

複数参照フレームを用いた効率的動き探索に関する検討

松尾 翔平[†] 永吉 功^{††,†††} 花村 剛^{††,†††} 富永 英義^{††,††††}

[†] 早稲田大学大学院 国際情報通信研究科 〒 367-0035 埼玉県本庄市西富田大久保山 1011

^{††} 早稲田大学 国際情報通信研究センター 〒 367-0035 埼玉県本庄市西富田大久保山 1011 93 号館

^{†††} 株式会社 メディアグルー 〒 169-0072 東京都新宿区大久保 2-4-12 新宿ラムダックスビル 8 階

^{††††} 早稲田大学 理工学部 コンピュータ・ネットワーク工学科 〒 169-8555 東京都新宿区大久保 3-4-1

E-mail: †{s-matsuo,isao,hana,tominaga}@tom.comm.waseda.ac.jp

あらまし H.264/MPEG-4 AVC は、高い符号化効率を実現するために、従来の標準動画符号化に加えて、様々な新規のツールが導入された。特に動き補償では、可変ブロックサイズの動き補償、1/4 画素精度の動き補償、そして、複数参照フレームによる動き補償、と 3 つのツールが取り込まれた。動き補償では、参照フレームの枚数が増加すると、それに比例して動き補償の処理が増大する。本稿では、複数参照フレーム間の動きベクトルの相関性と縮小画像のコストを考慮し、動き探索打ち切り処理を加えることで、探索枚数が増加しても、画質を保ちつつ探索の演算回数を参照枚数 1 枚の場合に近似する方式を検討する。既存の動き探索処理と容易に組み合わせを可能にしながら、符号化において甚大となる動き補償の処理量を大幅に削減する。

キーワード H.264/MPEG-4 AVC, 複数参照フレーム, 動き補償, 動きベクトル, 符号化処理量削減

A Study on Effective Motion Estimation Algorithm with Multiple Reference Frames

Shohei MATSUO[†], Isao NAGAYOSHI^{††,†††}, Tsuyoshi HANAMURA^{††,†††}, and Hideyoshi

TOMINAGA^{††,††††}

[†] Graduate School of Global Information and Telecommunication Studies, Waseda University, 1011 Ohkuboyama, Nishitomida, Honjou-shi, Saitama, 367-0035 Japan

^{††} Global Information and Telecommunication Institute, Waseda University, 93th building 1011 Ohkuboyama, Nishitomida, Houjou-shi, Saitama, 367-0035 Japan

^{†††} Media Glue Corp. Ramdax Bldg. 8th floor, 2-4-12 Ohkubo, Shinjuku-ku, Tokyo, 169-0072 Japan

^{††††} Department of Computer Science Engineering, Waseda University, 3-4-1 Ohkubo, Shinjuku-ku, Tokyo, 169-8555 Japan

E-mail: †{s-matsuo,isao,hana,tominaga}@tom.comm.waseda.ac.jp

Abstract We can utilize various new tools by H.264/MPEG-4 AVC as well as existing video coding methods in order to achieve a higher level of coding efficiency. For example, new technology in motion compensation tools is included in H.264/MPEG-4 AVC, such as variable block size motion compensation, multiple reference frames and quarter-pel motion compensation. The encoding cost increases in proportion to the number of reference frames. In this paper, we propose an effective motion estimation algorithm with multiple reference frames by using the correlativity of motion vector among reference frames and the cost of the reduced image. Simulation results show that the proposed algorithm reduces the encoding time to that of one reference frame while keeping image quality and bit rate.

Key words H.264/MPEG-4 AVC, Multiple Reference Frames, Motion Compensation, Motion Vector, Computational Complexity Reduction

1. はじめに

国際標準動画符号化方式 H.264/AVC(以下 H.264) において、高い圧縮効率を実現するために新たなツールが導入された。特に、動き補償においては、従来の標準に対して、可変ブロック動き補償、1/4 画素精度の動き補償、そして複数参照フレーム (MRF:Multiple Reference Frame) の動き補償、と3つのツールが新たに取り込まれた [1, 2]。本稿では、複数参照フレームの動き補償について議論する。

複数参照フレームの場合、参照フレームの枚数に比例して動き探索の処理が増加する。本稿では、複数参照フレーム間の動きベクトルの相関性、および、縮小画像のコスト評価に基づいて、参照フレームの絞込みを行い、動き補償の処理量削減を目的とする。本方式は、既存の動き探索処理と容易に組み込むことが可能であり、探索する参照フレーム数が増加しても、演算回数を参照枚数 1 枚の場合に近似することを目的とする。

2. 複数参照フレームによる動き補償

2.1 複数参照フレームの利点と問題点

MPEG-2, MPEG-4 などの従来の標準符号化方式では、直前のフレームのみを参照して動き補償を行う。これに対し、H.264 では複数の参照フレームを使用可能となった。概念図を図 1 に示す。複数参照フレームが導入された理由は、直前参照のみでは符号化効率が向上しない場合に対して、より前のフレームを参照し、その問題点を改善するためである。複数参照フレームの利用が効果的な場合として、以下の場合が挙げられる [3]。

- (1) 繰り返し (反復する) 動き
- (2) 物体で覆われていない背景
- (3) カメラの振動
- (4) 非整数画素の動き
- (5) 光や影の変化
- (6) カメラのノイズ

上記の場合に対しても、最適な動きベクトルを検出し、高い圧縮効率を実現可能とする。ところが、複数参照フレーム動き補償は、計算量を大幅に増大させる。符号化の処理全体に対して動き補償処理の負荷は、1 枚参照の場合で約 50 %、4 枚参照の場合は約 70 %を占有すると報告されている [4]。

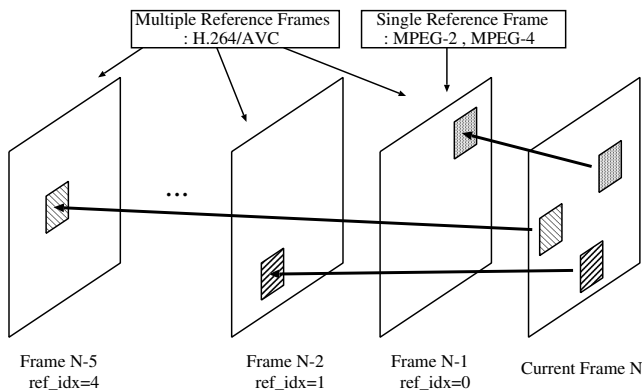


図 1 複数参照フレームの概念

表 1 参照フレーム分布測定実験環境

JM version	JM8.5
Profile	Baseline
Sequence	QCIF, CIF(7 sequences)
QP	I, P:28
Number of Frame	30
GOP Structure	IPPP...(all P)
Number of Reference Frame	5
Frame Rate	30.0[fps]
MC Block Size	16 x 16
Range of search window	+/-16

2.2 参照フレームの決定方法

動き補償において、補償するブロックを決定する場合に用いられるパラメータとして、動きコスト (Motion Cost) があり、主に SAD(Sum of Absolute Differences) が用いられる。SAD は以下の式 (1) で示される。なお、 $org(i, j)$ は処理対象となる画像、 $ref(i, j)$ は参照する画像を表す。

$$SAD = \sum_{i,j} |org(i, j) - ref(i, j)| \quad (1)$$

H.264 では従来の標準符号化方式と比べて、1/4 画素画素生成、そして SAD 計算と、その計算量が従来と比較して大幅に増加する。さらに、複数参照フレームの場合、 n 枚分の処理が必要なので、枚数に比例して動き補償の計算量が増加する。

2.3 参照フレームの分布測定実験

複数参照フレームによる動き補償において、最適な参照フレームの位置を調査するために基礎実験を行う。図 1 に示すように、各参照フレームを $ref_idx (= 0, 1, 2, \dots, NRF - 1)$ で表現する。 NRF は、Number of Reference Frame を指し、とり得る参照フレームの最大値と定義する。 $ref_idx = n$ は、参照フレームが $N - (ref_idx + 1)$ 番目のフレームであることを意味する。表 1 に示した条件にて求めた参照フレームの分布結果を図 2 に示す。Mobile and Calendar, および Paris の例を示す。また、30[frame] の分布平均値を表 2 に示す。

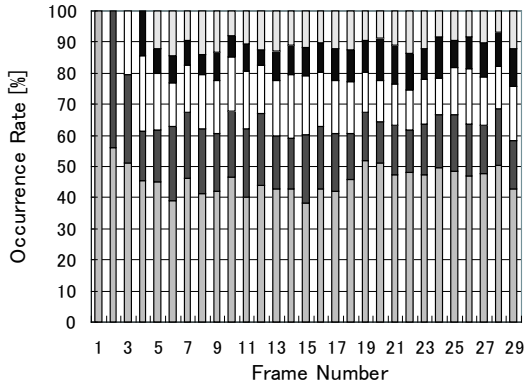
図 2 および、表 2 から、News, Paris は $ref_idx = 0$ が 95 %以上を占有する。しかし、Foreman, Mobile and Calendar は $ref_idx > 0$ がそれぞれ 25 %, 50 %以上を占有している。これは、Foreman は人の顔が交互に動き、Mobile and Calendar はボールの振動と、カレンダー部分の文字などの移動が微細であることが原因と思われる。これ以外の画像においては、News, Paris と同様の傾向にあった。

2.1 に述べたとおり、複数参照フレームによる動き補償は大きな処理量を要するが、最適な参照フレームは $ref_idx = 0$ 、すなわち、直前のフレームとなる場合が多い。したがって、あらかじめ最適な参照フレームが、 $ref_idx = 0$ かどうかを適切に判断できれば、 $ref_idx \neq 0$ の参照フレームにおける探索処理が省略でき、動き探索処理量の削減が可能となる。

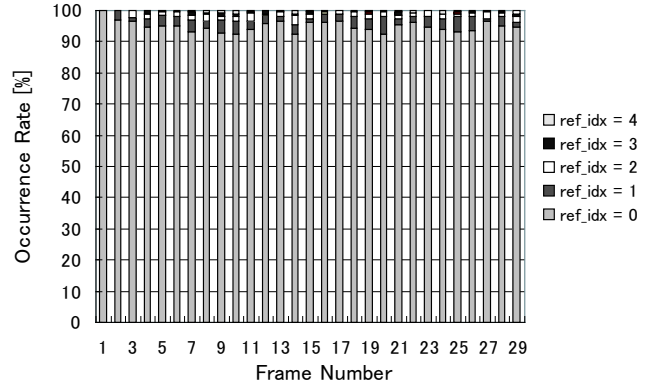
2.4 効率的な動き探索への方針

複数参照フレームでの動き補償において、以下の 2 点から、処理量の削減を狙う。

- (1) 探索不要な参照フレームの選定
- (2) 1 枚あたりの探索点の評価回数の削減



(a) Mobile and Calendar



(b) Paris

図2 ref_idx の分布

表2 ref_idx 分布結果 (30[frame] の平均値)

Sequence	Foreman	News	Mobile	Paris
ref_idx = 0 [%]	76	99	48	95
ref_idx = 1 [%]	13	0.70	19	2.8
ref_idx = 2 [%]	6.7	0.40	15	1.4
ref_idx = 3 [%]	2.9	0.10	8.9	0.40
ref_idx = 4 [%]	2.2	0.20	9.4	0.40

すなわち、参照フレーム全てを詳細に探索するのではなく、大まかな探索で参照フレームの選別を行い、その後、空間探索を用いて詳細な探索を実行する。

文献 [5] によれば、各参照フレームから求まる最適な動きベクトルには相関があると報告されている。特に、等速直線運動においては、物体が直前のフレームで得られた動きベクトルの延長線上の周辺になる確率が高い。ref_idx = n の動きベクトルを $\overrightarrow{MV}(n)$ と定義する。仮に、ref_idx = 0 で空間探索を行い、その結果得られる最適動きベクトル $\overrightarrow{MV}(0)$ の延長線上に、 $\overrightarrow{MV}(1)$ が近似されるならば、各 ref_idx に対して厳密な空間探索をする必要はない。 $\overrightarrow{MV}(0)$ の延長線上の周辺探索で十分である。したがって、最適な ref_idx が求まり次第、そのフレームに関して空間探索を再度詳細に行えば良い。具体的な本提案手法の内容を 3. にて述べる。

3. 複数参照フレームでの効率的な動き探索手法

本章では、2.4 の考察に基づき、参照フレーム間の動きベクトルの相関性を利用した効率的な動き探索手法に関して説明する。ref_idx = 0 に対して空間探索などを行い、 $\overrightarrow{MV}(0)$ は取得済みであることを前提とする。基本概念は、得られた動きベクトル $\overrightarrow{MV}(0)$ の延長線上の周辺画素を探索して得られる仮の動きベクトル $\overrightarrow{tMV}(n)$ を各参照フレームごとに求めて、最小コストを与えるフレーム番号 n を決定する。

3.1 提案手法のアルゴリズム

以下に提案手法の基本的な手順を示す。本提案手法の簡単な概念図を図3に示す。

Step1: ref_idx = 0 にて空間探索を行い、 $\overrightarrow{MV}(0)$ を得る。ここで、縮小画像による擬似的なコストを利用して、ref_idx > 0

の探索の必要性を判断する。この判断方法の詳細は 3.2.3 にて述べる。探索が不要と判断された場合は、Step7 へ移行し、best_ref_idx = 0 とする。そうでない場合は、Step2 へ移行する。

Step2: 一時的な最適動きベクトルを初期化する。すなわち、 $n = 1$, $\overrightarrow{tMV}(0) = \overrightarrow{MV}(0)$ とする。

Step3: 初期動きベクトルを $\overrightarrow{iMV}(n)$ と定義し、以下の式で取得する。

$$\overrightarrow{iMV}(n) = \frac{n+1}{n} \times \overrightarrow{tMV}(n-1) \quad (2)$$

Step4: $\overrightarrow{iMV}(n)$ の位置とその周囲 8 近傍 ($\pm 1/4$ 画素) に対して空間探索を実行し、コストが最小となるベクトルを $\overrightarrow{tMV}(n)$ として保存する。 $\arg_{\overrightarrow{mv}} \min_{\overrightarrow{mv} \in w_i}$ は、次に続く関数を最小化する \overrightarrow{mv} を返す。つまり、探索範囲 w_i において、コストを最小とする動きベクトルが選ばれる。

$$\overrightarrow{tMV}(n) = \arg_{\overrightarrow{mv}} \min_{\overrightarrow{mv} \in w_i} \left[\text{cost}\{\overrightarrow{iMV}(n) + \overrightarrow{mv}\} \right] \quad (3)$$

cost{ \overrightarrow{mv} } は、動きベクトルの動きコストで、式 (4) に定義する。 $R_{mv-bit}(\overrightarrow{mv})$ は動きベクトル自体の符号化に必要なビット量を示す。

$$\text{cost}\{\overrightarrow{mv}\} = \sum |org(\vec{p}) - ref(\vec{p} + \overrightarrow{mv})| + \lambda R_{mv-bit}(\overrightarrow{mv}) \quad (4)$$

Step5: n を 1 増加させる。もしも、 $n = NRF$ ならば Step6 へ進む。そうでないならば、Step3 へ戻る。

Step6: $\overrightarrow{tMV}(n)$ ($n = 0, 1, 2, \dots, NRF - 1$) の中で最小コストとなる n を best_ref_idx とし、式 (5) で求める。 $\arg_n \min_{n \in f_i}$ は、複数参照フレーム f_i 内で、最小コストとする n を返す。

$$\text{best_ref_idx} = \arg_n \min_{n \in f_i} \left[\text{cost}\{\overrightarrow{tMV}(n)\} \right] \quad (n = 0, 1, 2, \dots, NRF - 1) \quad (5)$$

Step7: $n = \text{best_ref_idx}$ となる参照フレームに対して改めて空間探索を実行する。そして、最適動きベクトル $\overrightarrow{MV}(\text{best_ref_idx})$ を獲得する。

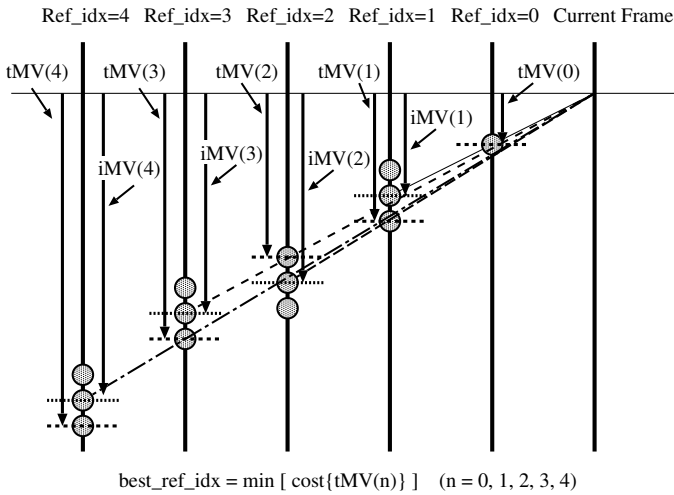


図3 提案手法の概念図

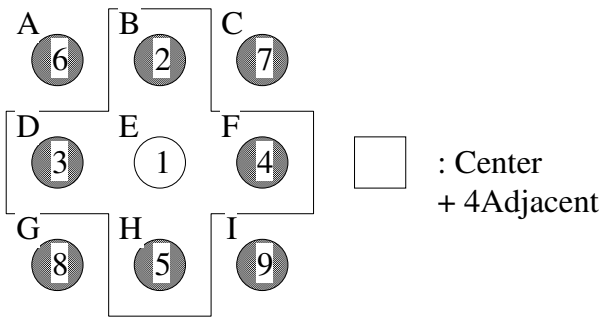


図4 近隣画素の探索順序

3.2 早期打ち切りのための処理

3.1 で説明した手順以外にも、動き探索処理を削減するために以下の打ち切り処理を加える。

3.2.1 延長線の打ち切り処理

Step3 から Step5 のループ処理にて、以下の式 (6) かつ式 (7) を満たした場合には、ループを打ち切り、Step6 へ移行する。

$$\text{cost}\{\overrightarrow{tMV}(n)\} > \text{cost}\{\overrightarrow{tMV}(n-1)\} \quad (6)$$

$$\overrightarrow{tMV}(n) = \overrightarrow{tMV}(n-1) \quad (7)$$

すなわち、あるフレーム n におけるコストが前フレーム $n-1$ より増加したにも関わらず、ベクトルが変わらない場合に関して打ち切り処理を加える。この場合、処理をしなかった参照フレームは評価対象外とする。

3.2.2 近傍探索に関する打ち切り処理

3.1 の Step4 における近傍探索について、図4に示す順序で8近傍の探索を行う。このとき、4近傍を含めた探索で、画素 E が選ばれた場合は、 $\overrightarrow{tMV}(n) = \overrightarrow{tMV}(n)$ として、図4における実線で囲んでいる範囲で探索処理を終了し、残りの A, C, G, I に関しては探索を行わない。そのまま次のフレームへと処理を移す。

3.2.3 縮小画像による擬似コスト考慮

本節では、Step1 における $ref_idx > 0$ の探索必要性を判断する手法についての詳細を述べる。16 × 16 マクロブロック (以下 MB) には、4 つの 8x8SubMB が含まれる。図5に示すよう

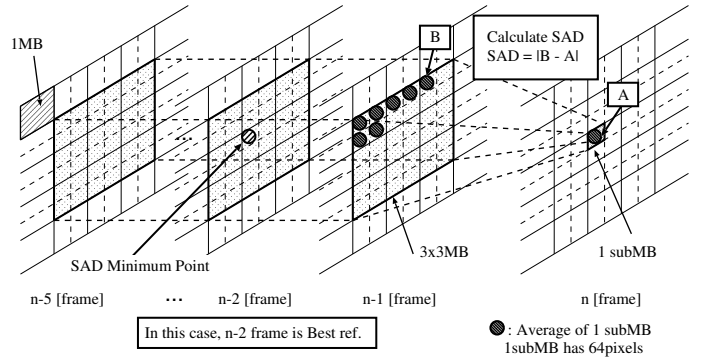


図5 縮小画像のコスト

に、1 つの SubMB の 64 画素の平均値を取り、それを単位として NRF フレーム前までのコストを算出する。この縮小画像を用いた打ち切り処理について、手順を以下にまとめる。

Step(a): 動き探索処理の該当 MB において、各 SubMB の 64 画素の平均値を算出する。

Step(b): 同様にして各参照フレームに対して、該当 MB と同位置を含めた周辺 MB に関して 64 画素ごとの平均値を算出する。全部で $4[\text{subMB}] \times r[\text{MB}] \times s[\text{frame}]$ 個の平均値を求める。 r は 1 枚のフレームの探索 MB 数を示す。

Step(c): Step(a) で求めた平均値と Step(b) で求めた平均値から SAD を計算する。

Step(d): 各 SubMB の最小の SAD が得られる ref_idx を $pre_best_ref_idx[x]$ ($x = 0, 1, 2, 3$) として保存する。 x は SubMB の位置を示す。

Step(e): $pre_best_ref_idx[x] = 0$ を満たすものがあれば、 $ref_idx > 0$ の探索は不要と判断する。この場合、Step3 から Step6 を省略する。

また、本手法において、実験的な結果から、3 × 3MB の SAD 和が以下の式 (8) を満足する場合は、 $ref_idx = 0$ となる場合が多いことを確認している。式 (8) で、 $SAD_{3 \times 3MB}(n)$ は、式 (9) のように、 $ref_idx = n$ の場合の 3 × 3MB の SAD 総和である。 SAD_{1MB} は 1MB 分の SAD の和であり、 (n, i) はそれぞれ $ref_idx = n$ と 3 × 3MB 内の位置を示す。

$$SAD_{3 \times 3MB}(0) < SAD_{3 \times 3MB}(n) (n \geq 1) \quad (8)$$

$$SAD_{3 \times 3MB}(n) = \sum_{i=1}^9 SAD_{1MB}(n, i) \quad (9)$$

4. 評価実験

4.1 実験条件

提案手法の有効性を評価するために、提案手法を参照ソフトウェア [6] に実装し、実験を行う。具体的な実験条件を表3に示す。特に、3.2.3 で示した縮小画像による擬似コスト考慮については、 $pre_best_ref_idx[x] = 0$ を満たす subMB の個数が、処理打ち切りの 1 つの閾値となる。また、動きベクトル探索手法として、文献 [3] による高速探索手法 (FME:Fast Motion Estimation) 適応時における有効性も合わせて評価する。

実験の内容を表4に示す。既存方式 $NRF = 5$ の場合、3.1 にて紹介した方式、3.2.3 で述べた打ち切り処理の追加した方

表 3 提案手法評価実験環境

JM version	JM8.5
Profile	Baseline
Sequence	QCIF, CIF(7 sequences)
QP	I, P:20 ~ 40 (by 2)
Number of Frame	150
GOP Structure	M=1, N=15
Number of Reference Frame	5
Frame Rate	30.0[fps]
MC Block Size	16 × 16
Range of search window	+/-16
Symbol mode	CAVLC
RD Optimization	On
Fast Motion Estimation	On/Off

表 4 比較対象方式

方式	Original ref=5	JM8.5 conventional method (NRF=5)
方式 1	RS on	Reference Select on (3.2.3 off)
方式 2	RS+RC th=1	RS on with RC ("pre.ref = 0" is 1)
方式 3	RS+RC th=2	RS on with RC ("pre.ref = 0" is 2)
方式 4	RS+RC th=3	RS on with RC ("pre.ref = 0" is 3)
方式 5	RS+RC th=4	RS on with RC ("pre.ref = 0" is 4)
方式 6	Original ref=1	JM8.5 conventional method (NRF=1)
方式 7		

式 (4 種類), そして比較対象として既存方式 $NRF = 1$ の場合を行う. 主な評価観点を以下に示す.

- (1) 動き探索時間
- (2) 総符号化時間
- (3) レート歪み特性

4.2 実験結果

実験結果について以下に述べる. FME を用いない場合の動き探索時間 (動きベクトルを決定するのに消費する時間), レート歪み特性 (画質) を図 6, 7 にそれぞれ示す. 各図における凡例の意味は表 4 に示される内容である. (a) Mobile and Calendar と (b) Paris (CIF: 352 × 288) の結果をそれぞれ示す.

既存手法 $NRF = 5$ に対する処理削減率の平均値を表 5 にまとめる. MET (Motion Estimation Time) は動き探索時間を, TET (Total Encoding Time) は総符号化時間をそれぞれ示し, 各々の削減率を α, β とする. 両者は式 (10), (11) で求めた. $MET_{OriginalNRF=5}, TET_{OriginalNRF=5}$ は既存手法 $NRF = 5$ における処理時間を意味し, $MET_{Proposed}, TET_{Proposed}$ は表 4 に示す方式 2 から 6 の処理時間の平均値を意味する.

$$\alpha = \left[1 - \frac{MET_{Proposed}}{MET_{OriginalNRF=5}} \right] \times 100 \quad (10)$$

$$\beta = \left[1 - \frac{TET_{Proposed}}{TET_{OriginalNRF=5}} \right] \times 100 \quad (11)$$

4.3 考察

4.3.1 処理時間

図 6 と表 5 により, 動き探索時間と総符号化時間は, 提案手法を用いることで, 既存手法 $NRF = 5$ よりも削減可能であり, 既存手法 $NRF = 1$ に接近した. 動き探索時間 (MET) は FME に関わらず, 約 70 % から 90 % の削減が可能となった. これは, 提案手法の参照フレーム絞込みが, 動き探索にかかる演算回数を既存手法 $NRF = 1$ の場合と同等になる程度, 削減

表 5 性能評価

Sequence	FME	Method	MET [sec] (α [%])	TET [sec] (β [%])
Foreman (176 × 144)	Off	Original	98.63	182.1
		Proposed	20.91 (78.80)	98.89 (45.69)
	On	Original	2.184	80.81
		Proposed	0.6354 (70.91)	75.62 (6.422)
News (176 × 144)	Off	Original	97.91	180.6
		Proposed	19.80 (79.78)	96.22 (46.72)
	On	Original	1.552	78.98
		Proposed	0.4866 (68.65)	74.09 (6.191)
Mobile (352 × 288)	Off	Original	383.1	831.6
		Proposed	88.18 (76.98)	516.2 (37.92)
	On	Original	11.05	440.0
		Proposed	3.149 (71.50)	421.7 (4.159)
Paris (352 × 288)	Off	Original	381.2	760.1
		Proposed	78.53 (79.40)	433.3 (42.99)
	On	Original	6.895	364.2
		Proposed	2.070 (69.98)	345.0 (5.272)

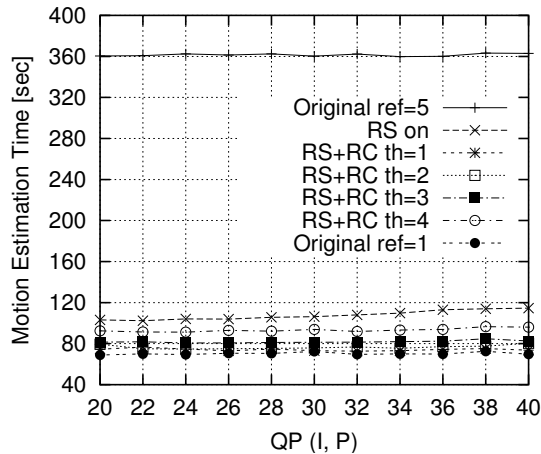
しているためと考えられる. 総符号化時間 (TET) は, FME を使用しない場合は約 40 % から 50 % 程処理時間を削減できたが, 使用する場合では, 10 % 未満になることが多く, 削減利得が減少した. これは, 空間探索の高速化に伴い, 総符号化時間に対する動き探索時間の占める割合が減少したためと考えられる.

また, 総符号化時間は QP の増加に従い, 減少する傾向が確認できた. これは, QP が増加した場合, 量子化ステップサイズが大きいため, MB 内の有意係数が減少する. ゆえに, 量子化, 整数変換, エントロピー符号化などの動き探索以外の処理が簡略化し, その分高速化すると考えられる.

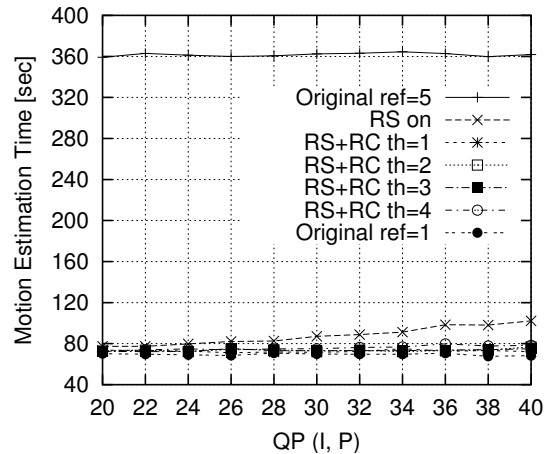
4.3.2 レート歪み特性

レート歪み特性の観点では, 図 7(a) より, Mobile and Calendar は, 既存方式 $NRF = 5$ と $NRF = 1$ で顕著に差が表れており, 複数参照フレームが有効であるとわかる. 提案方式では, $NRF = 5$ に近い特性を維持できている. すなわち, Mobile and Calendar では, 画質を保ちつつ, 動き探索などの処理時間を大幅に削減可能であると判断できる.

2.3 の基礎実験にて証明したように, News, Paris は 90 % 以上のブロックが $ref_idx = 0$ となる画像である. 一方, Foreman, Mobile and Calendar は共に $ref_idx = 0$ の選択率が約 40 % から 60 % であり, $ref_idx > 0$ の参照が有効である画像である. 前者は, 参照枚数を増加させても, ほぼ直前参照になるため, 画質に対する影響は非常に少ない. 既存手法においても $NRF = 1$ と $NRF = 5$ に関して画質の差は表れない. ところが, 後者ではその差が大きく表れる. 参照枚数を増加させることにより, レート歪み曲線が向上する. 提案手法では 5 枚分探索しているため, レート歪みに対して利得が得られると考えられる.

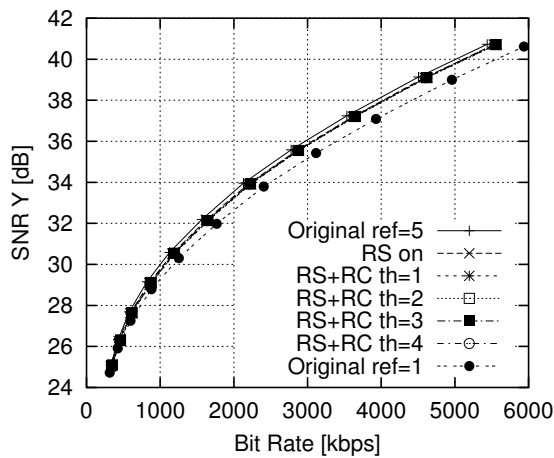


(a) Mobile and Calendar

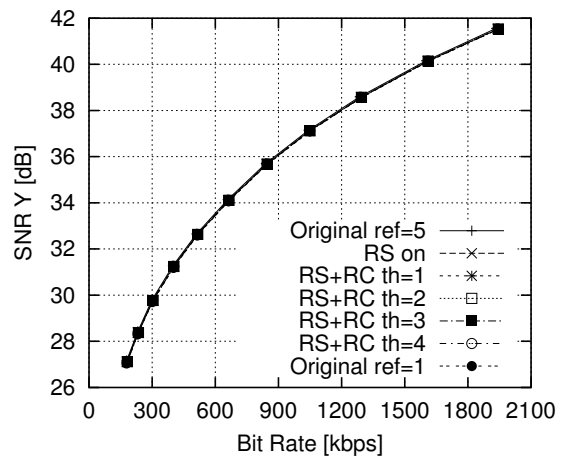


(b) Paris

図 6 動き探索時間



(a) Mobile and Calendar



(b) Paris

図 7 レート歪み特性

4.3.3 各方式の比較

各方式に関して比較する．本稿では，主に 5 種類の提案手法を用意した．縮小画像の擬似コストを導入することで，入れない場合に比べて動き探索の時間を削減することが可能と確認できた．どの閾値が最良の結果を与えるのかは画像と QP 値に依存していた．しかし，Mobile and Calendar などの直前以外の参照が多い場合は，閾値が低いほど処理時間の削減割合が大きい傾向が見られた．

5. まとめ

本稿では，複数参照フレームに対して動きベクトルの相関性に着目し，縮小画像によるコストを考慮して打ち切り処理を加えることで，効率的な動き探索手法を提案し，動き補償の処理削減，高速化の検討を行った．実験結果から，本提案手法により，レート歪み特性の劣化を抑えながら，総符号化時間と動き探索時間を参照フレーム 1 枚時の場合に近似が可能と確認できた．実験においては， 16×16 ブロックサイズでの場合のみを対象としているため，他ブロックサイズ (8×8 , 4×4 など) へ対応した場合の性能評価，また，双方向予測 (B ピクチャ) への拡張が今後の課題である．

文献

- [1] ISO/IEC 14496, "Advanced Video Coding," (ITU-T Rec. H.264) 2003.
- [2] Thomas Wiegand, Gary J.Sullivan, Gisle Bjontegaard, Ajay Lutha "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, VOL.13, NO.7, JULY 2003
- [3] Mei-Juan Chen, Yi-Yen Chiang, Hung- Ju Li, Ming-Chieh Chi "Efficient Multi-Frame Motion Estimation for MPEG-4 AVC/JVT/H.264," Circuits and Systems, 2004 ISCAS '04. Proceedings of the 2004 International Symposium on , Volume:3 ,23-26 May 2004
- [4] Woong IL Choi, Byeungwoo Jeon, Jechang Jeong, "Fast Motion Estimation with Modified Diamond Search for Variable Motion Block Sizes," Proceedings. 2003 International Conference on , Volume: 2 ,14-17 Sept. 2003 Pages:II - 371-4 vol.3
- [5] Xiang Li, Eric Q. Li, Yen-Kuang Chen, "Fast Multi-Frame Motion Estimation Algorithm with Adaptive Search Strategies in H.264," IEEE 2004
- [6] <http://iphome.hhi.de/suehring/tml/>, "JM Reference Software"