

HD/UD TV符号化用MEアーキテクチャ

佐藤 真樹 花村 剛 大山 公一 森田 一彦 山田 恭裕

株式会社 グラフィックス・コミュニケーション・ラボラトリーズ

HD/UD(High Definition/Ultra Definition)画像符号化チップ開発では、画像サイズが大きいため、動き推定(Motion Estimation, ME)の処理量及びメモリアクセス量の削減は重要課題となる。本稿では、MEの処理量削減を目的とした、フレーム探索簡略化階層型MEアーキテクチャを提案する。本アーキテクチャでは、4分木階層型MEと共に、インタレース符号化におけるフレーム/フィールド適応予測に必要なベクトルを独立で探索するのではなく、先に求めたフィールドベクトルを候補としてフレームベクトルを求めるという簡略化により処理量の削減を実現している。更に、階層型MEに対して幾つかの制限を加えることで、画質劣化を視覚的許容範囲に抑えながら、メモリアクセス量削減を実現する手法を提案し、シミュレーション実験により、その有効性を確認した。

Motion Estimation Architecture for HD/UD TV Coding

Masaki Sato Tsuyoshi Hanamura Koichi Oyama Kazuhiko Morita Yasuhiro Yamada

Graphics Communication Laboratories

6F ANNEX Tohshin BLDG., 4-36-19, Yoyogi, Shibuya-ku, Tokyo, 151, Japan

In the case of the development of a HD/UD(High Definition/Ultra Definition) TV coding chip, It is very important to reduce the amount of operation and memory access for a motion estimation(ME), since its image size is very large. This paper presents the ME architecture for HD/UD TV coding. This architecture uses the hierarchical motion estimation technique including the simplification of the estimation for frame vectors to reduce the amount of operation, and restricts some ME parameters to reduce the amount of memory access. This architecture is discussed and the simulation results show the usefulness of the proposed methods.

1. はじめに

MPEG[1][2]に代表される動き補償と予測誤差符号化を含むハイブリッド動画像符号化アルゴリズムの場合、その処理の大半が動き推定(ME)に費やされている。特にHD/UD(High Definition/Ultra Definition)画像の場合、画像サイズが大きく、MEの処理量及び、メモリアクセス量は膨大なものとなる。従って、HD/UD動画像符号化チップ開発では、MEの処理量及びメモリアクセス量の大幅な削減は重要課題となる。

本稿では、特にMPEG2[2]アルゴリズムのMEの処理量目的としたフレーム探索簡略化階層型MEアーキテクチャ及び、階層型MEのメモリアクセス量削減手法を提案する。本手法の概略を以下に示す。

1. 入力画像を4分木型に階層化し、各階層でのマッチング画素数を変えながら、低解像度層から順に動ベクトルを求め、これを次の階層での探索初期ベクトルとする。その結果、各階層での探索範囲が小さくて済み、処理量が飛躍的に削減される。
2. インタレース符号化におけるフレーム/フィールド適応予測に必要なベクトルを独立で探索するのではなく、先に求めたフィールドベクトルを候補としてフレームベクトルを求めるといった簡略化により更なる処理量の削減を実現する。
3. 階層型MEに対して幾つかの制限を加えることで、画質劣化を視覚許容範囲に抑えながらメモリアクセス量削減を実現する。

2. 階層型MEアーキテクチャ

本稿では、文献[3]の4分木階層画像による階層型MEに基づき、各階層でのマッチング画素数を変化させながら動き量の分解能を順次上げる手法と、インタレース符号化におけるフィールド/フレームベクトル導出手法として、文献[4]に示された簡略化の組合せによるMEアーキテクチャを提案する。

2.1 基本アルゴリズム

1) 4分木階層画像の生成[3]

まず、下記、漸化式(1)に基づき、予測対象画像、参照画像のそれぞれに対し4分木階層画像を生成する。但し、 $f_{i,0}(x,y)$ を原画像とする。

$$f_{i,h+1}(x,y) = \frac{1}{4} \sum_{i=0}^3 \sum_{j=0}^3 f_{i,h}(2x+i, 2y+j) \quad (1)$$

以下、4分木階層画像の第h階層画像 $f_{i,h}(x,y)$ に対し、そのひとつ上の第h+1階層画像を $f_{i,h+1}(x,y)$ と定める。

4分木階層画像では、上位層(低解像度層)では物体の動きを大局的に把握でき、雑音の影響も少ない。一方、下位層(高解像度層)では、物体の局所的な動きを画素解像度に近いレベルで把握できる。

2) 動き量検出手法[3]

4分木階層画像の最上位階層から最下位階層に向けて、各階層毎に絶対値差分評価ブロックマッチング法を用いて動き量を検出する。ここで、参照画像における階層hの画像を $f_{i,h}(x,y)$ 、予測対象画像における階層hの画像を $f_{i,h}(x,y)$ とする。第h階層におけるマッチングの探索は、一つ上の第h+1階層の被従属ブロック(親ブロック)の検出結果を初期偏位として、式(2)の評価関数を最小とする V_h' を求めることになる。

$$S(u_h', v_h') = \sum_x \sum_y \left| \begin{array}{c} f_{i,2h}(x - u_h' - 2u_{h+1}, y - v_h' - 2v_{h+1}) \\ - f_{i,h}(x, y) \end{array} \right| \quad (2)$$

初期偏位点: $2V_{h+1} = (2u_{h+1}, 2v_{h+1})$

探索点: $V_h' = (u_h', v_h')$

$$-p_{x_h} \leq u_h' \leq p_{x_h}, -p_{y_h} \leq v_h' \leq p_{y_h}$$

このとき階層hの検出動き量は $V_h = V_h' + 2V_{h+1}$ で表せる。従って、階層数をhmaxとすると、最終的に得られる動ベクトル V_{result} は式(3)で示される。

$$V_{\text{result}} = V_0 = V_{h_0}' + 2V_1 = \sum_{i=0}^{h_{\text{max}}-1} 2^i V_i', (V_{h_{\text{max}}} = 0) \quad (3)$$

3) マッチング画素数と近傍参照

文献[3]では、全ての階層でマッチング画素数を同一としているが、本アーキテクチャでは、図1に示すように、ある階層 h_n より上位の階層でマッチング画素数を同一とし、 h_n より下位層では、ブロックの画面全体に対する割合を一定とする。従って、上位層では、原画像における検出ブロックを幾つかまとめてサーチすることになり、最終的に得られる動ベクトル場が、空間的な連続性を持ちやすく、動ベクトルエントロピーが小さくなる。さらに、探索範囲をまとめて読み出せるので、メモリアクセス量も若干削減できる。また、上位層でプ

ブロックサイズが大きくなりすぎるという問題を解決している。

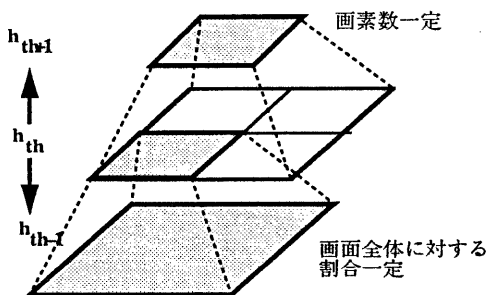


図1 各階層でのマッチング画素数

本来異なる動きを有する検出ブロックを幾つかまとめて親ブロックとし、そのブロックでサーチを行った場合、親ブロックでの動き検出ミスマッチは、次の階層では大きな誤差を持つことになる。このときの誤検出偏位が、下位層での探索範囲(本アーキテクチャでは $\pm 1\text{pel}$)を超えた場合、正確な動き量を検出することが不可能となり、最終的な予測誤差は大きく増大する。この問題は物体の境界領域で顕著となる。そこで本手法では、同一階層で既に探索の終了した近傍ブロックの検出結果を探索点に加える(図2参照)。これにより、親ブロックの検出結果とは独立して、近傍ブロックの検出結果を探索点として参照できるので、周辺領域に対する相対的な動きが小さければ、最終的な予測誤差の増大を防ぐことができる。

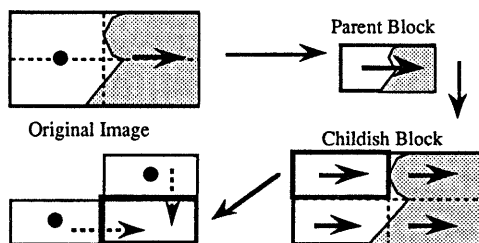


図2 近傍参照による誤検出の解消

4) 半画素精度探索

本アーキテクチャでは、半画素精度の探索は、階層探索の結果得られた整数画素精度のベクトルを用いて、その回り8点の半画素位置を改めて探索

することで実現している。

2.2 フレームベクトル探索の簡略化[4]

インタレース符号化におけるフィールド/フレームベクトル導出手法としては、各ブロックの両ベクトルを独立に探索することが一般的である。しかし、同一ブロック内の両ベクトルはほぼ同じ動きを示し、また近傍のブロックでの動ベクトルも考慮に入れ、これらを相互に参照して動ベクトルとすることで、動き補償予測するに十分な動ベクトルが得られると予想される。

本アーキテクチャでは、インタレース構造に適したフィールドベクトル4種類(予測対象フィールド2種類、参照フィールド2種類の組合せ)を、階層型MEで探索した後、その結果得られたベクトルのうち、同一位置ブロックを含む近傍ブロックの同パリティベクトルを候補ベクトルとして、フレームベクトルを探索するというフレームベクトル探索簡略化手法を用いた。半画素の動き探索は、x方向は候補ベクトルのx成分をそのまま使い、y方向は候補ベクトルのy成分をフレーム画像に換算した整数画素で探索後、 ± 0.5 (画素)位置を再探索する。

2.3. 性能評価

画像サイズ1920*1088,原画層でのブロックサイズ16*16,探索範囲をx,y共参照・予測対象画像間距離当たり $\pm 31.5[\text{pel}/\text{frame}]$ としたときの、階層型MEの処理量(付録1参照)を表1に示す。従来手法として、同一条件の全探索及び、動きの時間方向の連続性を利用して画像間距離により探索範囲を制御する探索手法(テレスコピックサーチ[1])の処理量も併せて表1に示す。表1から、本提案手法の階層型MEは、処理量が大幅に削減されていることがわかる。フレーム間距離が大きい場合でも、処理量の増加は僅かであり、HD画像に十分対応できる。

次に、テレスコピックサーチと階層型MEを用いて符号化した150フレームの平均SNRを表2に示す。今回は、画像品質とシミュレーション時間の問題から601画像を用いた。

SNRにおいても、テレスコピックサーチに比較して、ほぼ同等値を示しており、動き補償予測するに十分なベクトルが得られていることがわかる。

表1 ME処理量

ME処理量 [M・operations / frame]	Frame Distance	
	1	4
階層型ME	349	407
全探索型ME	33033	542405
Telescopic	33033	33033

表2 平均SNR[dB] (4Mbps, M=3, N=15)

SNR[dB] 4Mbps	Bus			Flower		
	Y	Cb	Cr	Y	Cb	Cr
Telescopic	31.64	38.63	40.78	29.77	33.25	35.01
階層型ME	31.67	38.67	40.82	29.82	33.27	35.04

3. メモリアクセス削減によるHD対応

HDのように画素数が大きい場合、階層型MEのチップ化では、メモリアクセス量が問題となる。以下に、1画素当たりのメモリアクセスを1として、本アーキテクチャの一画面当たりの予測対象画像の読み込みメモリアクセス量を示す。(付録2参照)

画像サイズ: 1920*1088pel

原画層でのブロックサイズ: 16*16pel (frame)

探索範囲: x,y共±127.5pel(半画素探索を含む)

Field: 階層型ME ($h_{max}=4, h_{in}=1$, 最上位層以外の探索範囲±1, 近傍参照ブロック数=2)

3層: 0.2M, 2層: 0.9M, 1層: 3.6M, 0層: 14.2M
half-pel: 5.9M

Frame: 簡略型 候補ベクトル数=(4近傍+同位置)*
同パリティベクトル=5*2=10本

: 25.0M

Total: 49.8M

(全探索=35.5M, テレスコピック=10.4M)

以上の計算結果から分かるようにフィールドベクトルの原画層(0層)とフレームベクトルの処理のメモリアクセス量が多いことがわかる。次節からこれらの処理を制限することでメモリアクセス量を削減する手法と、その制限による画質への影響について述べる。

3.1 フレーム用候補ベクトルの限定

前節では候補ベクトルとして、同一位置ブロックと4近傍ブロックの同パリティフィールドベクトルを用いている。しかし、フィールドベクトル探索時に既に近傍参照を行っていることから、4近傍

ブロックの候補ベクトル探索を省略する。この簡略化によりフレームベクトル探索時のメモリアクセス量は1/5となる。

3.2 原画層での近傍参照の限定

本階層型MEでは、最上位層を除く各階層で近傍2ブロックのベクトルを参照している。原画層(0層)の場合、ブロックの画素数及び、画面当たりのブロック数が共に大きく、メモリバンド幅上、大きな負担となる。ここで、近傍参照の目的を考えると、2.1節で述べたように、動きの境界領域で、ブロックをまとめてサーチした場合に生じる誤検出を解消することにある。例えば、 $h_{in}=1$ の場合、原画層(0層)のブロックと1層のブロックの画面全体に対する割合が等しい。従って、前述の動きの境界領域での誤検出は、1層における近傍参照で解消されている可能性が高い。(図3参照)

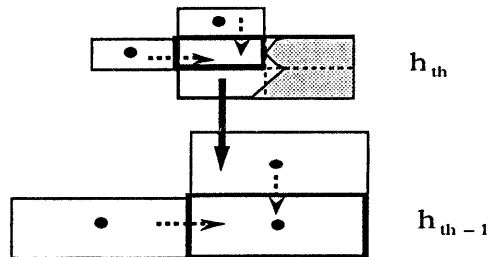


図3 h_{in} 未満の階層での近傍参照

またMPEGでは、予測可能なピクチャタイプとして、片方向予測のみのP-ピクチャと両方向予測を許すB-ピクチャの2種類がある。P-ピクチャは、他のピクチャの予測に用いられるため、B-ピクチャに比べ高画質が要求される。またB-ピクチャの場合、両方向予測のためメモリアクセス量は前述の計算結果の2倍となる。

以上の理由から、B-ピクチャのみ原画層での近傍参照を禁止する。原画層での近傍参照禁止による予測誤差の増加は、0層-1層間の高周波の損失分に起因し、低周波成分のマッチングは既に近傍参照も含めて終了しているため、ブロック歪みのような局所的劣化ではないことが期待できる。また、P-ピクチャでは制限を設けていないため、誤差の伝搬による他のピクチャの予測への影響はないと考えられる。この簡略化により、B-ピクチャの原画層

におけるメモリアクセス量は約0.4倍となる。

3.3 近傍参照ブロック位置

3.1,3.2節で述べたような、メモリアクセス量削減のための近傍参照の限定を行った場合、上位の階層でより効率的な近傍参照が必要となる。近傍参照ブロック数を2とした場合、参照位置パターンは $8*7/2=28$ 通りとなる。実際には既に処理の終了したブロックに限定され、またMPEGの場合、処理順直前のブロックの検出結果との差分符号化を考慮に入れなくてはならないため、直前のブロックの結果を常に参照する。従って、左上から順に処理していくとすると、近傍参照ブロック位置は、1)左、左上、2)左、上、3)左、右上の3種となる。図4にそれぞれの場合における検出結果の移入可能範囲を示す。参照効率を入力画像に依存するので、一概にはいえないが、3)がより広範囲のブロックからの移入が可能であり、特に下位層で近傍参照を制限した場合、より効率的であると考えられる。

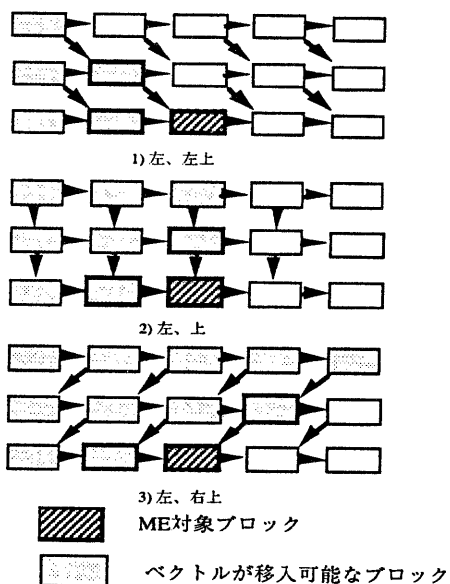


図4 近傍参照による検出結果の移入

3.4 メモリアクセス量削減による画質への影響

前述の制限手法による動き探索を行い、その結果を用いて符号化したときの150フレームの平均SNRを表3に示す。結果は、輝度成分のSNRで、フレ

ーム候補ベクトルを同一ブロックに限定した場合の劣化が0.04~0.07dB、B-ピクチャの原画層における近傍参照の禁止で0.09~0.1dB、2つを組み合わせることにより、0.16~0.25dBの劣化となった。近傍参照の効率を向上させるため、参照位置を変更した場合(左、上-->左、右上)、若干向上がみられ0.13~0.2dBの劣化に留まった。

一般に、MEに起因する画質劣化はブロック歪み等局所的な劣化が生じることが多いが、本提案手法では復号画像の局所的劣化は見出せなかった。これは、3.2節で述べたように、低解像度層には制限を加えていないため、低周波成分のマッチングが十分取れていることに起因すると考えられる。

Field: 階層型ME ($h_{max}=3, h_{min}=1$, 最上位層以外の探索範囲 ± 1 , 左と上ブロック参照)

Frame: 簡略型

表3 制限手法の平均SNR[dB] (4Mbps, M=3, N=15)

SNR[dB]	Bus			Flower		
	Y	Cb	Cr	Y	Cb	Cr
4Mbps						
制限無し	31.67	38.67	40.82	29.82	33.27	35.04
1) Frame候補ベクトル限定	31.60	38.64	40.76	29.78	33.26	35.02
2) B-pictureの原画層での近傍参照禁止	31.58	38.64	40.79	29.72	33.24	35.00
1)+2)	31.42	38.60	40.71	29.66	33.19	34.97
1)+2)+(左、上->左、右上)	31.47	38.62	40.75	29.69	33.22	34.99

4. むすび

HD/UD画像符号化チップ開発において重要課題である、動き推定処理量を大幅に削減したフレーム探索簡略型階層型MEアーキテクチャとHD/UD画像で問題となるメモリアクセス量を画質劣化を抑えて削減する手法を考案し、シミュレーション実験によりその有効性を確認した。テレスコピックサーチに比較して処理量で約1/80以下に、メモリアクセス量削減のための限定を加えても、SNRで約0.2dBの劣化に留めることができた。

今後の課題としては、処理量、メモリアクセス量の更なる削減の検討を続けると共に、様々な画像に対応できる高精度化の検討を行う予定である。

付録1

・階層型MEの処理量計算

式(2)の評価関数式から、1本のフィールドベクトルを求める場合の、ある階層hの画面当たりの処理量(絶対値誤差計算+誤差積算計算)は以下ようになる。但し、積算計算回数は絶対値誤差計算回数と同数とする。

$$(階層画像の画素数) \times (探索範囲(h) + 近傍参照数) \times 2 \quad (4)$$

半画素探索では、8点の探索位置の評価関数計算の他に半画素補間の処理が必要となり、原画像の1画素当たり、2画素平均2点、4画素平均1点づつある。従って、処理量は以下の式となる。

$$(0層の画素数) \times 8 \times 2 + (0層の画素数) \times (1 \times 2 + 3) \quad (5)$$

また、4フィールド分の4分木階層画像生成のための処理量は、以下のように表せる。

$$M_{QT} = 3 \sum_{h=0}^{h_{max}-1} \frac{(0層の画素数)}{4^h} \quad (6)$$

簡略化フレームベクトル探索の処理量は(候補ベクトルのx成分が、全て半画素精度の場合)、評価関数計算が、y方向2点の半画素探索を含むため、以下の式となり、

$$(原画像の画素数) \times (候補ベクトル数) \times (2+1) \times 2 \quad (7)$$

半画素補間処理が、原画像の1画素当たり、2画素平均点、4画素平均点共に1点づつあるため、以下のようになる。

$$(原画像の画素数) \times (1+3) \quad (8)$$

・全探索型の処理量計算

$$\begin{aligned} \text{整数精度} &= (\text{原画像の画素数}) \times \\ & \quad (\text{探索範囲}(\text{画像間距離})) \times 2 \quad (9) \end{aligned}$$

半画素精度=式(5)と同じ(0層=フィールド原画層)

・テレスコピックサーチの処理量計算

全探索で探索範囲を画像間距離に関わらず一定とする(画像間距離=1のときの探索範囲を用いる)。

例) 画像サイズ 1920*1088
 ブロックサイズ 16*16(frame)/16*8(field)
 探索範囲±127.5pcl (frame distance=4)

1) 階層型ME

フィールドベクトル4本分

$$\begin{aligned} \text{3層: } & (1920/8) \times (1088/16) \times ((15 \times 2 + 1) \times (7 \times 2 + 1) + 0) \times 2 \times 4 \\ & = 60.71\text{Mops} \end{aligned}$$

$$\begin{aligned} \text{2層: } & (1920/4) \times (1088/8) \times ((1 \times 2 + 1) \times (1 \times 2 + 1) + 2) \times 2 \times 4 \\ & = 5.74\text{Mops} \end{aligned}$$

$$\begin{aligned} \text{1層: } & (1920/2) \times (1088/4) \times ((1 \times 2 + 1) \times (1 \times 2 + 1) + 2) \times 2 \times 4 \\ & = 22.98\text{Mops} \end{aligned}$$

$$\begin{aligned} \text{0層: } & (1920) \times (1088/2) \times ((1 \times 2 + 1) \times (1 \times 2 + 1) + 2) \times 2 \times 4 \\ & = 91.91\text{Mops} \end{aligned}$$

半画素探索: $(1920 \times 1088/2) \times (8 \times 2 + 3) \times 4 = 87.73\text{Mops}$

$$\text{階層作成: } M_{QT} = 3 \sum_{h=0}^{4-1} \frac{(1920 \times 1088/2)}{4^h} = 4.16\text{Mops}$$

フィールド計=273.25Mops

フレームベクトル1本分

$$(1920 \times 1088) \times 10 \times 3 \times 2 + (1920 \times 1088) \times 4 = 133.69\text{Mops}$$

フィールド+フレーム(total)=406.94Mops

2) 全探索

$$(1920 \times 1088/2) \times (127 \times 2 + 1) \times (63 \times 2 + 1) \times 2 \times 4 +$$

$$\begin{aligned} & (1920 \times 1088/2) \times (8 \times 2 + 3) \times 4 + \\ & (1920 \times 1088) \times (127 \times 2 + 1) \times (127 \times 2 + 1) \times 2 + \\ & (1920 \times 1088) \times (8 \times 2 + 3) = 542404.73\text{Mops} \end{aligned}$$

3) テレスコピック (探索範囲は距離=1の±31.5pclとなる)

$$\begin{aligned} & (1920 \times 1088/2) \times (31 \times 2 + 1) \times (15 \times 2 + 1) \times 2 \times 4 + \\ & (1920 \times 1088/2) \times (8 \times 2 + 3) \times 4 + \\ & (1920 \times 1088) \times (31 \times 2 + 1) \times (31 \times 2 + 1) \times 2 + \\ & (1920 \times 1088) \times (8 \times 2 + 3) = 33032.72\text{Mops} \end{aligned}$$

付録2

・階層型MEのメモリアクセス量

フレームでのブロックサイズをBx*Byとする。

1) 最上位層(h=hmax-1=3)

予測画像の探索領域のうち処理ブロック順による更新部分(8*14pcl)だけ読み込むとすると、以下のようになる。

$$(\text{更新部分}) \times (\text{ブロック数}) \times 4\text{field} \quad (10)$$

例) 3層: $(8 \times 14) \times (510) \times 4 = 0.2\text{M}$

2) h_h以上の階層

同じ親ブロックをもつ近接ブロック(4個)は、探索領域を一度にまとめて読みできるので、

$$(4 \text{ブロックの探索領域}) \times (\text{ブロック数}/4) \times 4\text{field} \quad (11)$$

となり、更に近傍参照分(2ブロック分)が加わる。

$$(\text{ブロックの画素数} \times 2) \times (\text{ブロック数}) \times 4\text{field} \quad (12)$$

例) 2層: $(18 \times 10) \times (2040/4) \times 4 + (8 \times 4 \times 2) \times (2040) \times 4 = 0.9\text{M}$

3) h_h未満の階層

各ブロックは全て異なる親ブロックを持つので以下のようになる。

$$\begin{aligned} & (1 \text{ブロックの探索領域}) \times (\text{ブロック数}) \times 4\text{field} + \\ & (\text{ブロックの画素数} \times 2) \times (\text{ブロック数}) \times 4\text{field} \quad (13) \end{aligned}$$

例) 0層: $(18 \times 10) \times (8160) \times 4 + (16 \times 8 \times 2) \times (8160) \times 4 = 14.2\text{M}$

4) 半画素探索

ブロックのxy方向に半画素補間に必要な2画素分大きく読み込む必要がある。

$$(Bx+2) \times (By/2+2) \times (\text{ブロック数}) \times 4\text{field} \quad (14)$$

例) HP: $(18 \times 10) \times (8160) \times 4 = 5.9\text{M}$

5) 簡略型フレーム探索

候補ベクトルのx成分が全て半画素単位であるとして、y方向の半画素探索を含めた半画素補間分が必要となる。

$$(Bx+1) \times (By+2) \times (\text{候補ベクトル数}) \times (\text{ブロック数}) \quad (15)$$

例) frame : $(17 \times 18) \times (10) \times (8160) = 25.0\text{M}$

・全探索型のメモリアクセス量

フレームとフィールドを半画素探索分を含めて同時に読み込むとすると、以下のようになる。

$$(\text{更新部分}) \times (\text{ブロック数}) \quad (16)$$

例) $(16 \times (128 \times 2 + 16)) \times (8160) = 35.5\text{M}$

・テレスコピックのメモリアクセス量

全探索で画像間距離=1のときの探索範囲を用いる

$$\text{例) } (16 \times (32 \times 2 + 16)) \times (8160) = 10.4\text{M}$$

参考文献

- [1] ISO/IEC, IS 11172-2(1993).
- [2] ISO/IEC, DIS 13818-2(1994).
- [3] 富永英義, 小松尚久, 宮下壮史, 花村剛: "階層画素情報を用いた動画像における動き量検出方式", 信学論, Vol.J 72-D-II No3, pp.395-403 (1989).
- [4] 花村剛, 大山公一, 山田恭裕: "フレーム/フィールド適応予測符号化における動ベクトル探索の簡略化", 信学秋季全大, D-157(1994).