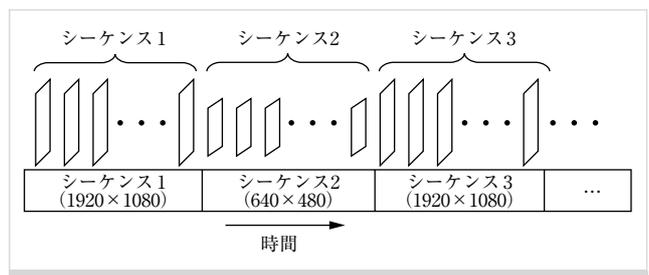


図2 シーケンスの例

[†] NHK 放送技術研究所

"Basic Technologies of Image Coding (9): Multi-layered Structure of Encoding Processing" by Atsuro Ichigaya (NHK Science & Technology Research Laboratories, Tokyo)

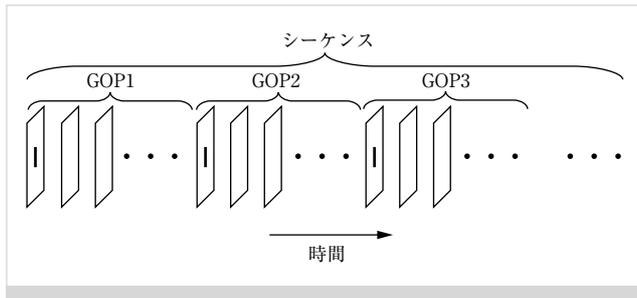


図3 GOP層の例

2.2 GOP層

GOP層は、GOPヘッダと1枚以上のピクチャデータによって構成される符号化処理層である(図3)。GOPヘッダには、スタートコードやタイムコード情報、当該GOPの独立性を示す識別フラグなどを格納している。GOPは、他フレームを参照せずに単独で復号可能なイントラピクチャ(Iピクチャ:次章参照)を必ず1枚含むように構成される。GOPの最初の復号可能なピクチャはこのIピクチャであり、Iピクチャから次のIピクチャの前までがGOPの範囲となる。

2.3 ピクチャ層

ピクチャは、1枚のフィールドまたはフレームで構成される符号化処理単位である。①ピクチャ単位で当該ピクチャを単独で復号可能なイントラ符号化によって処理するか、②フレーム間の依存関係を利用した動き補償予測によるインタ符号化処理を用いるか、を切替える。それぞれの適用の仕方の違いによって、3タイプのピクチャ(I/P/Bピクチャ)が規定されており、ピクチャヘッダには当該ピクチャのピクチャタイプやGOP内での表示順を示すインデックスが格納されている。

2.3.1 Iピクチャ

I(Intra Coded)ピクチャは、単独で復号可能なピクチャである。他ピクチャと依存関係がなく、他ピクチャの復号状況に関わらず復号可能である。一方で、動き補償予測を用いる後述するPまたはBピクチャによって参照される。このことから、ストリームの切替えなどの際の復帰点として機能し、GOPの先頭に必ず配置される。PピクチャまたはBピクチャが参照するために復号後、デコーダのフレームメモリに一定期間保存される。動き補償予測を用いないため、エラー耐性は高いが、符号化効率は低いという特徴を持つ。

2.3.2 Pピクチャ

P(Predictive Coded)ピクチャは、前方向の動き補償予測を用いたピクチャである。復号処理時にデコーダのフレームメモリに保存されているピクチャを参照し、復号する。参照ピクチャとして直前に復号された1枚のIまたはPピクチャを用いる。時間的に過去のピクチャを参照することから単方向予測符号化画像と呼ばれる。Iピクチャと同様、PまたはBピクチャによって参照されるため、復号後

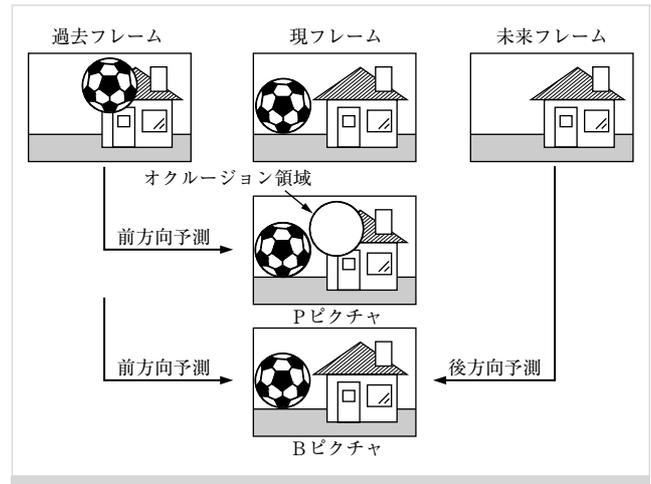


図4 動き補償予測の効果

デコーダのフレームメモリに保存される。

2.3.3 Bピクチャ

B(Bi-directional Predictive Coded)ピクチャは、両方向の動き補償予測を用いたピクチャであり、Pピクチャと同様デコーダのフレームメモリに保存されているピクチャを参照し、復号する。Pピクチャとの違いは参照するピクチャとして直前に復号された最大2枚のIまたはPピクチャを必要とする。MPEG-2では、当該ピクチャよりも過去および未来に位置するIまたはPピクチャを参照することから、両方向予測符号化画像と呼ばれる。AVC/H.264では参照する2枚のピクチャを、過去または未来の2枚以上のピクチャの中からブロックごとに選択できるように拡張されたため、方向に関係なく双予測ピクチャ(Bi-Predictive Picture)と呼ばれる。

2.3.4 Bピクチャの効果と遅延

映像コンテンツでは、時間とともに変化する映像信号をピクチャ間で予測する場合、過去のピクチャを参照することだけでも大部分の信号を予測することができる。しかし、単方向予測では重なりあった物体など、参照ピクチャには存在しなかった(見えなかった)領域(オクルージョン領域)を予測することができない。このような場合、異なる時間のピクチャをさらに用いるとオクルージョン領域を参照することができ、予測効率を改善することができる。そのためBピクチャの圧縮率は高く、符号化効率を重視する場合、Bピクチャが多用される(図4)。

しかし、未来のピクチャを事前に参照するためには、符号化時に入力信号を一定枚数フレームメモリに蓄積し、順番を並び替える必要がある。この符号化処理のために入れ替えたピクチャの順序を符号化順とよび、ディスプレイへの表示順と区別する。表示順と符号化順の対応を図5に示す。このように表示順と異なる順序で符号化を行うことによって、未来に位置するピクチャの参照が実現される。一方で、一定期間ピクチャの蓄積を必要とするため、符号

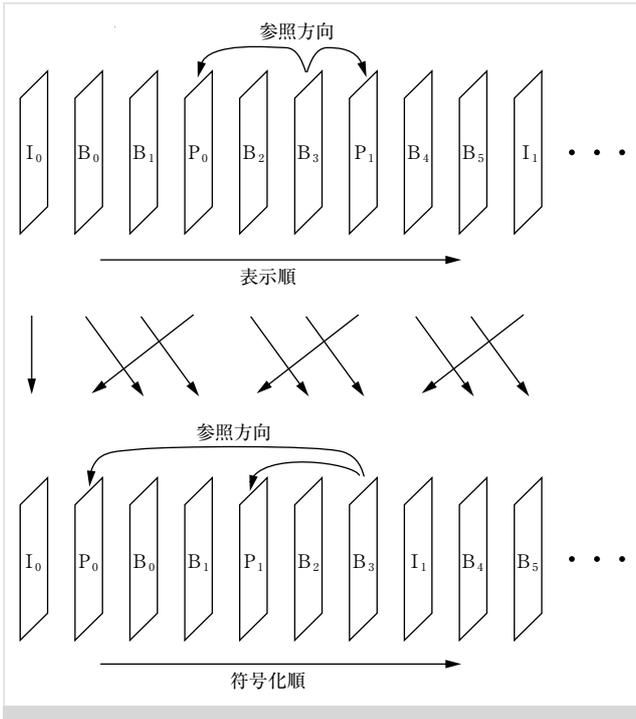


図5 表示順と符号化順

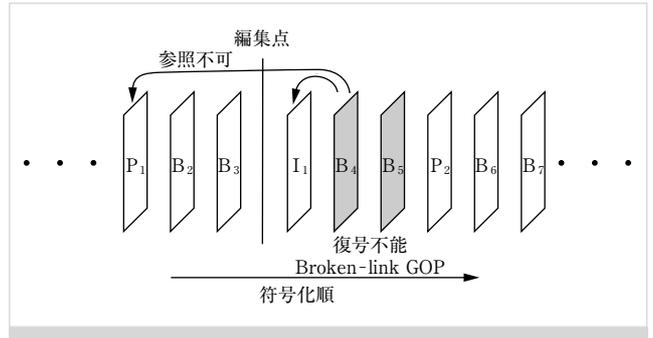


図6 Broken link GOPの例

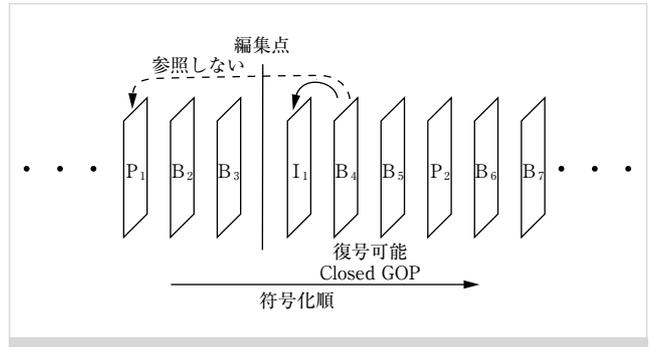


図7 Closed GOPの例

化処理による遅延の原因となる。

2.3.5 GOP構成と編集

ところで、図6の符号化順では、B4、B5ピクチャはI1、P1ピクチャを参照する。2枚のピクチャを参照するB4、B5ピクチャは、所属するGOPの先頭のI1ピクチャをまたいで、時間的に過去のフレームP1を参照する。GOPの切れ目において編集を行うとB4、B5はP1を参照できず復号できない。このような編集によって参照不可能なピクチャを含むGOPを、Broken link GOPと呼び、B4、B5ピクチャを正常に復号できないことを示す識別フラグがGOPヘッダに設定される。しかし、図7のようにB4、B5ピクチャで前方向予測を制限し、P1を参照せずにI1のみを参照するように制御するとGOP単位できれいに編集が行える。このようなGOPをClosed GOPと呼び、GOPヘッダにB4、B5ピクチャが符号化可能であることを示すことを識別するフラグが設定される。

前述したように、Closed GOPでない限りどこに編集点を設定しても数枚のフレームが復号不能となる。しかし、編集業務においては編集点を自由に設定できることが重要である。そこで編集用途では、すべてのフレームを単独で復号可能なIピクチャで符号化した信号を用いることが多い。このようなGOP構造をM=1、N=1あるいはオールイントラ符号化と呼ぶ。M値はPピクチャの間隔を示し、N値はGOP(Iピクチャ)の間隔を示す値である。一方、放送サービスでは圧縮効果を最大とするためにM=3、N=15(Pピクチャの間隔が3フレーム、Iピクチャの間隔が15フレーム)が広く利用されている。

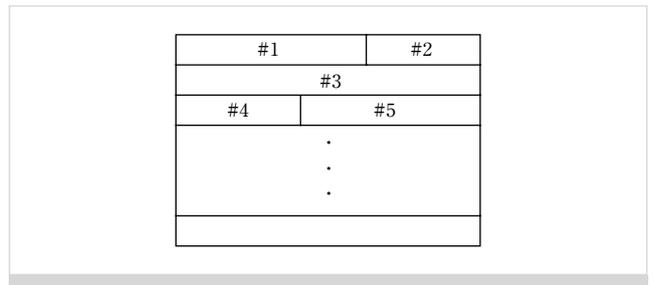


図8 MPEG-2におけるスライス分割例

2.4 スライス層

スライス層は、スライスヘッダと一つ以上のマクロブロック(後述)で構成される。固定長のスタートコードを持つ最下層のレイヤであり、ストリームにエラーが発生しても次のスライスでデコード処理を正常に復帰できる。スライスヘッダには当該スライスのピクチャ内における位置情報などが含まれる。

MPEG-2のスライス構成例を図8に示す。スライスとはマクロブロック幅1行に相当する短冊状の領域で定義される。エラー耐性の向上を目的としてスライスが設定されるが、ピクチャを過剰にスライス分割するとスライスヘッダが増え符号化効率の低下を招く要因となる。

AVC/H.264のスライス構成例を図9に示す。AVC/H.264ではイントラ予測が導入され、上または左に隣接するマク

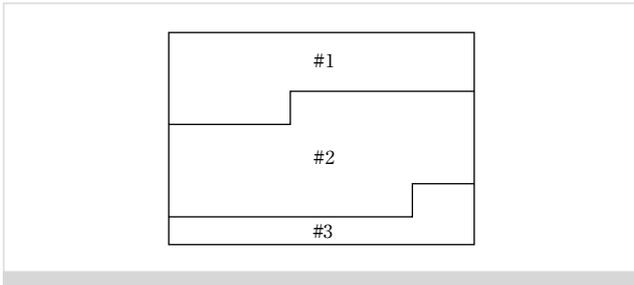


図9 AVC/H.264におけるスライス分割例

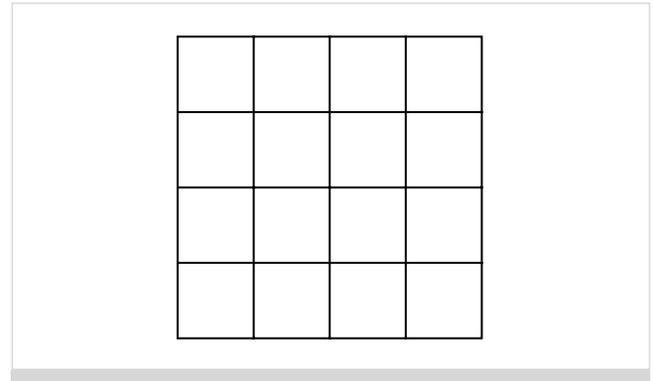


図11 マクロブロック構成例

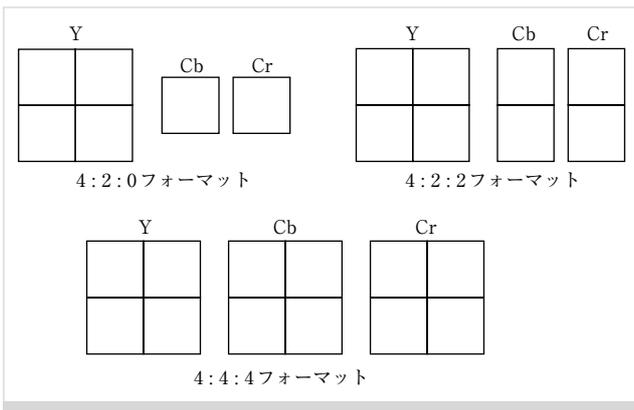


図10 マクロブロック構成例

3. 符号化構造を利用した応用 (インタレース対応)

前述した符号化構造を用いることにより、符号化方式をさまざまなアプリケーションに柔軟に対応させることができる。映像信号には走査線を1ラインずつ飛び越し走査するインタレース信号と、順次走査するプログレッシブ信号がある。映像符号化方式ではこれらの異なる性質の信号を統一的に扱えるような工夫が加えられている。

インタレース信号は、二つのフィールドによって1フレームを構成する信号である。フィールドはフレームを構成する走査線の奇数ラインと偶数ラインのいずれかで構成されており、フィールド間に時間差が生じている。動きの少ない映像であればこの差はほとんど感じられず、1枚のフレームとして高精細な映像が提示できる。しかし、動きの激しい映像では、例えば30fpsのインタレース映像では、1/60秒のフィールド間の時間差が映像の差となって感じられる。つまり、垂直解像度が半分だが1/60秒間隔(60fps)の映像の提示が可能となっている。

このようなインタレース信号を符号化する場合、次のような課題が生じる。動きの速い映像の場合、フレーム単位で信号を見ると、フィールド間の時間差から図12(左)に示すような、二重にぶれた映像となる。動きの激しい映像におけるインタレース映像特有のぶれは、フィールド単位で見ると図12(右)に示すように、それぞれのピクチャからぶれを除去することができる。ぶれは符号化効率の低下を生じるため、このようにフィールド単位でピクチャを構成すると予測もしやすく、符号化を効率的に行うことができる。フレーム単位のピクチャ構成をフレームピクチャ、フィールド単位のピクチャ構成をフィールドピクチャと呼び、それぞれのピクチャを映像信号にあわせて適応的に切替えることができる。このような符号化をピクチャ適応フレーム/フィールド符号化(PAFF: Picture Adaptive Frame-Field Coding)と呼ぶ(図13)。

同様に画面の一部において動きの大きい映像では、画面の一部でのみぶれの生じた映像が提示される。このような場

マクロブロックを参照する。そのためスライスを上下にまたぐ処理が必要となった。そこで、AVC/H.264におけるスライスは図に示すように、MPEG-2における隣接する複数のスライスを接続した領域をスライスとして設定できるように拡張されている。

2.5 マクロブロック層

マクロブロックは動き補償予測を行う最小単位である。MPEG-2, AVC/H.264におけるマクロブロックは輝度信号の 16×16 画素の正方領域で構成され、マクロブロックヘッダに動き補償予測モード、動きベクトル、量子化値、DCTタイプなどの情報を格納する。MPEG-2では、一つのマクロブロックは輝度信号のブロック四つで構成されるが、色差信号のマクロブロックは、サブサンプリングフォーマットの違いによって図10に示すように構成される。

2.6 ブロック層

ブロックはマクロブロックを構成する正方形領域であり、変換処理を行う単位である。前述の通りMPEG-2では輝度信号では四つのブロックでマクロブロックを構成している。AVC/H.264ではマクロブロックを四つに分割したブロックに加え、図11のようにマクロブロックを最大16個のブロックに分割可能なように拡張されている。柔軟なブロック分割により、より入力信号に適した符号化が実現されている。

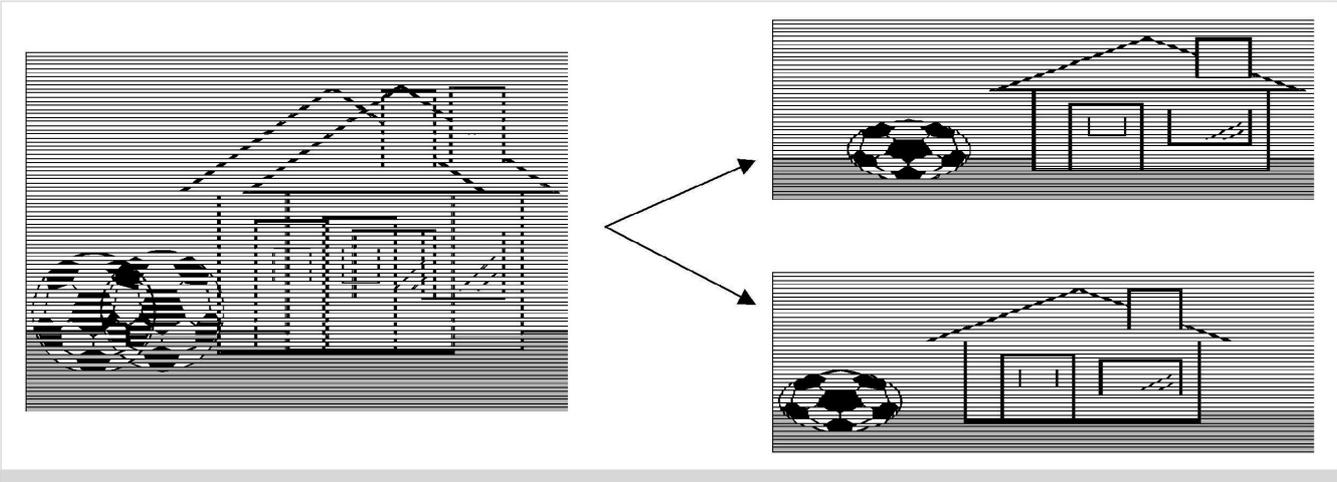


図12 インタレース信号のピクチャ構成例

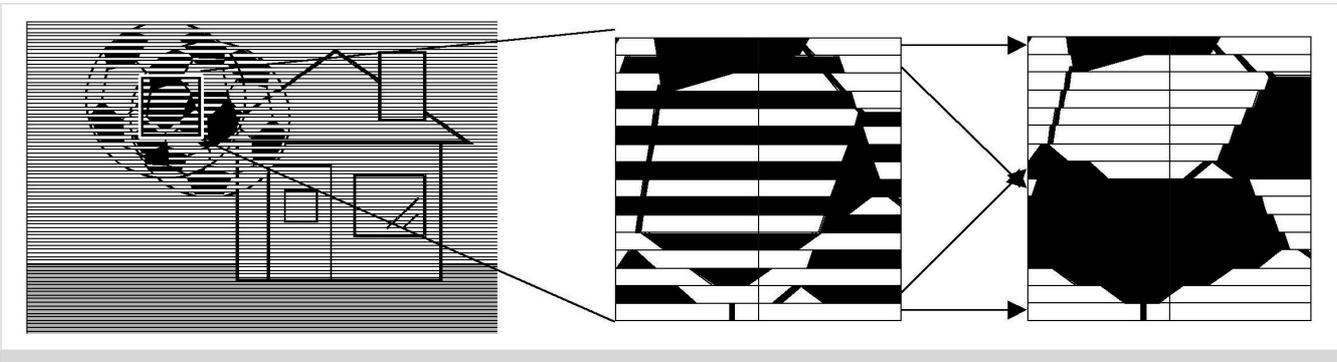


図14 MBAFFの例

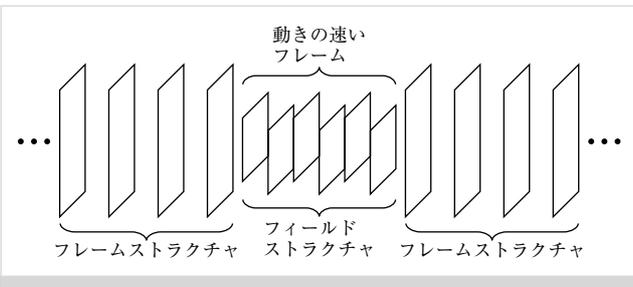


図13 PAFF例

4. む す び

本稿では映像符号における階層処理について解説した。階層構造によって効率よく、かつ、さまざまなアプリケーションに適用することができることを示した。本稿はMPEG-2とAVC/H.264を念頭に執筆したが、本講座の目的として符号化方式に依存しない基礎的な仕組みの解説を心がけた。そのため、AVC/H.264で高度に拡張された機能については解説せず、MPEG-2をベースにした部分も多い。AVC/H.264の高度な拡張については詳しくは参考文献を参照されたい。

(2013年1月31日受付)

【文 献】

- 1) 映情学編：“総合マルチメディア選書 MPEG”，オーム社
- 2) 大久保栄，角野真也，菊池義浩，鈴木輝彦：“H.264/AVC教科書”，インプレスR&D
- 3) ビーター・サイムス：“デジタル映像圧縮の基礎”，兼六館出版



いちがや じゅんろう
市ヶ谷敦郎 1998年、筑波大学理工学研究所修了。同年、NHKに入局。名古屋放送局を経て、2000年より、放送技術研究所。画質評価、映像符号化方式の研究開発に従事。現在、テレビ方式研究部にて、次世代映像符号化方式などに従事。正会員。

合、マクロブロックでフレーム/フィールド適応を行うこともできる。マクロブロック単位で信号に合わせてフレーム、フィールドで処理を切替えることをMBAFF (Macro Block Adaptive Frame-Field Coding) と呼ぶ(図14)。

このように符号化処理の階層構造を利用することによりピクチャ単位、マクロブロック単位で高効率な符号化制御を行うことができる。プログレッシブ信号だけでなくインタレース信号も効率的に取り扱えるなど、柔軟な符号化処理構造は、コンピュータだけでなく、放送やビデオ機器などさまざまなアプリケーションで広く映像符号化技術が普及した理由となっている。



符号化装置の制御技術

正会員 坂手 寛治†

1. ま え が き

画像符号化は、テレビ放送やインターネットでの動画配信、携帯電話などわれわれが日常で接する機器やサービスのほか、商業施設やプラント等で稼動する監視システム、各所自治体で運用されている防災システムなど非常に多用途で使用されている。

AVC/H.264やHEVC¹⁾などの符号化方式を使用する場合、各方式の勧告書(Recommendation)にしたがって設計された符号化装置(エンコーダおよびデコーダ)を用意することになる。勧告書には主にビットストリームのデータ構造や復号方法が記載されているが、符号化装置の設計方法や制御方法に関する条件は少なく、設計者やユーザの裁量に委ねられている部分が多い。

本稿では、用途や目的に合った符号化装置を実現するための制御技術について解説する。また、画像符号化以外の付加機能として、多くの符号化装置に搭載されている機能についても紹介する。

2. 画像符号化の制御

2.1 レート制御

映像の圧縮率はコンテンツによって異なる。例えば、動きの複雑な映像や高周波成分の電力が大きい映像は圧縮効率が悪く、ビットレートが高くなりやすい。一方、映像をリアルタイムに伝送したり、媒体に記録したりする場合、通信路の帯域や品質、媒体の書き込み速度等によってビットレートに制約を受ける。そのため、エンコーダから出力される符号化データ(ビットストリーム)の速度を所望のビットレートに制御するレート制御技術が必要となる。

レート制御はフィードバック方式とフィードフォワード方式の二つに大別される。

(1) フィードバック方式

MPEG-2 TM-5²⁾に代表される方式である。エンコーダに

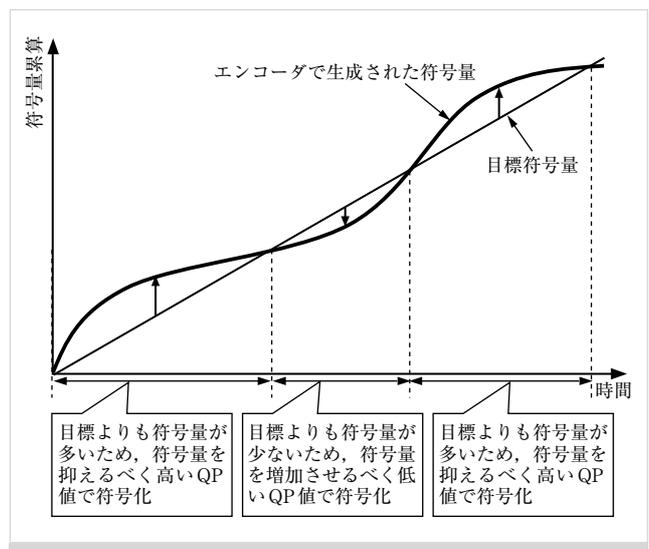


図1 フィードバック方式の例

において実際に生成された符号量と目標符号量との差分を求め、差分が小さくなる方向に符号化パラメータを切替えていくことで、所望のビットレートを実現する。

図1にフィードバック方式の具体例を示す。グラフは、エンコーダで生成された符号量および目標符号量の符号化開始時刻からの累算を表している。生成された符号量が目標符号量を上回った場合は、QP(Quantization Parameter)値を大きくすることで符号量の発生を抑える。逆に、目標符号量を下回った場合は、QP値を小さくすることで符号量を増加させる。

本方式では、符号量の差が生じた後に補正を行うため、レート制御に破綻が生じる場合がある。例えば、入力映像が符号化難易度の高いシーンに切替わった場合、QP値を上昇させる前に、所望のビットレートを上回る符号量が生成されてしまうケースがある。

(2) フィードフォワード方式

2パスエンコード³⁾に代表される方式である。2パスエンコード方式は、正規の符号化処理に先立ち仮符号化を実行する手法である。仮符号化の結果から目標符号量に収束させるために適切な符号化パラメータを決定し、正規の符号化に反映させる。フィードバック方式に比べて高精度な制

†三菱電機株式会社 情報技術総合研究所

"Basic Technologies of Image Coding (10): Control Techniques for Video Codec" by Hiroharu Sakate (Information Technology R&D Center, Mitsubishi Electric Corporation, Tokyo)

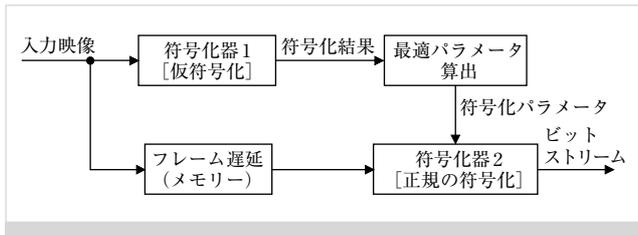


図2 リアルタイム2パスエンコーダの構成例

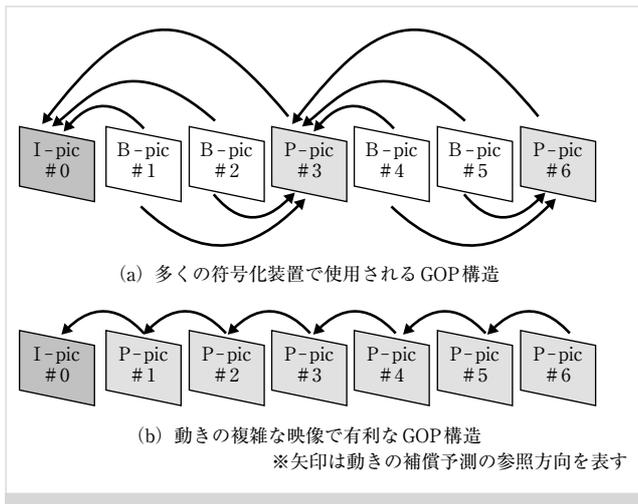


図3 GOP構造のバリエーション例

御が実現できるが、符号化を複数回実行するため、実行時間や演算量が大きくなるデメリットもあり、主にオフラインでの符号化に用いられる。

リアルタイム処理に対応した2パスエンコード構成を図2に示す。入力映像を符号化する前にフレーム遅延を設け、その期間に別の符号化器で仮符号化を行う。リアルタイム処理の制約上、入力映像の全フレームを仮符号化の対象とすることは困難であるが、数フレームから数秒分の映像を仮符号化しておくことで、シーンチェンジ等によるビットレート増加の可能性を事前に検知し、あらかじめ符号量を抑えておくといった対処が可能となる。

2.2 高圧縮制御/画質制御

同じ映像を同じビットレートで符号化する場合でも、GOP (Group of Pictures) 構造や符号化パラメータの選び方によって圧縮率や画質が変化する。入力映像に応じて符号化条件を切替えることで圧縮率や画質を向上させることが可能である。

図3に示す二つのGOP構造はいずれもAVC/H.264やHEVCで使用可能である。図3(a)は多くの符号化装置で使用されているGOP構造であるが、符号化フレームと動き補償予測における参照フレームとの距離が離れているため、絵柄が激しく変化するような映像の場合、予測が当たらないケースがある。そのような映像では、図3(b)のように参照フレーム距離が近いGOP構造が有利となる。ただ



図4 適応量子化の実施例

し、図3(b)は双方向予測を使用しないため、動きの緩やかな映像では不利となるケースが多い。高圧縮制御の一つの方法として、入力映像の動きの複雑さに応じてGOP構造を切替えることで効率の良い符号化が実現できる。

主観画質を向上させるための手段として適応量子化が有効である。適応量子化は、画面内で量子化精度を切替え、着目点や画質劣化が目立ちやすい部分の画質を維持する手法である。図4に実施例を示す。低ビットレートで符号化するようなケースでは、適応量子化を用いることで、見た目の印象を飛躍的に向上させることができる。

2.3 遅延制御

画像符号化の実行は遅延を伴う。符号化装置の用途によっては遅延が無視できないケースもある。

フレームレートが30fpsの映像をAVC/H.264により符号化した場合の遅延量を図5に例示する。高速な符号化装置を使用した場合でも、エンコーダとデコーダの合計で600~1,000ms程度の遅延が発生する。

遅延の主な要因は、双方向予測に伴うフレーム並び替えと送受信バッファでの待ち時間である。図6に各遅延について示す。なお、符号化や復号はマクロブロック単位のパイプライン処理により1ms以下の遅延で動作させることが可能である。表示処理では、復号した映像をモニタの表示タイミングに合わせて出力するような場合に最大33msの遅延が生じる。

低遅延化のための代表的な符号化方法として、イントラスライス方式またはGDR (Gradually Decoder Refresh) 方式と呼ばれる方式がある⁴⁾。Iピクチャを用いず、図7に示すように、Pピクチャの内部にイントラブロックのみで構成されるエリアを設け、同エリアの位置を巡回させながら符号化する。通常、Iピクチャはビットストリームを任意の位置から復号するためのアクセスポイントとして必要となるが、イントラスライス方式のビットストリームを任意の位置から復号した場合も、イントラブロックエリアが一周した時点から正しく復号可能となる。

イントラスライス方式では、全フレームが同様の符号化条件となるため、フレームごとに符号量のばらつきが生じ

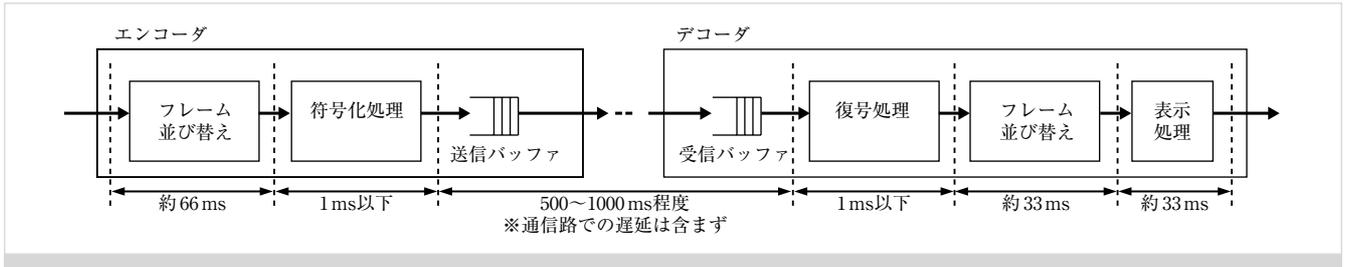


図5 符号化装置の遅延量

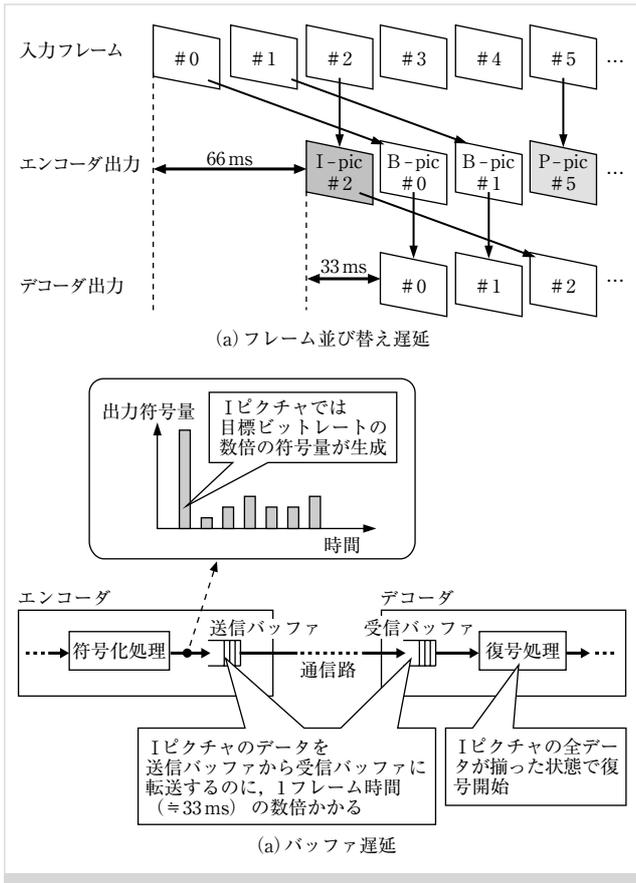


図6 フレーム並び替え遅延とバッファ遅延

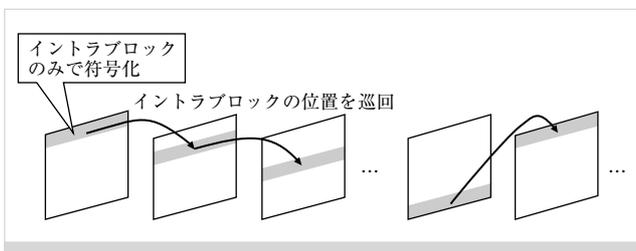


図7 イントラスライス方式

にくい。そのため、バッファ遅延を100ms程度に抑えることができる。加えて、前方予測のみに制限することでフレームの並び替えも省略できる。これらの結果、約200msの遅延量で符号化および復号処理が可能となる。

10msクラスの超低遅延化手法も提案されている⁵⁾。イン

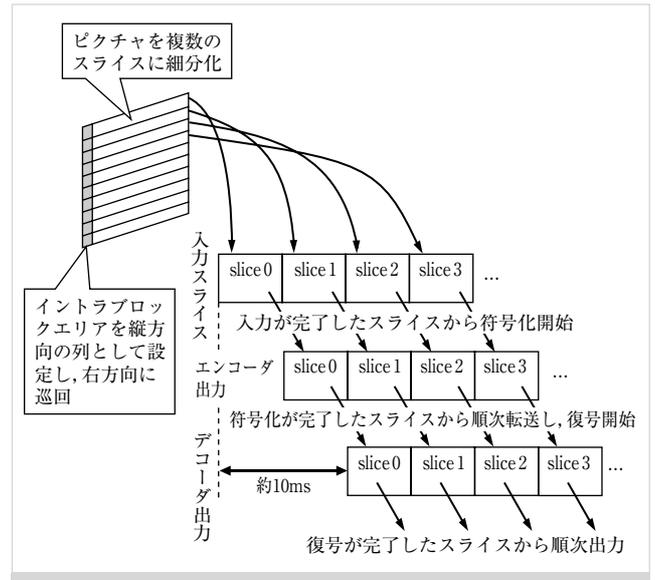


図8 超低遅延方式

トラスライス方式を改良したものであり、図8のようにピクチャを複数のスライスに細分化し、スライス単位で順次符号化・復号を実行していくことで低遅延を実現している。イントラブロックエリアを縦の列として設定し、水平方向に巡回させることで、各スライスのイントラブロック数を均一化し、スライス間の符号量のばらつきを抑える工夫がなされている。

なお、イントラスライス方式は通常のGOP構造よりも圧縮性能が低くなる傾向があるため、用途によって使い分けることが望ましい。

3. 符号化装置に搭載される機能

高性能で利便性の高い符号化装置を実現するためには、レート制御や遅延制御などの画像符号化に対する直接的な制御のほか、多数の機能が必要となる。

符号化装置に搭載される主な機能を表1に示す。以下、概要を説明する。

(1) 異常検出

映像信号の入力異常を検出し、フェールセーフを実現する機能である。符号化装置に入力される映像信号は必ずしも正常なものとは限らず、信号が途切れたり、同期ずれが生じたりすることもある。



表1 符号化装置の主な機能

エンコーダ	入力処理	異常検出
エンコーダ	前処理	解像度・色変換
		画像補正(帯域制限, ノイズ除去)
	符号化・データ処理	画像符号化
		音声・音響符号化 補助データ処理
伝送・記録処理	メディア多重・分離	
デコーダ	後処理	伝送制御
		符号化歪み除去 エラーコンシール
	出力処理	メディア間同期(A/V同期)
		出力同期

放送システムなど安定動作が要求される用途では、入力信号に対する位相調整機能やAVDL (Automatic Variable Delay Line) と呼ばれる映像のラインずれ吸収機能を搭載し、異常が発生しても動作を継続できる構成とすることが多い。

(2) 解像度・色変換

HDTV映像における水平解像度(1920/1440画素)の変換や、色差成分における4:2:2形式と4:2:0形式の変換、RGB空間からYUV空間への変換などのフォーマット変換機能である。多くの符号化装置ではこれらを基本機能としてサポートしている。

(3) 画像補正

入力された映像信号を符号化しやすい状態に補正する機能である。例えば、帯域制限フィルタ(ローパスフィルタ)による高周波成分の減衰や時間フィルタによるランダムノイズの除去により、映像信号のエントロピーが下がり、高い圧縮性能が得られやすくなる。

(4) 符号化・データ処理

音声・音響信号の符号化や、補助データ(タイムコードやメタデータ等)の生成を行う機能である。映像と音はセットで扱われることが多く、音の符号化は必須となることが多い。

(5) メディア多重・分離

符号化した映像、音、補助データ等の各メディアを1本のビットストリームに多重したり、デコーダ側にて各データをビットストリームから分離したりする機能である。

多くの符号化装置では、MPEG-2 System規格(ISO/IEC 13818-1/ITU-T H.222.0)に準拠したトランスポートストリーム(TS)形式のビットストリームを扱い、同規格に従った多重・分離処理を行う。

(6) 伝送制御

ビットストリームの伝送制御機能である。ベストエフォート回線で伝送する場合、通信状態を監視し、レート制御にフィードバックするような機能が有効となる。

(7) 符号化歪み除去

符号化によって生じる歪みやボケ感を補正する機能である。例えば、リングングノイズの除去にはε-フィルタ、ボケ感の改善にはアンシャープマスクなどの手法が有効である。符号化歪みの種類に応じて複数の補正方法を使い分けることも多い。

(8) エラーコンシール

正常に復号できず出力映像が異常となった場合に、それを補正する機能である⁶⁾。低品質な通信路やベストエフォート回線で伝送する場合、ビットストリームの一部欠落や、復号に必要なデータが所望の時刻までに到達しないことにより、復号時にエラーが発生することがある。そのような場合に、正常に復号できたフレームを繰り返し出力したり、エラーにより異常となった箇所を近隣のフレームから補完したりすることで、出力映像の見栄えを改善する。

(9) メディア間同期

映像と音、あるいはそれらに付随する補助データの出力タイミングを合わせる機能であり、A/V同期とも呼ばれる。テレビ会議など双方向で通信するようなアプリケーションやテレビ放送などでは、リップシンク(Lip-Sync)が求められる。数ms精度の同期合わせが必要となる。

(10) 出力同期

外部から入力された同期信号にタイミングを合わせて映像を出力する機能である。本機能は、複数の機器を同期運転させるような場合に必要となる。同期信号には、B.B (Black Burst) 信号や3値同期信号などが用いられる。

4. む す び

画像符号化に対する主要な制御技術と、多くの符号化装置に搭載されている機能について解説した。各技術の詳細は個々の専門文献を参照されたい。(2013年2月25日受付)

〔文 献〕

- 1) 関口俊一：“映像符号化の標準化動向～次世代国際標準HEVC～”，信学技報，110，456，pp.85-90 (2011)
- 2) MPEG-2 Test Model 5, <http://www.mpeg.org/MPEG/MSSG/tm5/>
- 3) H.B.Yin, X.Z. Fang, L. Chen and J. Hou: "A practical consistent-quality two-pass VBR video coding algorithm for digital storage application", IEEE Trans. Consumer Electronics, 50, 4, pp.1142-1150 (2004)
- 4) 大久保榮ほか：“H.264/AVC教科書”，pp.192-195，インプレス(2004)
- 5) 牧野鉄雄：“低遅延技術の開発とテレビ番組制作への応用(前編)”，映情学誌，64，7，pp.983-986 (2010)
- 6) Y. Wang, S. Wenger, J. Wen, A.K. Katsaggelos: "Error resilient video coding techniques", IEEE Signal Processing Magazine, 17, 4, pp.61-82 (2000)



坂手 寛治 1998年、大阪大学大学院基礎工学研究科博士前期課程修了。同年、三菱電機(株)に入社。同社情報技術総合研究所にて、画像符号化に関する研究開発に従事。正会員。

静止画像符号化方式

石川 孝明[†]

1. ま え が き

インターネットが普及し、膨大な量の画像や動画がデジタルデータとして流通している。多くの人がパソコンやスマートフォンを常用し、ブラウザを開けば文字と画像で美しくレイアウトされたコンテンツが表示され、スマートフォンのカメラで気軽に写真を撮り、友人と共有できる。この一連の流れの中で、一般のユーザが符号化という言葉を意識する瞬間は存在しない。符号化の存在を感じる人は、一部の研究者や業界関係者、そして本稿の読者だけと推察される。それほどに符号化技術は広く社会に浸透しているといえる。

本稿では、符号化技術の集合体である静止画像符号化方式について解説する。JPEG, JPEG 2000, そしてJPEX XRの3つの方式について構造を概説し、特徴となる処理部や機能について述べる。最後に、国際標準化について述べる。

2. JPEG 符号化方式

JPEG (Joint Photographic Experts Group) は、ほとんどのデジタルカメラで採用されている画像を圧縮するための符号化方式である。JPEGの特徴として、高い圧縮率と演算量の低さがある。性能と処理負荷のバランスの良さから、後継となる方式のJPEG 2000やJPEG XRが作られた現在においても、依然として幅広いアプリケーションに採用されている。

2.1 要素技術

JPEG符号化方式は、以下に示す複数の要素技術に支えられている。各要素を組合せることで高い圧縮率を実現している。

- ・ 色空間変換
- ・ DCT (Discrete Cosine Transform)
- ・ DCT係数の量子化
- ・ 予測符号化

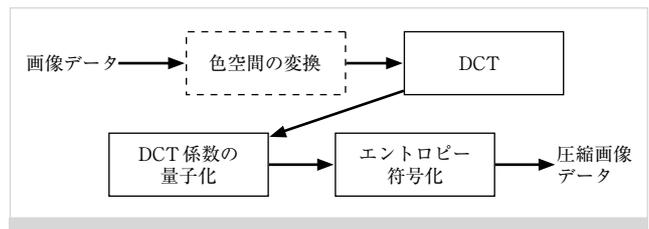


図1 JPEG画像の符号化手順

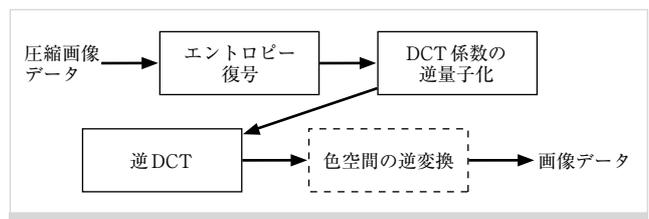


図2 JPEG画像の復号手順

- ・ ランレングス符号化
- ・ ハフマン符号化

JPEG符号化方式の符号化器と復号器の処理フローを、図1と図2に示す。デジタルカメラと画像閲覧ソフトを例にすると、まず、デジタルカメラに組込まれた符号化器が、撮影と同時に圧縮された画像データを生成する。その後、パソコンに取り込まれた圧縮画像データは、復号器を実装したソフトウェアによって画像データに戻されディスプレイに表示される。

2.2 圧縮率を高める工夫

図1と図2に示した処理フローには4つの処理部しか示されていない。しかし、各処理部には高効率な符号化を実現するために、入力画像の特徴や人間の視覚特徴を利用するさまざまな工夫が施されている。以下にそれらを示す。なお、他の符号化方式にも類似した処理部が含まれている。

2.2.1 色空間変換と間引き

デジタルカメラで撮影した瞬間の画像は、RGBの3色で表現されるが、JPEG符号化では始めにRGB色空間をYCbCr色空間に変換する。これは、人の視覚特性が輝度成

[†] 早稲田大学 国際情報通信研究センター

"Basic Technologies of Image Coding (11): JPEG Still Image Coding and Standardization" by Takaaki Ishikawa (Global Information and Telecommunication Institute, Waseda University, Tokyo)



分(Y)に対して強く反応し、色差成分(CbおよびCr)に対して弱く反応する特性を利用するためである。Y成分はそのまま残し、色差成分(CbおよびCr)を間引きすることで、視覚的な影響を最小限に抑えつつ、情報量を大幅に削減できる¹⁾。なお、色空間変換と信号の間引きは、本来のJPEG符号化方式では定められておらず、JPEGで圧縮したデータを扱うためのファイルフォーマットに定められている²⁾。

2.2.2 DCT

YCbCrの各成分を8×8画素ごとのブロックに区分して、Discrete Cosine Transform (DCT)を施す。DCTは、入力となる信号をコサイン波の成分量を表す係数値の集まりとして表現する。各周波数成分には、空間的な変化の乏しい波に対応する係数から変化の激しい波に対応する係数が存在し、最も変化の乏しい波に対応する係数をDC成分と呼び、それ以外をAC成分と呼ぶ。すなわち、64個の画素値は1つのDC成分と63個のAC成分に変換される。

自然画像にDCTを適用すると、DC成分が大きな値を持ち、AC成分が小さな値になる性質がある。さらに、人間の視覚特性には小さな波の変動を敏感に検知できないという性質があるため、AC成分の情報がある程度失っても劣化として知覚されにくい。このような特性を利用して係数を量子化する。

2.2.3 エントロピー符号化

量子化された係数群から冗長な情報を取り除き、より少ないビット数で画像を表現する仕組みがエントロピー符号化である。DCT係数のDC成分は、隣接するブロックのDC成分と近い値を取る傾向がある。これは、局所的な画素値の変化が急激には変化しない自然画像において顕著に現れる。この特性を利用して、DC成分を予測符号化する。一方、AC成分はランレングス符号化とハフマン符号化を組合せてビット列へと変換する。

2.2.4 EOB (End of Block)

AC成分の符号化では、DCTで得られた係数値を効率よく扱うために、ジグザグスキャンとEOBを使う³⁾。図3(a)にDCT後の係数の状態を概念図として示す。2次元平面において、DC成分は左上に現れ、それ以外の数値はAC成分である。係数値を量子化すると、絶対値が小さな数値は0に置き換わるため、図3(a)の破線で括られた領域には0が集中することになる。

この特性を圧縮処理で利用するために、まず図3(b)のように、係数値をジグザグスキャンする。ジグザグスキャンは、2次元の数値を1次元のデータに変換するための処理である。量子化済みの2次元DCT係数のブロックに適用することで、1次元データとして並べられる係数値はある係数以降がすべて0になることが多い。そこでJPEG符号化では、これ以降は0が並んでいることを示す印としてEOBを導入している。EOBを導入することで、係数に0が並んでいることを少ない情報量で表現可能になる。

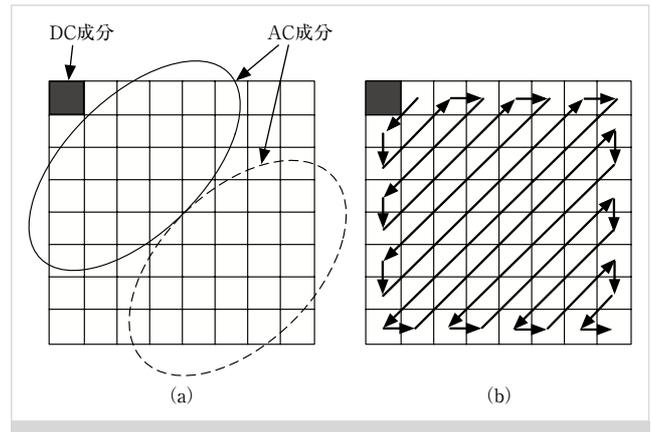


図3 係数成分とジグザグスキャン

3. JPEG 2000 符号化方式

前章で示したJPEG符号化方式と比較して、JPEG 2000符号化方式は符号化効率の向上に加えて、圧縮画像データの機能性を重視した符号化方式である。また、JPEG符号化方式の欠点である8×8画素に区分することで生じるブロックノイズの発生を本質的に解決し、低ビットレートでも画像が激しく崩れないなどの点で改善されている。

JPEG 2000符号化方式に関連する規格は、13個のパートから構成されている⁴⁾。そのうち符号化方式の中心を担う部分は、Part1: Core Coding Systemとして規格化されている。Part2以降には、動画像のファイルフォーマット、セキュリティ技術、APIや通信プロトコルへの拡張などが定義されている。

3.1 要素技術

JPEG 2000符号化方式は、以下の要素技術で構成されている。可逆符号化と非可逆符号化を同じ処理構造で実現している点や、ビットレートを指定して符号化できる点でJPEG符号化方式と違いがある。また、JPEGと比較してほぼ半分のビットレートで符号化できる。

- ・色空間変換
- ・DWT (Discrete Wavelet Transform)
- ・タイリング
- ・係数モデリング
- ・ビットプレーン展開と量子化
- ・エントロピー符号化
- ・符号順序制御

JPEG 2000符号化方式の符号化器の処理フローを、図4に示す。復号器は、JPEGと同様に符号化器の処理を逆向きにたどる。ただしJPEG 2000符号化方式には、符号化器側に符号順序を制御する処理が加えられている。そのため、エントロピー符号化が施された後のビット列を組み替えるだけで、目的の異なるアプリケーションに利用できる。

圧縮画像データの一部を復号して情報を抽出する機能を

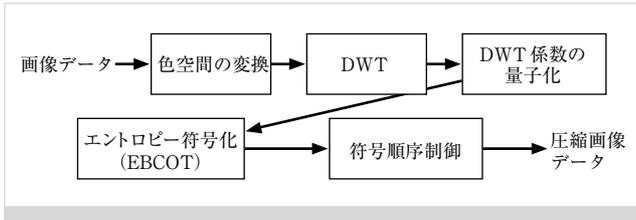


図4 JPEG 2000画像の符号化手順

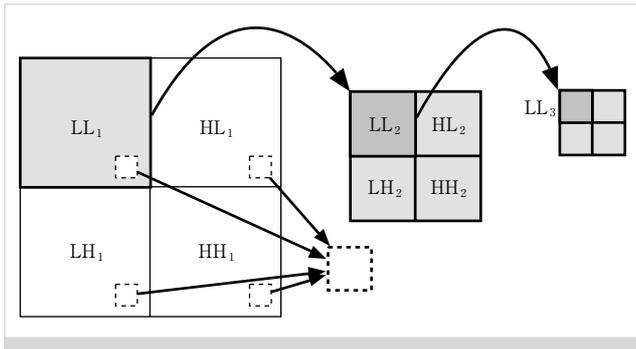


図5 入力画像のサブバンド分割

スケーラビリティと呼ぶ。JPEG 2000符号化方式には、空間スケーラビリティおよび品質スケーラビリティがあり、画像の一部の領域だけを復号することや、特定の色コンポーネントだけを復号することも可能である。

3.2 圧縮率を高める工夫

JPEG 2000符号化方式の圧縮率を高める工夫のうち、2次元ウェーブレット変換とEBCOT (Embedded Block Coding with Optimized Truncation) について述べる。

3.2.1 2次元ウェーブレット変換

JPEG 2000符号化方式では、入力画像データを区別せず2次元のウェーブレット変換を画像全体に施す。これにより、JPEG符号化方式で生じていたブロックノイズの問題を本質的に解決している。実際の処理手順は、水平方向と垂直方向のそれぞれに1次元のウェーブレット変換を施す。その結果、4つの領域に低周波数成分と高周波数成分に対応する係数が並ぶことになる。ウェーブレット変換後のDWT係数の状態を図5に示す。信号処理の分野ではこのような周波数分割をサブバンド分割と呼んでいる。

左上のサブバンドは低周波数成分に対応し、人間の視覚には縮小画像のように見える。その他のサブバンドは、原信号のエッジ成分と細かな変動を表現している。低周波数成分のサブバンドを再帰的に2次元ウェーブレット変換すると、入力信号のエネルギーが少数の係数で構成される領域に集約される。その結果、JPEG符号化と同様に、人間の視覚特性を利用した量子化が可能になる。また、高周波数成分のサブバンドでは、エッジ成分が大きな値を持つ係数

として抽出されているため、効率のよい圧縮が可能になる。

なお、JPEG 2000符号化方式では、入力画像を一定サイズに区切り、それぞれを独立な入力画像として扱うこともできる。この機能はタILINGと呼ばれる。極めて高解像度な入力画像を効率よく符号化するために有効であるが、圧縮率を高めるとJPEG符号化同様にタイルの境界に視覚的なノイズが生じる。

3.2.2 EBCOT

JPEG 2000符号化方式では、2次元ウェーブレット変換で得られた係数群を効率良く符号化するために、EBCOT (EmbeddedBlock Coding with Optimized Truncation) と呼ばれるエントロピー符号化を導入している。EBCOTでは、まず係数群を 32×32 や 64×64 などの、コードブロックと呼ばれる一定サイズのブロックに区分する。次に、ブロック内の係数値を2進数で表してレイヤ化するビットプレーン展開を施し、レイヤごとに周辺ビットの関連性を3つの符号パス (significance propagationパス, magnitude refinementパス, cleanupパス) を用いて表現する。これは、係数値そのものではなく、係数の状態を符号化することを意味している。係数モデリングの結果を算術符号化することで、高い圧縮効率を実現している。

3.3 機能性

再帰的な2次元ウェーブレット変換とEBCOTにおけるコードブロック単位の処理により、JPEG 2000符号化方式は複数のスケーラビリティを有している。符号化におけるスケーラビリティは、ビット列の一部を復号するだけでもアプリケーションに利用でき、残りのビット列があればその精度が向上する機能である。ビット列を追加的に復号することで、空間解像度や画像品質が変化する。また、図5の破線で表したように、画像の部分的なアクセスを可能にしている。

3.4 空間スケーラビリティ

図5に示したサブバンドをLL3からLL1に向かって順番に符号化すれば、縮小画像に対応する情報がビット列の前半部に集約される。例えば、分割レベル2で得られる7つのサブバンド (LL3, HL_k, LH_k, HH_k (k=2, 3)) に対応するビット列だけを復号すれば、原画像と比較して解像度が半分の画像を再生できる。

3.5 品質スケーラビリティ

ウェーブレット係数をビットプレーン展開し、最上位ビット (MSB: Most Significant Bit) から順番に符号化することで品質スケーラビリティを実現している。例えば、ある係数値が100である時、表1のようにMSBだけを符号化

表1 ビットプレーン展開の例

MSB(64)	(32)	(16)	(8)	LSB(4)
1	1	0	0	1



すれば、64を復号できる。MSBから2つ目のビットまでを符号化すれば96を復号できるため、係数値の真値である100に近づくことがわかる。

4. JPEG XR 符号化方式

JPEG XR符号化方式は、JPEGとJPEG 2000の中間的位置づけにある方式である。符号化効率はJPEGより高く、JPEG 2000に及ばない。その一方で、スケーラビリティなどの機能性を有し、JPEG 2000よりも低い処理量で符号化でき、ハードウェアでの実装が容易であるという利点がある。JPEG 2000符号化方式がJPEG符号化方式の後継規格として利用されていない一つの理由として、JPEG 2000符号化方式の処理の複雑さが指摘されている。後に規格化されたJPEG XR符号化方式は、この欠点を解決した方式である。

Microsoft社が開発したHD Photoを基盤とするJPEG XR符号化方式は、Microsoft社のブラウザやAdobe社のFlash Playerでサポートされ、ハードウェア化も開始されているため⁵⁾、今後の普及が期待される。

4.1 要素技術

JPEG XRは以下の要素技術で構成される。JPEG 2000と同様に、可逆符号化と非可逆符号化を同じ処理構造で実現でき、空間および品質スケーラビリティを有する。

- ・色空間変換
- ・重複双直交変換
- ・可変サイズタイリング
- ・係数予測符号化
- ・ハーモニック量子化
- ・動的な係数スキャン
- ・エントロピー符号化
- ・符号順序制御

JPEG XR符号化方式の符号化器の処理フローを、図6に示す。復号処理は他の方式と同様に符号化の逆処理である。

4.2 低処理負荷で圧縮率を高める工夫

JPEG XR符号化方式は、圧縮効率を高めるために周波数変換、量子化、エントロピー符号化を施す点でJPEGとJPEG 2000と同じだが、各処理部では処理負荷を低く抑えた方式が採用されている。

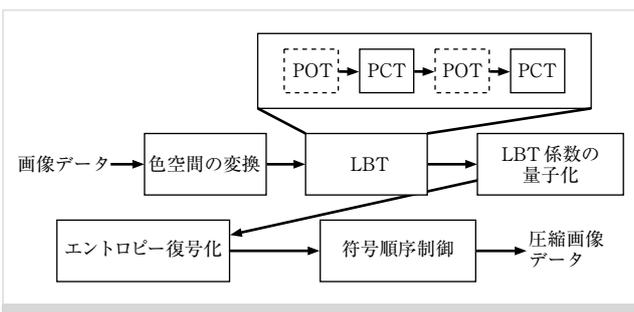


図6 JPEG XR画像の符号化手順

4.2.1 重複双直交変換

JPEG XRでは、JPEGのように入力画像を一定のブロックサイズに分割して独立に処理する。16×16画素の領域をマクロブロック (MB) と呼び、MBをさらに4×4画素に分割した領域において周波数変換を施す。JPEG XRの周波数変換は、重複双直交変換 (LBT: Lapped Biorthogonal Transform) を採用している。実際の変換処理は、POT (Photo Overlap Transform) と PCT (Photo Core Transform) の2つの処理の組合せを2回繰り返す。POTは4×4画素のブロック境界をまたぐ処理のため、高圧縮時に境界に生じるノイズを低減する役割がある。一方、PCTを2回実行することで3レベルの階層化を実現している。POTとPCTの内部は、2×2の2次元アダマール変換や係数値のスケーリング、係数の入れ替えなどの単純な演算処理で構成されるため、処理負荷を低く抑えることができる。なお、POTを常に実行する必要はない。

4.2.2 エントロピー符号化

LBTで得られる係数を予測符号化する。3レベルの階層のうち、最も低い周波数成分を表すDC係数は、上に位置するMBからの予測、左に位置するMBからの予測、上と左の両方のMBからの予測、予測なしの4つの予測モードで符号化される。また、それ以外の周波数成分を表すLP係数とHP係数については、水平方向と垂直方向の予測と予測なしの3つの予測モードがあり、最適なモードが動的に選択される。予測後の係数はランレングス符号化もしくは適応的可変長符号化される⁶⁾。適応的可変長符号化では、複数の可変長符号テーブルを予め定義されており、符号化処理の過程で適応的にテーブルを選択する。そのため、入力画像の特性が局所的に変化する場合でも、その性質に近いテーブルを選ぶことで符号化効率を高められる。また、最適な符号を符号化時に都度算出する処理ではないため、処理の複雑化を緩和し、特にハードウェア実装において優位である。

5. 国際標準化方式

前章までに解説した3つの符号化方式は、いずれも国際標準化された画像符号化方式である。方式名称であるJPEGは、Joint Photographic Experts Groupという標準化組織の名前でもある。JPEGの規格は、ISO/IEC JTC 1/SC 29とITU-Tの下部組織による共同作業で策定されている。

本稿で述べた各符号化方式の主なアプリケーションを表2にまとめる。各方式が広く利用されている背景として、国際標準化の議論に製品化を具体的に検討する団体や組織が参加していることがあげられる。

5.1 標準化の流れ

符号化の国際標準化は以下に示す手順に沿って進められる。最終的な方式を示した仕様書が完成するまでには、(4)と(5)のプロセスが数回繰り返され、その度に国際会議で





表2 応用アプリケーション

JPEG	JPEG 2000	JPEG XR
デジタルカメラ ブラウザ 画像ソフト全般	デジタルシネマ アーカイブ 免許証	ブラウザ

審議し、各国が投票権を持つ投票を経て規格が成立する。

- (1) 市場の要求や既存技術の分析
- (2) 要求機能の実現可能性の検証
- (3) 提案方式の募集
- (4) 提案方式の比較(実験と評価)
- (5) 仕様書の草稿の作成および修正
- (6) 仕様書の完成

5.2 コンフォーマンス

コンフォーマンスは、国際標準の符号化方式に対応した機器やソフトウェアを開発する時に、それらが正しく動作するかを検証するための適合試験である。実装の正しさを検証する手段が広く提供されることは、標準に準拠した製品を作るために重要な役割を果たす。また、標準化の過程においてもコンフォーマンスに関する議論を深めることで、規格書の解釈の違いから生じる問題が明らかになり、より明確な記述に改善できる。JPEG 2000符号化方式では、Part4にコンフォーマンスが規定されている。

5.3 参照ソフトウェア

参照ソフトウェアは、コンフォーマンスと同様に、標準に準拠した製品を作るために参考とするべき実装例である。例えばJPEG 2000ではPart5で参照ソフトウェアを規定しており、C言語で実装されたJasperと、Javaで実装されたJJ2000の2つが提供されている。

5.4 プロファイル

符号化方式の標準では、復号器の構成だけを規定するのが一般的である。これは、符号化器に対する制約を弱め、

独自のアルゴリズムを適用しやすくするためである。どのように入力画像が符号化されようとも、復号器が必ず画像を再生できればよいとの考え方である。符号化器に対する自由度は、符号化器を作るベンダー同士に競争をもたらし、各社、より符号化効率の高い、もしくは、高速に動作するアルゴリズムの考案を目指すことになる。

しかし、アプリケーションに応じて機能を絞り込んだり、復号器の実装規模を制限することを目的として、符号化器が生成するビットストリームに対して一定の制約が必要な場合がある。符号化方式の標準では、このような条件をプロファイルとして規定している。例えば、JPEG 2000符号化方式には2つのプロファイルが規格化されており、画像サイズやコードブロックサイズを制限している。上位のプロファイルは下位のプロファイルを包含するように規定されている。

6. む す び

本稿では、国際標準化された画像符号化方式としてJPEG, JPEG 2000, そしてJPEG XRについて解説した。

(2013年3月6日受付)

【文 献】

- 1) 末松, 山田: “画像処理工学”, コロナ社 (2000)
- 2) E. Hamilton: "JPEG File Interchange Format (JFIF)", C-Cube Microsystems (1992)
- 3) 小野, 渡辺: “国際標準画像符号化の基礎技術”, コロナ社 (1998)
- 4) 野水: “JPEG2000符号化方式解説”, トリケップス (2003)
- 5) 小高, 佐藤: “JPEG XRとそのデジカメへの適用”, 第33回VMA研究会 (July 2012)
- 6) 原: “JPEG XR画像符号化方式と性能評価”, トリケップス (2010)



石川 孝明 いしかわ たかあき 2005年、早稲田大学大学院国際情報通信研究科修士課程修了。2007年、同大学国際情報通信研究センター助手。2010年、同センター招聘研究員となり、現在に至る。画像符号化の研究に従事。ISO/IEC JTC1/SC29/WG11/Video小委員会およびWG1小委員会各委員。



動画像符号化に関する標準規格

数井 君彦†

1. ま え が き

本講座では、11回にわたり、画像符号化に用いられる各種技術を解説してきた。最終回では、動画像符号化技術の実用化とサービス普及に大きく貢献してきた、標準規格について解説する。本稿では主に、1990年代初頭から2000年代末までの間に、主要な標準化団体であるISO/IEC JTC1*1、ITU-T*2、SMPTE*3でそれぞれ規格化された、MPEGシリーズ*4、H.26xシリーズ、およびVC-1について、用途、要素技術(符号化ツール)、圧縮性能等を解説していく。

これら標準規格の基本構成は、1990年代初期にはすでに固まっており、概ねY/Cb/Cr色表現、エントロピー符号化、周波数変換、量子化、ピクチャ内(イントラ)予測、ピクチャ間(インタ)予測、ループ内フィルタといった符号化ツールから構成され、最新規格HEVCでも同様である。図1および図2にそれぞれ、エンコーダ、およびデコーダの概略構造を示す。点線で囲った箇所は、両者で共通である。

各標準規格の基本構成は同じでも、IT技術の進歩、および各符号化ツールの地道な最適化によって、動画像解像度と圧縮効率は向上している。

2. 標 準 規 格

2.1 規格の範囲

動画像符号化の標準規格は、符号化動画像データ(ビットストリーム)の構造、および符号化動画像データの復号

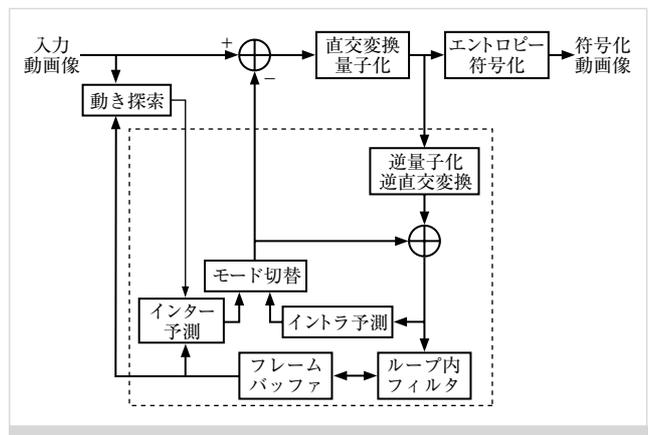


図1 エンコーダ構造

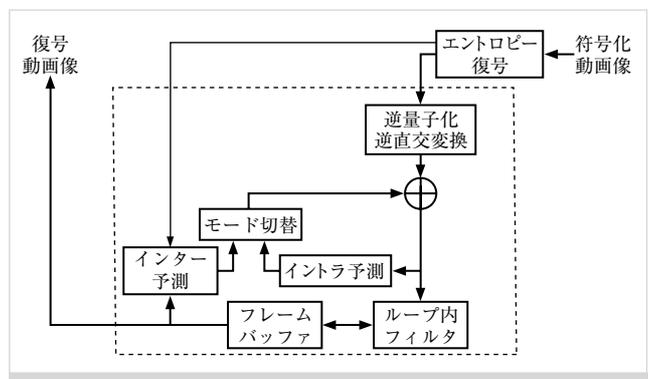


図2 デコーダ構造

- *1 ISO/IEC JTC1：国際標準化機構 (International Organization for Standardization) と国際電気標準会議 (International Electrotechnical Commission) との第一合同技術委員会 (Joint Technical Committee)。
- *2 ITU-T：国際電気通信連合 (International Telecommunication Union) の電気通信標準化部門 (Telecommunication Standardization Sector)。
- *3 SMPTE：米国映画テレビ技術者協会 (Society of Motion Picture and Television Engineers)。
- *4 MPEGシリーズ：本規格化を担当した作業グループSC29/WG11のニックネームMPEG (Moving Picture Experts Group) が、規格のニックネームにもなっている。なお、MPEG規格は、動画像符号化以外にも、システム多重化やオーディオ符号化等の規格も含んでいる。

† 株式会社富士通研究所

"Basic Technologies of Image Coding (Final Study): Video Coding Standards" by Kimihiko Kazui (Fujitsu Laboratories Ltd., Kanagawa)

方法を規定の対象とする。入力動画像の符号化方法は規定の対象としない。この理由は、動画像の符号化処理は非常に複雑であり、かつ処理量が膨大なことに起因する。製品の圧縮率および実装コスト(両者は一般的に相反)への要求は、用途によって多岐にわたるため、特定の符号化方法を規定することは不適切であり、また事実上不可能である。逆に、各社製品の自由な性能競争が可能になり、動画像符号化市場の発展につながっている。

標準規格はまた、相互接続性(同一のビットストリームの復号結果が復号装置によらず同一であること)の実現支



援を目的に、参照ソフトウェアおよび試験用ビットストリームを規定する。参照ソフトウェアは、元々標準化作業の中で、開発方式の性能評価のために開発されたものである。

なお、各標準規格の間での互換性は、基本的にはない。

2.2 プロファイル

1990年代後半以降に策定された標準規格は汎用性を持ち、多様な動画画像フォーマット、ビットレート、符号化ツールをサポートする。製品がこれらすべてを満たすことはコスト的に困難であるため、標準規格では、プロファイルおよびレベルという仕組みを規定している。プロファイルは、主要用途毎に必要な符号化ツールのセットを、レベルは動画画像解像度やビットレートといった符号化ツールに依存しないパラメータのセットを、それぞれ階層的に規定する。標準規格は、あるプロファイルレベルに準拠した復号装置が、それより下位のプロファイルレベルに準拠したビットストリームを復号可能であることを義務付けている。

2.3 ライセンス

標準規格には、各企業・研究組織が保有する特許技術が、非常に多く含まれている。各標準化団体は、特許のライセンスには関与しないが、規格に含まれる技術は「妥当で公平なライセンス」(Reasonable and Non Discriminatory Licensing: RAND)であることを、提案企業(ランセンサ)に要請している。一方、製品を開発する企業(ライセンシー)が、すべてのランセンサに特許使用の個別交渉を行うことは、現実的ではない。そこで、ランセンサがライセンスプール(代表格はMPEG-LA)を立ち上げ、ライセンス処理を一括して行う仕組みを構築している。

3. 各標準規格の概要

以下、標準化時期の順番に紹介していく。

3.1 H.261

H.261は、ITU-T(当時はCCITT)によって1990年に策定された規格である。用途は、デジタル回線上での実時間通信である。64kbpsのISDN回線を複数束ねて、QCIF(176×144)およびCIF(352×288)解像度の動画画像(4:2:0, 8ビット)を、1Hz～30Hzのフレームレートで伝送することを想定している。企業向けのテレビ会議システム等で製品・実用化されている。

H.261は、動画画像符号化の主要符号化ツールである、Y/Cb/Cr色表現、可変長符号化、直交変換(8×8、浮動小数点演算)、量子化、インタ予測を備えている。H.261の特徴は以下の通り。

- ・過去方向のみのインタ予測：処理量が増加する両方向は、実時間処理を優先するためサポートしない。
- ・マクロブロック(MB)を符号化単位とするピクチャ内分割：輝度成分の大きさは16x16画素であり、これは本解説で紹介しているすべての標準規格で共通である。
- ・Groups of Blocks (GOBs) 構造：スライス構造(画面内

分割)の原型、各GOBの位置と大きさは固定。

- ・整数精度の動きベクトル(MV)：小数精度とした以降の規格と同等な効果をもたらす、予測画像への2次元Low-pass filter (LPF)が採用されている。
- ・誤り修正(Forward error correction: FEC)：以降の大半の規格では、外部で規定される。

3.2 MPEG-1 Video

MPEG-1 Video (ISO/IEC 11172-2)は、ISO/IEC JTC1により、1992年に策定された規格である。用途は、蓄積メディアへの保存である。1.5Mbps(当時の主流メディアであるCD-ROMの読出しレートに相当)以下のビットレートで、25Hzもしくは30HzのSIF画像(25Hzの場合は352×288, 30Hzの場合は352×240)を高効率に圧縮することを想定している。VideoCDやインターネット映像配信等で製品・実用化されている。

MPEG-1 Videoで新たに導入された技術は、以下の通り。

- ・両方向インタ予測：圧縮率向上に大きく貢献する。
- ・Group of Picture (GOP) 構造：蓄積用途に必須な機能であるランダムアクセス(ビットストリーム途中からの再生)の実現に不可欠である。
- ・1/2画素精度のMV
- ・量子化マトリクス：周波数成分毎に量子化値の調整が可能になる。主観画質の制御に用いられる。
- ・直交変換DC係数の予測符号化

3.3 MPEG-2 Video

MPEG-2 Video (ISO/IEC 13818-2)は、ISO/IEC JTC1により、1995年に策定された規格である。ITU-TでもH.262として規格化されている。用途は、蓄積、通信、放送と広く、高い汎用性を持った規格である。日本のデジタルテレビ放送や蓄積メディア(DVDおよびBlu-ray)等の分野で、現在も広く使用されている。MPEG-2 Videoは、インタレース形式の標準解像度テレビ(SDTV)および高精細テレビ(HDTV)をサポートする。

MPEG-2 Videoで新たに導入された技術は、以下の通り。

- ・フィールド・フレーム適応符号化:インタレース動画画像の各フィールド(トップおよびボトム)を個別のピクチャとして符号化するか、もしくは一つのフレームとしてまとめて符号化するか、柔軟に選択できる。
- ・インタレース対応の動き補償：フレームピクチャのMBは、フィールド毎に個別のMVを指定可能である。
- ・インタレース動画画像に適した直交変換、および直交係数のスキャン順序
- ・2通り(線形、非線形)の量子化スケール
- ・イントラ符号化時のDC成分の量子化精度向上
- ・可変長符号化テーブルの複数化
- ・4:2:2色差フォーマットのサポート：素材伝送等のプロフェッショナル用途向け。

またMPEG-2 Videoは、階層符号化に初めて対応した。



階層符号化は、複数の階層（例えば下位がCIF/15Hz/1Mbpsで、上位がSDTV/30Hz/4Mbps）のビットストリームをまとめて効率的に符号化する。上位ビットストリームは、下位ビットストリームとの差分となり、各階層を個別に符号化する場合（サイマルキャスト）に比べ、トータルのデータ量が削減可能である。階層化の方向は、時間方向（フレームレート）、空間方向（解像度）、品質方向（量子化値）、の3軸がサポートされている。また、時間方向階層化の応用として、3Dステレオ動画の符号化をサポートしている。階層符号化は、伝送路のビットレートや誤り率が変動する状況に合わせて最良の画質を実現するために有効であるが、MPEG-2 Videoの階層符号化の実用例は僅かである。

MPEG-2 Videoでは、多くのプロファイルおよびレベルが定義されている。例えば、日本のデジタル放送で使用されているのは、MP@HL (Main プロファイル High レベル、両方向インタ予測あり、階層符号化なし、HDTV (1,920 × 1,080/30Hz/4:2:0) 対応) である。

3.4 H.263

H.263は、ITU-Tにより1996年3月に基本規格が策定された規格である。用途は、超低ビットレートの実時間伝送である。16CIF (1,408 × 1,152) までの解像度、プログレッシブ形式、4:2:0色差フォーマットに対応し、携帯電話上のテレビ電話やコンテンツ配信等で製品・実用化されている。

H.263の基本規格では、H.261と比較して、以下の符号化ツールが追加されている。

- ・1/2画素精度のMV：MPEG-1 Video等と同じ
- ・3次元可変長符号化

H.263の拡張規格では、多くの符号化ツールが追加導入されている。H.263以降の標準規格にも採用されている符号化ツールを以下に示す。なお、これらの符号化ツールの使用は、H.263外部の通信プロトコル規定に基づく、送受信間の能力ネゴシエーションによって決定される。

- ・画面外領域を示すMVの許容
- ・算術符号化：MPEG-4 AVCのCABACの原型
- ・8 × 8単位動き補償、およびオーバーラップ動き補償
- ・イントラ予測：直交係数DC/AC成分の予測符号化
- ・デブロッキングフィルタ

3.5 MPEG-4 Visual

MPEG-4 Visual (ISO/IEC 14496-2) は、ISO/IEC JTC1により、1999年に初版が策定された規格である。

標準化開始時の目的は、蓄積・通信時の圧縮率を大幅に向上することであった。当時注目を集めたオブジェクト符号化（顔等の被写体をモデル化し、少ないパラメータで画像を記述する）により、自然動画を数kbpsで符号化する、といった野心的なプロジェクトであった。

しかしながら、オブジェクト符号化の実用化は難しく、結局は、自然動画の符号化方式はH.263をベースとした

ものとなった。なお、自然動画用とは別に、CG画像向けに、顔オブジェクト記述および人体オブジェクト記述、2D/3Dメッシュ記述といった符号化ツールも規定されている。本規格の名称が"Video"でなく"Visual"とされているのは、自然動画およびCG動画を、個々の映像オブジェクトとして扱い、複数オブジェクトを合成して最終的な動画とするフレームワークの採用による。

自然動画向けの符号化ツールとして、以下のものが新たに導入されている（※を付けた符号化ツール以外は、以降の標準規格では採用されていない）。

- ・VOP (Video Object Plane)：映像オブジェクトの記述フレームワーク。オブジェクト形状の符号化を含む。
- ・オーバーラップ動き補償：H.263と同様
- ・スプライト符号化：全景を除いた背景全体を予め伝送しておき、個々のピクチャにて背景に前景を合成する。
- ・8ビット以上の画素精度*
- ・データパーティション、および逆方向からも復号可能な可変長符号化：誤り耐性向上に用いる
- ・ウェーブレット変換：静止画像符号化時のみ
- ・グローバルMV (GMV)：画面全体の動きの記述
- ・動的解像度変換：動き補償誤差信号の解像度を可変にして、圧縮率を向上する。
- ・4:2:2/4:4:4色差フォーマット対応*：スタジオ素材の蓄積・伝送向け
- ・1/4画素精度MV*
- ・階層符号化：品質を細かに制御可能なFine Granularity Scalability (FGS)

MPEG-4 Visualでは、自然動画およびCG動画向けに多くのプロファイルレベルが定義されているが、製品・実用化されているのは、自然動画向けのいくつかのプロファイル (Advanced simple等) に限られる。

3.6 AVC

AVC (Advanced Video Coding) は、ITU-TとISO/IEC JTC1との共同で作成された標準規格である。当初ITU-Tが、H.263の後継規格として検討を進めていたものである。正式名称は、ITU-T側ではH.264、ISO/IEC JTC1側ではISO/IEC 14496-10 (MPEG-4 AVC) である。2003年に初版が作成され、2005年に、現在広く使用されているHighプロファイルの拡張規格が作成された。

AVCは、幅広い画像フォーマットをサポートする。

- ・解像度：QCIFから4K (4,096 × 2,180) まで
- ・フレームレート：60Hzプログレッシブまで
- ・色差フォーマット：4:2:0、4:2:2、4:4:4に加えRGB等の色空間もサポート

AVCはMPEG-4 Visualと同様に、汎用的な低ビットレート符号化を目的とするが、既存方式と同じく、動き補償および直交変換という基本方式を継承している。

AVCで新しく採用された符号化ツールは以下の通り。



- ・整数精度直交変換：4×4および8×8 (High プロファイルで導入) の二つのサイズを持つ。既存規格は浮動小数点精度のため、復号装置の実装形式によって復号結果に±1の変動が生じていたが、AVCでは復号結果が完全に一致し、また実装コストも軽減している。
- ・動き補償ブロックサイズ：最大16×16、最小4×4ブロック単位での動き補償が可能となり、インタ予測の効率が向上している。
- ・イントラ予測：画素領域での予測が初めて導入された。9個の予測モード(予測方向)を持つ。イントラ予測の効率が大幅に向上している。
- ・エントロピー符号化：可変長符号化(VLC)の拡張であるCAVLC(コンテキスト適応VLC)に加え、CABAC(コンテキスト適応二値算術符号化)が採用されている。CABACは圧縮効率が10%程度高く、CALVCはスループット(クロック当たりの処理ビット数)および実装簡易性の面で勝る。用途に応じて両者を選択可能である。
- ・フィールド・フレーム適応符号化：ピクチャレベルの適応選択(PAFF)に加え、MB単位での適応選択(MBAFF)がサポートされた。
- ・デブロッキングフィルタ：基本規格から採用
- ・MVの予測符号化：隣接3ブロックのMVの中間値を予測値とする。
- ・ロスレス符号化：量子化をスキップする。その他、MB単位で画素値をそのまま伝送するPCMモードを持つ。
- ・参照ピクチャ：既存規格では、前方向参照ピクチャと後方向参照ピクチャは、GOP構造によって一意であった。AVCではインタ予測の効率向上のため、明示的に

参照ピクチャを指定可能にする仕組みが導入された。長時間過去のピクチャや、両方向予測ピクチャも参照可能になっている。

- ・重み付き動き補償：フェードシーン等での輝度値変動を考慮して、動き補償結果を補正することで、インタ予測効率を向上する。
- ・階層符号化：既存規格に比べ、符号化効率および汎用性が向上している。
- ・多視点動画画像符号化：インタ予測を拡張した視点間予測を導入し、最大1024視点の圧縮に対応する。

AVCは、日本のワンセグ放送、外国のデジタル放送、Blu-ray、インターネット映像配信(YouTube等)といった、幅広い分野で使用されている。

3.7 VC-1

VC-1(SMPTE 421M)は、マイクロソフト社のWindows Media Video 9をベースに、SMPTEが2006年に規格化したものである。VC-1の用途は、低ビットレートでのネットワーク伝送等であり、サポートされる動画画像フォーマットは、4:2:0色差、8ビット画素精度に限定される。インタレースおよびプログレッシブの両方、最大HDTV 60Hzの解像度・フレームレートに対応する。

VC-1の符号化ツールは、MPEG-4 Visualの自然動画画像符号化ツールに近い。ただし、スプライト符号化や、グローバルMCは含まない。他の規格にはないユニークな符号化ツールは以下の通り。

- ・画素値レンジ圧縮：入力8ビット動画画像のレンジを最大7ビットまで圧縮して符号化する。
- ・直交変換：整数画素であり、かつ8×8、8×4、4×8、

表1 各標準規格の比較

標準規格	H.261	MPEG-1	MPEG-2/H.262	H.263	MPEG-4 Visual	AVC(MPEG-4 AVC/H.264)	VC-1
標準化時期	1990	1992	1995	1996	1999	2003	2006
最大解像度	CIF	CIF	HDTV	16CIF	HDTV	4K	HDTV
色空間	4:2:0	←	4:2:0/4:2:2	4:2:0	4:2:0~4:4:4	4:2:0~4:4:4, RGB	4:2:0
画素精度	8ビット	←	←	←	8~10ビット	8~12ビット	8ビット
予測方向	片方向	両方向	←	←	←	←	←
MV精度	1	1/2	1/2	1/4	1/4	1/4	1/4
ME	16×16	←	16×16 8×16/16×8 Dual Prime	4×4(OBMC) 16×16 スプライト, GMV	16×16 8×16/16×8	4×4~16×16	8×8~16×16
イントラ予測	No	DCT係数DC成分	←	DCT係数DC/AC成分	←	画素領域9方向	DCT係数DC/AC成分
Entropy	VLC	←	←	VLC 算術符号化	3D VLC	CAVLC CABAC	VLC
ループ内フィルタ	Yes 動き補償用	No	No	Yes デブロッキング	No	Yes デブロッキング	←
MB size	16×16	←	←	←	←	←	←
DCT	8×8 浮動小数点	←	←	←	←	4×4/8×8 整数	4×4~8×8 整数
Scalability	No		Yes	←	←	←	No
Multi-view	No	No	Stereo	No	No	Yes	No





4×4の4種類をサポートする。

- ・ビットプレーン符号化：符号化モード等，MB単位の二値情報を，ピクチャ単位でまとめて可逆圧縮する。
- ・動き補償：LPF係数をBi-cubicおよびBilinearの二つに切替え可能にする。

VC-1は，Blu-rayにも採用されている。

4. 各標準規格の比較

4.1 対象動画フォーマットおよび符号化ツール

H.261からVC-1までの各標準規格が対応する動画フォーマット，および使用する符号化ツールを表1にまとめる。

4.2 圧縮性能

各標準規格の用途や対応動画フォーマットが異なるため，総括した比較は困難であるが，個々の比較では以下のようになっている。

- ・プログレッシブ動画の場合，MPEG-1とMPEG-2とは同等の圧縮性能
- ・MPEG-2を1とした場合のH.263およびMPEG-4 Visual, AVCの圧縮率はそれぞれ，2倍弱，2倍強である¹⁾。

5. むすび

本稿では，主要な国際標準化機関によって標準化された動画符号化規格の概要を解説した。詳細については，各標準化団体が発行する規格文書を参照されたい。以下のURLから，有償もしくは無償で入手可能である。

ISO/IEC: <http://www.iso.org/iso/store.htm>

ITU-T: <http://www.itu.int/en/ITU-T/publications/>

SMPTE: <http://store.smpte.org/>

(2013年2月28日受付)

〔文 献〕

- 1) J.R. Ohm et al. "Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC)", IEEE Transactions on Circuits and Systems for Video Technology, 22, 12 (Dec. 2012)



数井 君彦 1993年，東京工業大学修士課程修了。現在，(株)富士通研究所勤務。動画符号化方式の研究開発，およびMPEG等の国際標準化に参加。

2013年7月号予定目次

【ふぉーかす】会長就任挨拶..... 理科大(新会長) 伊東 晋

【特別寄稿】テレビ放送開始60周年..... 総務省, NHK, 日本テレビ, シャープ

【特 集】「新しい画像符号化技術」

1. HEVC (MPEG-H/ITU-T H.26x) 技術解説

1-1 国際標準化作業..... Mitsubishi Electric Research Lab (USA) 杉本和夫

1-2 HEVCの階層ブロック構造および変換・量子化..... NEC 蝶野慶一

1-3 HEVC (High Efficiency Video Coding) の標準化動向..... ソニー 鈴木輝彦

1-4 可変長符号化..... JVCケンウッド 福島 茂

1-5 ループ内フィルタ..... 東 芝 中條 健

1-6 ビットストリームの構成とその機能..... パナソニック 西 孝啓

1-7 HEVCプロファイルレベルと拡張作業..... シャープ 山本智幸

2. JCT-3Vにおける3次元映像符号化方式の標準化動向..... NTT 志水信哉

3. SC29/WG1標準化動向..... 早 大 石川孝明・ソニー 福原隆治

【講 座】「ディスプレイの基礎」

(開講に当たって)

(第1回) 液晶ディスプレイの基礎..... 長岡技科大 木村宗弘

【映像情報メディア年報2013シリーズ】

(第7回) 無線・光伝送技術の研究開発動向..... NHK 田中祥次・池田哲臣, 東 芝 古川雄一, 千葉大 齊藤一幸, NEC 福本正義, 住友電工 太田順一

【知っておきたいキーワード】

(第90回) ソーシャルキュレーション..... パナソニック 近藤敏志

【メディアウォッチ】

(第17回) セコム株式会社IS研究所..... 電通大 早房長敏・竹内尚也・百瀬元気・宮鍋克麻・塩飽 淳

