RI-001

時間方向からの予測動きベクトル推定に基づくメニーコア向け並列動き探索手法

Highly Parallel Motion Estimation Method based on Temporal Motion Vector Prediction Estimation for Many-core Platform

工藤忍 † 北原正樹 † 清水淳 †
Shinobu Kudo Masaki Kitahara Atsushi Shimizu

1. まえがき

H.264/AVC や HEVC などの動き補償を利用したハイブリッド映像符号化方式では，符号化処理の大部分を動き探索処理が占めている．近年，動き探索処理を高速化するアプローチとして，GPU やメニーコアといった大規模並列処理を利用する手法が検討されている．従来は，低遅延性を実現するためにフレーム内をブロック並列するアプローチが採用されている．隣接するブロックの動き情報を利用することができないことによる符号化効率の低下を抑えるために，符号化済みフレームの動きベクトルを利用する手法などが提案されているが，階層構造で予測動きベクトルの推定精度が低下するという 問題がある．本稿では，対象フレームと同じ参照フレームを参照する未符号化フレームの動きベクトルに対してブロックサイズ判定を行い，適切なブロックサイズの動きベクトルのみを用いて時間方向から予測動きベクトルを推定することで，動きベクトル残差のレート 歪み最適化を考慮した並列動き探索手法を提案する．評価実験により 本手法の有効性を示す．

2. 従来の動き探索並列化

本章では，HEVC の予測動きベクトルと 従来の動き探索並列化手法を述べる．

2.1.HEVC の予測動きベクトル

映像符号化ではフレームをブロックに分割し，予測ブロックごとに動き探索を実施する．動きベクトルは予測対象ブロックに隣接するブロックが持つ符号化済みの動きベクトル (予測動きベクトル) との差分を計算し，残差ベクトルを符号化する．HEVC[1] では予測動きベクトルは図 1 のように参照される．HEVC の参照モデルである HM10.0[2] では，最適な動きベクトルは以下のレート 歪最適化を考慮して算出される．

$$\mathbf{mv} = \arg\min_{\mathbf{mv}}\{\mathrm{SAD} + \lambda \mathrm{R}(\mathbf{mv} - \mathbf{mvp})\} \quad (1)$$

ここで，SAD は予測ブロックと参照ブロックの絶対誤差和，λ はラグランジアン定数，$\mathrm{R}(\mathbf{mv} - \mathbf{mvp})$ は動きベクトル $\mathbf{mv}$ と予測動きベクトル $\mathbf{mvp}$ の差分符号量をそれぞれ表す．つまり，最適な動きベクトルは予測ブロックの誤差と 差分動きベクトルの符号量が最適なバランスになるように決定される．

動き探索を並列化する方法として，低遅延性を実現するためにフレーム内をブロック並列するアプローチが考えられる．しかし，フレーム内を分割すると分割境界で隣接するブロックの予測動きベクトルを参照することができないため，(1) 式において差分動きベクトルの符号量を考慮することができず，符号化効率が低下するという 問題がある．

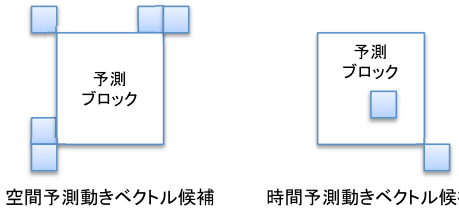

図 1 HEVC の予測動きベクトル候補

2.2. 従来の並列動き探索手法

前節で述べた問題を解決する方法として，パイプライン処理に基づく 並列化手法と 予測動きベクトルの時間方向推定に基づく 並列化手法が報告されている．

1 つ目は隣接するブロックの予測動きベクトルを参照できるようにパイプライン処理を行う 手法である．Zhou らは符号化ブロック内部を複数の領域に分割し，その領域間では動きベクトルを参照しないように制限することで，それらの領域の動き探索を並列に行う 手法を提案している [3]．Yan らはこの手法に改良を加え，依存関係のない予測ブロックの更なる並列化と，符号化ブロックのパイプライン処理によって並列度を高めた [4]．しかし，これらの手法はシーケンスサイズが 1080p の場合で並列度が最大でも 120 並列であるため，数百から数千個の並列度を持つ GPU やメニーコアを用いた大規模並列処理には適していない．

2 つ目は空間的な依存関係を利用せず，予測動きベクトルを時間方向から推定することで，全ての予測ブロックに対する動き探索の並列化を実現する手法である．Gao らは直前の符号化済みフレームの動きベクトルを利用して予測動きベクトルを推定し，動きベクトル残差のレート 歪最適化を考慮した動き探索手法を提案している [5]．しかし，この手法は参照構造がインターピクチャが P ピクチャのみで構成されるような低遅延構造を想定しているため，低遅延構造よりも符号化効率の高い階層 B 構造のような参照構造に適用する場合には符号化済みフレームと符号化対象フレームの時間距離が離れるため，被写体の動きが連続していない場合には予測動きベクトル精度が低下するという 問題がある．

† 日本電信電話株式会社, メディアインテリジェンス研究所

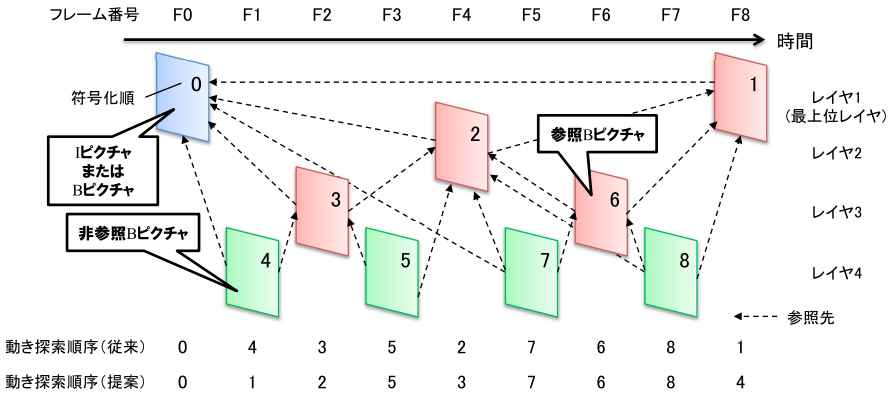

図 2 提案法の動き探索順序

3. 提案法

本章では，提案する並列動き探索手法について述べる．まず提案法のポイントについて説明した後，具体的なアルゴリズムについて説明する．

3.1. 提案法のポイント

本稿では，時間方向から予測動きベクトルを推定するアプローチに基づいた手法を提案する．以下に提案法のポイントを述べる．

1. 未符号化フレームの動きベクトルを利用
   符号化対象フレームと同じ参照フレームを参照する未符号化フレーム (中間フレーム) に着目し，図 2 に示すように参照フレームとの距離が近いフレームから順に動き探索を実施する．前のフレームで求めた動きベクトルを用いて予測動きベクトルを推定し，動き探索を行う．これを符号化対象フレームまで繰り返す．符号化対象フレームと参照フレームの間に位置するフレームの動き探索結果を利用することで，時間方向からの予測動きベクトルの推定距離が短くなり，精度の高い推定を行うことができる．また，参照フレームとの距離が近いフレームから探索することで動き探索範囲を小さくし，演算量の削減効果も期待できる．更に，中間フレームで算出した動きベクトルは後の符号化時に再利用することができる．

2. 動きベクトルの連結
   予測動きベクトルの推定では，中間フレームと対象フレーム間の動き探索結果 (例えば図 2 では F4 から F8 への動きベクトル) も利用し，動きベクトルを連結して算出する．このように中間フレームと対象フレーム間の動きも考慮することで，被写体の動きが複雑な場合においても，従来のように前のフレームの動き探索結果をスケーリングして利用する場合よりも高精度な予測動きベクトル推定を実現できると考えられる．

3. ブロックサイズ判定による動きベクトルの選定
   中間フレームにおいてブロックサイズ判定を実施することで，予測動きベクトルの推定に利用する動きベクトルを最適な予測ブロックサイズのみに限定する．ブロックサイズ判定によって決定した予測ブロックサイズは被写体の動きを的確に表現するブロックであるため，ここで決定した予測ブロックサイズのみを利用することで，誤った動きベクトルによる推定誤差を防ぎ，予測動きベクトル精度を向上させる．

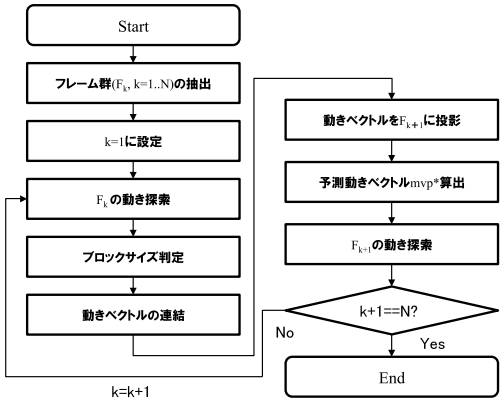

図 3 提案法フロー

3.2. 動き探索アルゴリズム

本提案手法の具体的な手順について説明する．図 3 に提案法のフローを示す．

提案法では同じ参照フレームを参照するフレーム群 $F_k(k = 1..N)$ を対象として以下の手順を実施する．フレーム群とは，例えば図 2 では F0 を共通の参照フレームとする {F1,F2,F4,F8} である．

まず，フレーム群 $F_k(k = 1..N)$ の中で参照フレームに近いフレームから動き探索を実施する．F_k の動き探索では参照フレーム F_0 に対する動き探索 (L0 方向) と次のフレーム F_{k+1} に対する動き探索 (L1 方向) を実施する．この時，L1 方向は対象フレームの符号化が終わっていないため，原画像を用いて探索を行う．そして，各予測ブロックサイズに対してブロックサイズ判定を実施する．判定は以下の RD コストを用いる．

$$\mathrm{RDcost} = \mathrm{D} + \lambda \mathrm{R} \qquad (2)$$

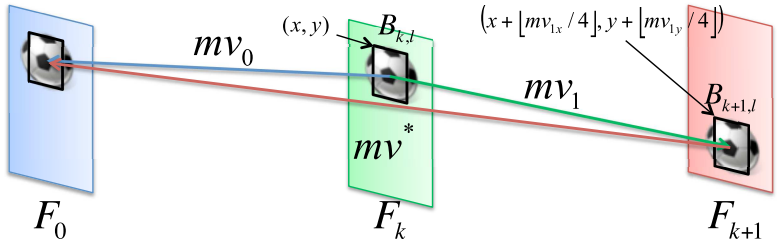

図 4 動きベクトルの連結及び投影

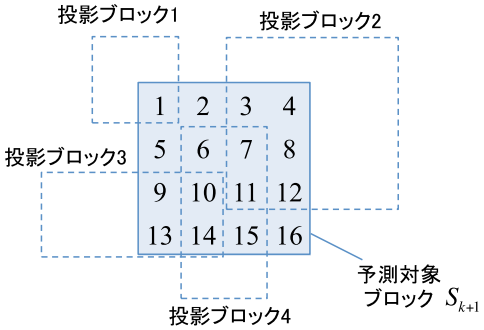

図 5 投影されたブロックの例

表 1 符号化条件

フレーム数	先頭 25 フレーム
テスト条件	RA(ランダムアクセス), Main Profile
イントラ間隔	32
参照構造	階層 B 構造 レイヤ数：4 レイヤ 最上位レイヤ間隔：8
量子化値 (QP)	22, 27, 32, 37
動き探索範囲	64
動き探索手法	EPZS(default)

ここで，D は原画像と予測画像の誤差，R は符号量，λ はラグランジアン定数を表す．

判定の結果，RD コストが最小となった予測ブロックサイズの動きベクトルを F_{k+1} に投影する．図 4 に動きベクトルの連結及び投影の概要を示す．

F_k の l 番目のブロック $B_{k,l}$ の左上隅の座標 (x, y) に対する L0 方向の動きベクトルを $\mathbf{mv}_0 = (mv_{0x}, mv_{0y})$，L1 方向の動きベクトルを $\mathbf{mv}_1 = (mv_{1x}, mv_{1y})$ とすると，F_{k+1} に対応するブロック $B_{k+1,l}$ の左上隅の座標 (x^*, y^*) 及び投影動きベクトル $\mathbf{mv}^*$ は以下のように算出される．

$$(x^*, y^*) = (x + \lfloor mv_{1x}/4 \rfloor, y + \lfloor mv_{1y}/4 \rfloor) \quad (3)$$

$$\mathbf{mv}^* = \mathbf{mv}_0 - \mathbf{mv}_1 \quad (4)$$

$\lfloor \cdot \rfloor$ を行っている理由は動きベクトルが 1/4 画素精度であることを考慮して，投影位置を整数にするためである．

次に，投影された動きベクトルから F_{k+1} の動き探索で利用する予測動きベクトル $\mathbf{mvp}^*$ を推定する．図 5 に投影されたブロックの様子を示す．

予測動きベクトルは以下のように予測対象ブロック S_{k+1} 内に投影されている動きベクトルの各要素の平均値を用いる．

$$\mathbf{mvp}^* = \frac{1}{M} \sum_{i=0}^{M} \mathbf{mv}^*{}_i, \mathbf{mv}^*{}_i \in S_{k+1} \quad (5)$$

ここで，M は投影されている動きベクトルの個数を表す．予測対象ブロックの中に動きベクトルが 1 つも投影されていない場合はゼロベクトルを予測動きベクトルとする．推定した予測動きベクトルを用いて (1) 式によるレート歪最適化を考慮した動き探索を実施する．

上記の手順を符号化対象フレームまで繰り返し実施する．

4. 評価実験

本章では，提案する並列動き探索手法の有効性を示すために，シミュレーション実験を行う．

4.1. 実験条件

本提案手法を HEVC の参照モデルである HM10.0[2] に実装した．実験条件は標準化共通条件 [6] に従い，4K 映像 (3840x2160) 及び，HD 映像 (1920x1080) の 2 種類を用いた．4K 映像として使用する AquaSkytree 及び AquaTrain は，前者は車が左右に高速に移動するシーン，後者は画面全体が平行移動するシーンを特徴とする映像である．実験環境として Xeon-Phi 7120P(1.238GHz, 61core, 16GB メモリ) を用い，240Thread のネイティブ実行にてエンコードを行った．表 1 に符号化条件を示す．

提案法では空間からの予測動きベクトルは参照せず，提案する時間方向からの予測動きベクトルを用いて動き探索を実施した．この時，最上位レイヤ (レイヤ 1) での効果を検証するために，レイヤ 1 での動き探索にのみ提案法を適用し，それ以外はゼロベクトルとし，予測動きベクトルの推定にはレイヤ 2 の動き探索を利用した．また，提案法の各ポイントの効果を明確にするために，HM10.0 において動き探索時に空間方向からの予測動きベクトルをゼロベクトルとした場合 (手法 1) 及び，提案法において中間フレームのブロックサイズ判定を行わずに全ブロックサイズの動きベクトルを

表 2 各手法の設定

	動き探索順序	動きベクトル投影	ブロックサイズ判定
手法 1	従来 (符号化順)	×	×
提案手法 1	L0 ： 提案, L1：従来	従来 [5]	全ブロックを投影
提案手法 2	提案	提案 (連結)	提案

投影し，動きベクトルを連結せずに L0 方向の動きベクトルに対して従来法 [5] の予測動きベクトル推定を適用した場合 (提案手法 1) の実験も行った．表 2 に各手法の設定について示す．

4.2. 結果及び考察

表 3 に手法 1 及び，本提案手法 1, 2 の符号化性能 (BD-BR) 及び動き探索の演算量削減率 (TR) の平均をそれぞれ示す．演算量削減率は以下の式を用いた．

$$\mathrm{TR}(\%) = \frac{T_{anchor} - T_{test}}{T_{anchor}} \times 100 \qquad (6)$$

なお，本結果のアンカーは HM10.0 オリジナルの結果である．

表 3 より，手法 1 では動き探索の処理時間が HM オリジナルよりも平均で 97.0%削減されている．しかし，全ての映像において符号化性能が低下し，平均で 12.4%の劣化となっている．動きの少ない Kimono や BQTerrace では 1%未満の符号化性能低下であるが，動きの大きい AquaTrain や BasketballDrive では 10%以上も符号化性能が低下している．性能低下の主な要因は，特に参照フレームが離れたフレームにおいて，空間からの予測動きベクトルが参照できないため，動き探索範囲が不足しているためであると考えられる．提案手法 1 では中間フレームの動きベクトルを利用した効果によって，手法 1 よりも符号化効率が改善されている．提案手法 2 は手法 1 及び提案手法 1 と同程度の動き探索の削減率を達成しつつ，提案手法 1 よりも符号化性能の平均が優っている．特に BasketballDrive においては提案手法 1 よりも 2.3%の改善率を示している．BasketballDrive では動きが複雑であるため，モード判定によるブロックサイズの選定と動きベクトルの連結の効果が現れたと考えられる．

5. まとめ

本稿では，符号化対象フレームと同じ参照フレームを参照するフレームに着目した時間方向からの予測動きベクトル推定に基づき，レート歪最適化を考慮した大規模並列動き探索手法を提案した．評価実験により 240 並列において平均で 2.2%の符号化効率低下で動き探索演算量を 96.6%削減した．今後の課題としては，GOP 内でシーンチェンジが発生した場合への対策や並列度を高めた場合における処理速度の検証などが挙げられる．

謝辞

本稿で用いた映像の一部は NTT ドコモ様の許諾を得て使用したものです．この場を借りて御礼申し上げます．

参考文献

[1] J. -R. Ohm, G. J. Sullivan, W. J. Han and T. Wiegand,“ Overview of the High Efficiency Video Coding(HEVC) Standard,” IEEE Trans. Circuits and Systems for Video Tech., Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.

[2] HM software 10.0, https://hevc.hhi.fraunhofer.de/trac/hevc/browser/branches/HM-10.0-dev.

[3] M. Zhou,“ AHG10: Configurable and CU-group level parallel merge/skip, ” JCTVC-H0082, Feb. 2012.

[4] C. Yan, Y. Zhang, F. Dai and L. Li,“ Highly Parallel Framework for HEVC Motion Estimation on Many-core Platform, ” Data Compression Conference 2013, pp. 63-72, 2013.

[5] Y. Gao and J. Zhou,“ Motion vector extrapolation for parallel motion estimation on GPU, ” Springer Science+Business Media, 2012.

[6] F. Bossen, “ Common HM test conditions and software reference configurations, ” JCTVC-L1100, 12th Meeting Geneva, CH 14-23 January 2013.

表 3 符号化性能と動き探索の演算量削減率

		手法 1		提案手法 1		提案手法 2	
Sequence		BD-BR[%]	TR[%]	BD-BR[%]	TR[%]	BD-BR[%]	TR[%]
4K	AquaSkytree	7.5	96.9	5.0	96.6	5.0	96.5
	AquaTrain	51.8	97.0	2.3	96.6	3.2	96.6
HD	Kimono	0.7	97.2	0.0	96.8	-0.3	96.8
	ParkScene	0.9	97.4	0.9	96.9	0.5	96.8
	BasketballDrive	13.2	96.8	7.2	96.3	4.9	96.3
	BQTerrace	0.3	96.9	0.2	96.6	0.1	96.6
Overall[%]		12.4	97.0	2.6	96.6	2.2	96.6