

## H.264 の複数参照フレームにおける動き探索の高速化手法

A Fast Motion Estimation Method for Multi Reference Frame of H.264

鈴木 真吾†

Shingo Suzuki

川島 裕司†

Yuji Kawashima

菊池 義浩†

Yoshihiro Kikuchi

## 1. まえがき

H.264/MPEG-4 AVC の動き探索では、ブロックサイズ、参照フレーム枚数、動き探索精度の自由度を高めることにより、精度の良い動き予測を行っている。しかし、上記符号化モード全ての組み合わせについて動き探索を行うことは、演算量が増大するため現実的とは言えない。そこで、動き探索の高速化手法としてブロックサイズ、参照フレーム枚数の削減や探索範囲を縮小するなど既に様々な検討がなされている。

本稿では、動き探索の更なる高速化を行うため、参照フレーム毎の探索範囲に関して検討を加える。また H.264 の 1/2 画素精度の動き探索は、6 タップのフィルタを用いているため従来の動画像符号化のような平均値フィルタと比較すると演算量が増加している。本稿では、整数画素精度探索(以後、整数探索)と 1/2, 1/4 画素精度探索(以後、非整数探索)を分けて考え、整数探索の結果に基づき適応的に非整数探索を行う参照フレームを絞り込む手法を提案する。

## 2. 複数参照フレームの高速化

複数参照フレームは、直前の参照フレームで予測しづらい Uncovered-back-ground での予測効率の低減を抑制する効果があるため、必要な符号化モードであると考える。そこで本稿では、参照フレームを 3 枚用い、その際の演算量削減について説明する。

## 2.1. 探索範囲の縮小による演算量削減

予備検討として、参照フレーム毎の動きベクトルの分布を調査した。調査には動きの大きい画像 1 および動きの小さい画像 2 を用いた。画像 1 の結果を図 1 に示す。符号化対象フレームの直前の参照フレーム 1 の動きベクトルは広い範囲に分布しているのに対して、参照フレーム 2, 3 の動きベクトルは狭い範囲に分布しており、約 7 割の動きベクトルが ±2 画素以内に収まっていることを確認した。また、画像 2 では、参照フレーム 2, 3 を選択した動きベクトルのほぼ全てにおいて ±2 画素以内に収まっていることを確認した。

このように参照フレーム 2, 3 枚目の動きベクトルが比較的狭い範囲に収まっている要因としては、以下の 2 点が挙げられる。

- (1) 符号化対象フレームと参照フレームの相関は時間距離に依存しているため、通常時は参照フレーム 1 枚目での予測効率が良い。
- (2) 参照フレームの符号量は符号化対象フレームから離れるほど大きくなるので、そのトレードオフとして予測

誤差が小さくなるか、動きベクトルの符号量が小さくなる必要がある。動きベクトルの大きさが同じならば基本的には参照フレーム一枚目が有利であると考えられる。つまり 2, 3 枚目の参照フレームの場合、動きベクトルが小さい領域が選択されやすい傾向にあると考えられる。

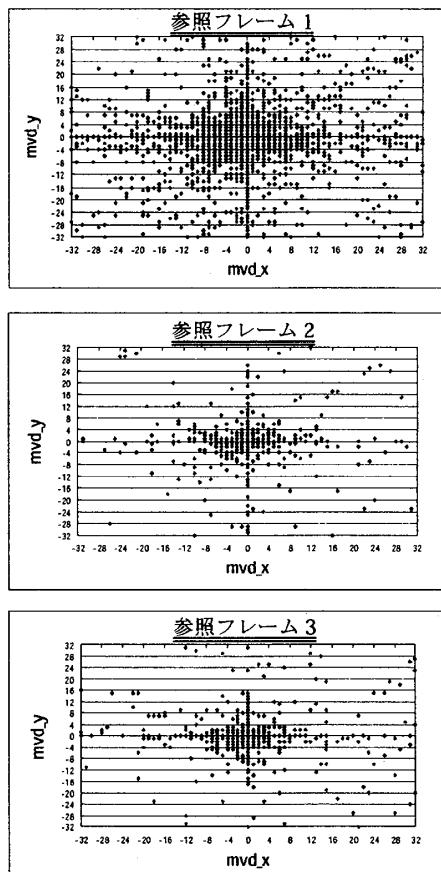


図 1. 動きベクトルの分布

上記結果を用いて本稿では、図 2 に示すように 2, 3 枚目の参照フレームの探索範囲を小さく設定することで演算量の削減を図る。今回、参照フレーム 1 枚目の探索範囲を ±16 とし、2, 3 枚目の参照フレームの探索範囲を ±2 と設定した。

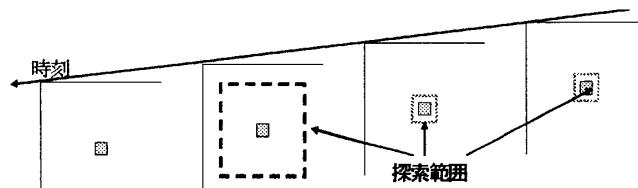


図 2. 各参照フレームの探索範囲

## 2.2. 参照フレーム省略による演算量削減

整数探索は2.1章のように参照フレーム*i*枚目以降の探索範囲を縮小することにより演算量を大幅に削減した。しかし、複数参照フレーム予測では、参照フレーム枚数に応じて演算量が増加する。そこで本稿では、整数精度の探索において各ブロックの最適な参照フレームを決定し、その結果に基づき非整数探索を行う参照フレームを適応的に絞り込む手法を導入する。

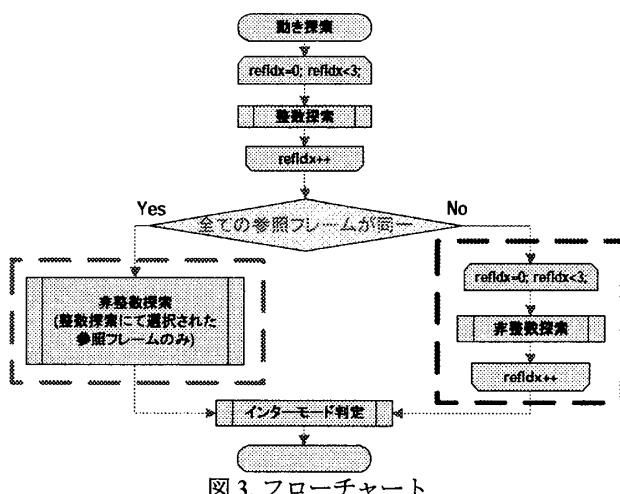
予備検討として、整数探索後に16x16画素ブロックやブロック分割された各ブロックが参照しているフレームと非整数探索後に選択されたブロックが参照しているフレームが全て同一である頻度を調査した。調査には動きの大きい画像1と動きの小さい画像2を用いた。結果を表1に示す。結果より、整数探索時に各ブロックが選択した参照フレームに偏りが見られる場合、その参照フレームが非整数探索後に選択される参照フレームとなる割合が高く、動きの少ない画像で約9割、動きの大きい画像でも7割以上となっている。

表1. 整数探索と非整数探索の参照フレームの関係

画像	QP	整数探索と非整数探索後の 参照フレームが同一
画像1	33	80.62%
	28	76.49%
画像2	21	88.91%
	16	86.16%

上記結果を用いて本稿では、適応的に非整数探索を行う参照フレームを絞り込む手法を導入する。フローチャートを図3に示す。

- (1) 整数探索の結果、動き探索の対象としたブロックが最適とみなす参照フレームを決定する。
- (2) 動き探索の対象とした全ブロックが同一の参照フレームを選択している場合、非整数探索を行う参照フレームを1枚に絞り込み、探索を行う。



## 3. 実験と考察

上記2つの提案方式のシミュレーション実験を行った。実験に用いたシーケンスは動きの大きい画像1と動きの小さい画像2である。他のシミュレーション条件を表2に示す。また、2.1章の探索範囲削減手法を提案手法1と

し、2.2章の参照フレーム絞り込み手法を提案手法2とする。

表3に示すように参照フレーム2,3枚目の探索範囲を±16から±2に縮小することで画像に依存することなく探索点数を約65%程度削減していることがわかる(※表3,4の“演算量削減前”は全参照フレームにて探索範囲±16, 全探索である)。また、表5の符号化性能を示す。提案方式1は演算量削減前の符号化性能と比べ1%程度に画質劣化に抑えられていることがわかる。

参照フレーム絞り込みによる非整数探索の演算量削減の結果を表4に示す。提案方式2は整数探索時に各ブロックが選択する参照フレームの偏りを絞り込みの判定条件に用いているため画像に依存するが、平均的に探索点数を50%程度削減している。また、符号化性能においても1%程度の画質劣化に抑えられていることがわかる。

表2. シミュレーション条件

プロファイル	Baseline	
QP	全フレーム固定	
エンコード枚数	300枚	
フレームレート	15[Hz]	
ピクチャ構成	I, P, P, P, P...	
参照フレーム枚数	3枚	
ブロック形状	8x8以上	
探索手法/探索範囲	参照フレーム1	全探索/±16
	参照フレーム2, 3	全探索/±2
最適化手法	R-D最適化なし, Hadamard変換なし	

表3. 探索範囲縮小による探索点数の変動(1MB)

探索範囲探索点数	提案方式1	提案方式2
画像1, 画像2	3267	1139(削減率:65%)

表4. 参照フレーム制限による探索点数の変動(1MB平均)

探索範囲探索点数	提案方式1	提案方式2
画像1	48	28(削減率:43%)
画像2	48	21(削減率:57%)

表5. 符号化性能

画像	QP	方式	Distortion(kbps)	SNR(dB)
画像1	33	演算量削減前	206	33.78
		提案方式1	209	33.75
		提案方式2	210	33.73
	28	演算量削減前	388	36.78
		提案方式1	391	36.76
		提案方式2	394	36.75
画像2	21	演算量削減前	67	43.67
		提案方式1	67	43.66
		提案方式2	68	43.68
	16	演算量削減前	124	47.12
		提案方式1	124	47.13
		提案方式2	125	47.11

## 4. まとめ

本稿では、参照フレームと動きベクトルの相関を利用して探索範囲の削減した。また、整数探索での結果の有効性を確認し非整数探索を行う参照フレームを適応的に絞り込む手法を導入した。上記2つの手法ともに画質の劣化を抑えつつ、演算量の削減が図れることを確認した。