動画圧縮/伸張のソフト vs. ハード

――アルゴリズムをくふうすれば MPEG-4でも専用ハードは不要

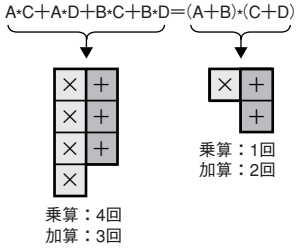
田中正文

音声を含む動画像データの圧縮規格であるMPEG-4を採用したシステムでは，DCT（離散コサイン変換）演算や動き検出などの負荷が大きいため，これらの部分を専用のハードウェアで実装することが少なくない．ここでは，MPEG-4ソフトウェアのCPU負荷を従来の1/3に低減できるアルゴリズムについて解説する．このアルゴリズムを用いると，アクセラレータなどの専用ハードウェアがなくてもソフトウェアのみでQCIF（176×144），15フレーム/sの性能を実現できる．　（編集部）

MPEG-4は，動画像の標準符号化方式の一つです．低ビット・レート（データ量の少ない符号化）がうたい文句であり，MPEG-1，MPEG-2と比べてデータ圧縮のためのさまざまなくふうがなされています．そのおかげでMPEG-4の符号化/復号化の処理量は大幅に増えており，ソフトウェアのみでの実現は困難と言われていました．またFPGAなどのハードウェアで実現する場合でも，100MHz以上の動作周波数が必要でした．

しかし，筆者ら（テクノ マセマティカル）が開発した「DMNA（Digital Media New Algorithm）」と呼ばれる新しいアルゴリズムをMPEG-4に応用すると，ソフトウェアで実装した場合に画像サイズがQCIF（176ピクセル×144行）で15フレーム/sの性能を実現できます．また，FPGAで実装した場合には，画像サイズがCIF（352ピクセル×288行），フレーム速度が15フレーム/s，27MHz動作で，符号化（エンコード）と復号化（デコード）の同時並行処理を実現することができます．

本稿では，このDMNAを用いたアルゴリズムの概要とその実装例を解説します．



〔図1〕
因数分解法

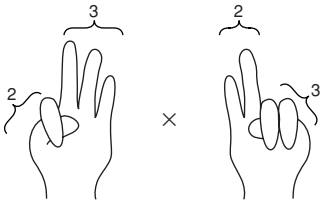
DMNAの一つの手法として因数分解による演算の削減が挙げられる．図の例では，数式どおりに計算すると乗算が4回，加算が3回必要であるのに対して，DMNAの因数分解を用いると乗算が1回，加算が2回ですむ．

1. 数式の変更や変換などにより演算回数を削減

信号処理にDMNAを応用すると，符号化/復号化の数式どおりに順を追って実行するのではなく，前処理として数式の変更や変換，組み合わせを行い，さらに演算実行順序を最適化します．これによって，実行する演算量を大幅に削減することができます．以下に，その具体的な手法を説明します．

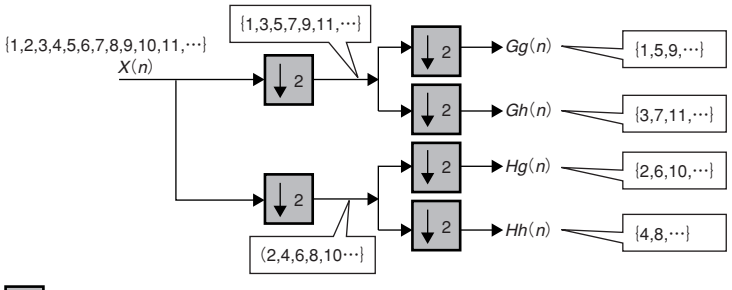
例えば演算を実行する前に，まず因数分解を行ってから演算を実行することによって演算量を削減できます．**図1**の例では，数式どおりに計算すると，乗算が4回，加算が3回必要です．しかし，因数分解を演算実行の前に行うと，加算が2回，乗算が1回ですみます．

第2の手法としては，折り返しによる演算の削減が挙げられます．折り返しとは，フランスのブドウ園で働いていた農夫たちが，片方の手で計算をじょうずに行っていたことに由来するため，別名“Grapes”と呼ばれています．では，自分の手を使って折り返しを確認しましょう．「1，2，3，…」と数えながら，親指から順に指を折っていってください．「5」ですべての指が折れ曲がりました．次に，「6，7，…」と折りまげた指を伸ばしてください．「10」ですべての指が伸びて元に戻りました．これが折り返しの基本です．**図2**を見てください．先ほどの折り返しによると，向かっ



〔図2〕折り返しの原理

向かって左側の手は数字の‘8’を，右側の手は数字の‘7’を表している．ここで8×7を実行する．左右の立っている指の数を数えてみると3＋2＝5になり，これを10の位とすると50になる．次に，左右の折れ曲がっている指の数を掛けてみると2×3＝6で，これを1の位とすると，10の位と合わせて56になる．



〔図3〕優先順位付き階層構成

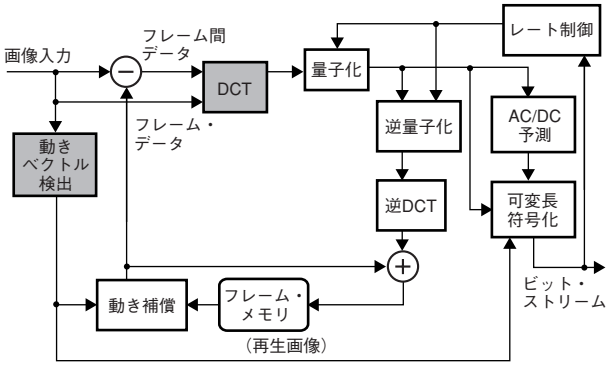
優先順位が高い$Gg(n)$だけ実行すればよいアプリケーションの場合，DMNAでは必要な部分だけを演算するため高速に処理できる．

て左側の手は数字の‘8’を表しています．また，向かって右側の手は数字の‘7’を表しています．

では，8×7を実行してみましょう．ここで，左側の指と右側の指について立っている指の数を数えてみると，3＋2＝5です．これを10の位とすると50になります．次に，左右の折れ曲がっている指の数を掛けてみると，2×3＝6です．これを1の位とすると，このままなので6になります．先ほどの10の位と合わせると56になります．すなわち，8×7＝56が折り返しを使うことで，すべて5以下の数字で実行できました．1回の折り返しによって，5までの数値空間で10進数の演算を行えることが，この例からわかります．

第3の手法は階層構成です．複雑な計算を階層的に変形して，簡単な領域に持ち込んでから計算を実行する方法です．また，この階層構成では優先度をつけて重要な計算から順に実行できます．そのため，動画像処理のような高速リアルタイム演算が必要な用途には，非常に有効な手法と言えます．

図3は，入力信号$X(n)$をサブサンプリング（間引き）の方法で高周波領域，低周波領域に分割しています．画像処理の場合，低周波領域のデータが重要になります．これを重要度順に$Gg(n)$，$Gh(n)$，$Hg(n)$，$Hh(n)$とすると，応用分野によってはもっとも重要な$Gg(n)$部分の計算だけで用が足りることがあります（人間の目がもっとも反応するのは低周波領域．高画質になるほど高周波領域も必要になる）．



〔図4〕MPEG-4の符号化処理

DCT演算と動き検出（ME）は負荷が重いため，別チップ（ハードウェア）で実装されることが多い．

2. MPEG-4のDCT演算と動き検出の負荷を軽減

次に，DMNAの応用例として，MPEG-4の中でも重要な演算であるDCT（離散コサイン変換）演算と動き検出（ME：motion estimation）について説明します（**図4**）．MPEG-4には多くのプロファイル（要求機能）が用意されていますが，ここでは話を簡単にするため，simple profile（MPEG-4のもっとも基本的，かつ一般的な機能を規定している）に限定して説明します．

●因数分解を利用してDCTの演算を簡略化する

DCT演算の高速化や計算の簡易化は，過去数十年にわたってもっとも議論，検討，考察がなされてきた部分ですが，意外と決定打がないというのが実情です．

MPEG-4では，2次元のDCT演算を実行することになっていますが，2次元のDCT演算を一度に実行しているわけではありません．まず，行方向に1次元のDCT演算を実行