

## H.264 向け RDO に基づいた動きベクトル検出手法の一検討

小島 康介<sup>†</sup> 橋本 亮司<sup>††</sup> 藤田 玄<sup>†</sup>

<sup>†</sup> 大阪電気通信大学大学院工学研究科

<sup>††</sup> 大阪大学大学院情報科学研究科

あらまし 本稿では、RDO(Rate-Distortion Optimization)に基づく H.264 動き検出手法を提案する。符号化結果を正確に反映する RDO では優れた符号化効率を得られることから、さまざまな符号化パラメータの選択に用いられているが、その演算量の大きさから動き検出への適用例はほとんど報告されていない。提案手法では、小数画素精度の動き検出にのみ RDO を適用することで、実用的な処理時間で符号化効率の向上を実現する。ソフトウェアシミュレーションの結果、既存の SAD 方式に比べ、31.5%~47.1% の実行時間の増加にて、1.71%~3.11% のビット量削減を実現した。

キーワード H.264, 動き検出, RDO

## RDO based sub-pixel motion estimation method for H.264

Kosuke KOJIMA<sup>†</sup>, Ryoji HASHIMOTO<sup>††</sup>, and Gen FUJITA<sup>†</sup>

<sup>†</sup> Graduate School of Engineering, Osaka Electro-Communication University

<sup>††</sup> Graduate School of Information Science and Technology, Osaka University

**Abstract** This paper describes a motion estimation method for H.264 based on RDO (Rate-Distortion Optimization). RDO have not been adopted for motion estimation, because calculation of RD cost, which consists of accurate bits and distortion gotten from encoding and decoding, incurs considerable computation cost. In the proposed method, high precision motion estimation with practical computation cost can be attained by only applying RDO to sub-pixel motion estimation. In software simulation results for several video sequence, the proposed method can achieve about 1.71%~3.11% bit reduction in compared with conventional SAD based approaches.

**Key words** H.264, motion estimation, RDO

### 1. はじめに

BS デジタルや地上デジタルなど HD 動画像の普及が進んでいる。このような大容量・高精細映像を圧縮・復号する技術として H.264/AVC [1] が現在注目されている。H.264 は優れた符号化効率の実現が可能であるが、その要因として符号化の際に用いるさまざまなパラメータ (以下、符号化パラメータ) を状況に応じて細かく使い分けていることが挙げられる。これらの符号化パラメータの選択が適切でない場合、本来の性能が発揮できないことからパラメータ選択手法が非常に重要となる。H.264 の優れたパラメータ決定手法として RDO(Rate-Distortion Optimization) が知られている。RDO では符号化結果そのものを評価値として用いることで、適切な符号化パラメータ選択を行うことができるが、処理量が非常に大きいため、選択肢の多いパラメータへの適用が困難という課題がある。

一方で符号化パラメータの中でも、動きベクトルは圧縮性能に大きく影響を与えることから、そのパラメータ選択手法であ

る動き検出に関しては多くの研究がなされている。動きベクトルは候補数が非常に多いため、処理量の少ない評価値を用いることが一般的であるが、RDO に比べて必ずしも適切な動きベクトルを選択できていないとは限らない。

本稿では、動きベクトルの評価手法として RDO を用い、より高精度に動きベクトルを検出する手法の性能評価とその評価結果について述べる。

### 2. H.264 と RDO の概要

#### 2.1 H.264 動画符号化標準

H.264 は ITU-T と MPEG により共同標準化された動画符号化方式である。この H.264 は、従来符号化方式に比べ、高い符号化効率を実現している。その要因として、様々なブロックサイズによる動き補償、1/4 画素精度の動き補償などが挙げられる。H.264 は目的によって複数の符号化ツールから必要なツールを組み合わせてることにより、低解像度から高解像度の画像まで柔軟に対応できる。

## 2.2 RDO(Rate-Distortion Optimization)

RDOとは、複数のパラメータ候補が存在する場合に、パラメータ*i*で符号化した時の符号化歪み $D_i$ と、発生符号量 $R_i$ から式(1)により計算される $RDcost J_i$ が最小になるパラメータを選択する手法である。この時の $\lambda$ は式(2)により計算され、量子化パラメータからラグランジュ未定乗数法に基づいて定まる。RDOでは $D_i$ を求める際にSSD(Sum of Squared Difference)を用いる事が多い。SSDとは式(3)で表される演算であり、現画像と復号各画の各画素の差分値二乗和となる。

RDOは実際の符号量と復号時に得られる画像データを評価値としてパラメータを決定するため、既存の評価方式よりも精度の高い選択を行うことができる。しかし候補パラメータ毎に局所符号・復号処理を必要とするため演算量が多くなる。特に動きベクトルといった候補数が多いパラメータに適用した場合は演算量に大きな課題を抱えることとなる。その為、式(1)式(2)に基づくRDOを用いたのは動き検出の実装例はこれまでほとんど報告されていない。

$$J_i = D_i + \lambda R_i \quad (1)$$

$$\lambda = (2 \frac{QP}{6} - 2)^2 * 0.9 \quad (2)$$

$$SSD = \sum_{i,j=0}^{Blocksize} (f_{cur}(i,j) - f_{dec}(i,j))^2 \quad (3)$$

## 3. 従来手法

H.264では動きベクトル探索時に使用する小数精度向けの画素値はFIR(Finite Impulse Response)フィルタを用いて生成する。まず整数画素から6タップFIRフィルタ：係数 $[(1, -5, 20, 20, -5, 1)/32]$ を用いて1/2画素の予測信号を求める。次に2タップFIRフィルタ：係数 $[(1/2, 1/2)]$ を用いて1/4画素の予測信号を求める[2]。リファレンスソフトであるJM[3]にて実装されている動き検出では以下の階層探索を行う。

- step1. 整数精度探索
- step2. 1/2画素精度探索
- step3. 1/4画素精度探索

まず整数精度で探索を行う。次に整数精度で求めた動きベクトル周辺の1/2画素の探索を行う。最後に1/2画素精度で求めた動きベクトル周辺の1/4画素の探索を行う。整数精度探索アルゴリズムは全探索以外に、高速探索アルゴリズムであるDiamond Search[4]やUnsymmetric-Cross Multi-Hexagon Grid Search(以下UMH)[3]がよく用いられている。

また多くの実装[5]~[10]では候補動きベクトルの評価値を求める際、計算量の少ない以下の近似値を使用している。

- 現画像と参照画像のSAD(Sum of Absolute Difference)
  - 動きベクトルの大きさに応じ予め用意されたコスト値
- 前者は、符号化歪によって生じる画像の誤差を簡易計算によって近似的にあらわしたものに相当する。SADとは式(4)で表される演算であり、現画像と参照画像のマクロブロックの各画

素の差分絶対値和となる。これらの評価値は、どちらも実際に符号化せずに得られるため高速な動作が可能となる。しかし、これらの値と実際に符号化し得られる値との相関が低いケースでは、動きベクトルの精度が低下するという課題がある。

$$SAD = \sum_{i,j=0}^{Blocksize} (f_{cur}(i,j) - f_{ref}(i+mv_x, j+mv_y)) \quad (4)$$

## 4. 提案手法

提案手法では動きベクトルの評価方法としてRDOを用いる。動きベクトルのような候補数の多いものにRDOを適用すると演算量増加の問題が発生する。本稿では、探索階層において1/2画素、1/4画素の探索にのみRDOを適用することによって演算量増加の問題を解決する。

提案手法では復号時に実際得られるデータに基づいて高精度な探索処理を行うために以下の評価値を使用する。

- $D$ :現画像と復号画像のSSD(式(3))
- $R$ :実際に符号化した場合に発生する符号量

上記の $D, R$ の値から式(1)によって求まる $RDcost$ を評価値とする。 $\lambda$ は量子化パラメータQPから求まる式(2)を用いる。また小数精度の探索アルゴリズムとしてDiamond Searchを用いる。

## 5. 性能評価

提案手法の有効性を検証するために、比較評価を行う。表1に示す条件で評価実験を行う。またマクロブロックタイプの選択には全ての手法でRDOを適用する。これは性能評価を行う際に動き検出のみの影響を評価するためである。評価手法として総演算時間、BD-Bitrate(Bjontegaard's Deltan Bitrate)、レート歪特性を用いる

表1 実験環境

フレーム数	100
参照フレーム数	1
動き探索範囲	16
GOP	IPPP...
マクロブロックタイプ	i16x16, i4x4 p16x16, p16x8, p8x16, p8x8
使用シーケンス	akiyo foreman salesman stefan suzie
解像度	352 x 288
CPU	Intel Core2Duo E6750 2660[MHz]
Encode soft	ISEAVC [11]

### 5.1 階層別RDO適用の評価

まず3章で述べた探索階層において、どの階層にRDOを適用するのが良いか評価を行う。表2に示す手法について比較評価を行う。

表 2 階層別 RDO 適用 比較手法

	整数精度探索		小数精度探索	
	評価関数	アルゴリズム	評価関数	アルゴリズム
手法 1	SAD	Full	SAD	Diamond
手法 2	SAD	Full	RDO	Diamond
手法 3	RDO	Full	RDO	Diamond

QP=26 時の総演算時間を表 3 に、BD-Bitrate を表 4 に、レート歪特性を図 1, 2 に示す。表 3 から、手法 3 は手法 1 に比べ総演算時間が約 40~60 倍となっているのに対し、手法 2 は約 1.3~1.5 倍程度となっている。また表 4 から手法 2, 3 共に手法 1 に対して 1.7~3.1% 符号化効率が向上している。一方で総演算時間は手法 3 が圧倒的に増加しているが、必ずしも手法 3 の方が手法 2 よりも符号化効率が良いとは限らない。これは整数精度探索で選択された動きベクトルが局所解となる可能性があるためと考えられる。よって総演算時間と符号化効率の観点から RDO を適用する階層は小数精度のみで良いと考えられる。

表 3 階層別 RDO 適用 演算時間 [s] QP=26

	akiyo	foreman	salesman	stefan	susie
手法 1	14.26	15.32	14.75	15.23	14.70
手法 2	18.75	21.20	20.26	22.39	20.12
( $\Delta$ [%])	(31.5)	(38.4)	(37.4)	(47.1)	(36.9)
手法 3	664.45	775.73	845.75	950.20	651.62
( $\Delta$ [%])	(4559.5)	(4963.5)	(5633.9)	(6139.0)	(4332.8)

表 4 階層別 RDO 適用 手法 1 に対する BD-bitrate [%]

	akiyo	foreman	salesman	stefan	susie
手法 2	-1.96	-3.11	-2.57	-1.71	-2.77
手法 3	-2.26	-3.11	-2.65	-1.82	-2.72

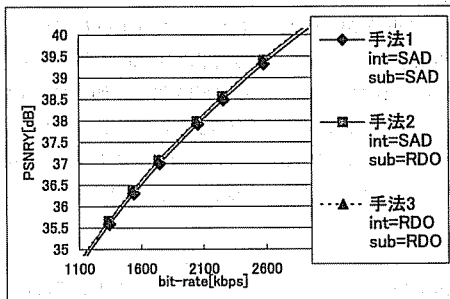


図 1 階層別 RDO 適用 RD 曲線 (stefan)

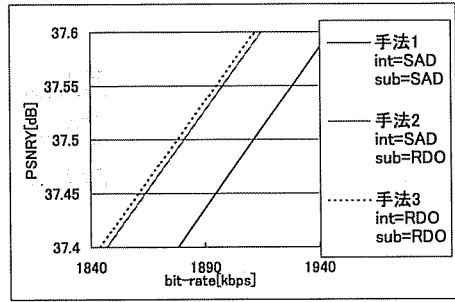


図 2 階層別 RDO 適用 RD 曲線 拡大 (stefan)

## 5.2 小数精度探索範囲の評価

次に小数精度探索範囲について評価を行う。表 5 に示す手法について比較評価を行う。ただし、

HM : 1/2 画素精度探索時 Diamond Search の最大回数

QM : 1/4 画素精度探索時 Diamond Search の最大回数

とする。

表 5 RDO 小数精度探索範囲 比較手法

	整数精度探索		小数精度探索	
	評価関数	アルゴリズム	評価関数	アルゴリズム
全手法	SAD	Full	RDO	Diamond
	HM		QM	
手法 1	1		1	
手法 2	3		3	
手法 3	5		5	

QP=26 時の総演算時間を表 6 レート歪特性を図 3, 4 に示す。手法 2, 3 は手法 1 に比べて少ない総演算時間増加にとどまっている。これは多くの場所で Diamond Search の打ち切りが発生している為と考えられる。また最大回数を 3 回から 5 回に増やした場合においても符号化効率の向上があまり見られない。よって最大回数は 3 回が妥当なラインであると考えられる。

表 6 RDO 小数精度探索範囲 演算時間 [s] QP=26

	akiyo	foreman	salesman	stefan	susie
手法 1	18.15	19.81	19.46	21.14	19.00
手法 2	18.75	21.20	20.26	22.39	20.12
手法 3	18.71	21.18	20.26	22.37	20.26

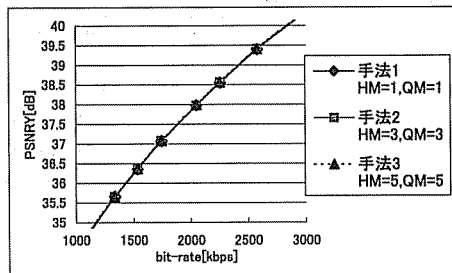


図 3 小数精度探索範囲 RD 曲線 (stefan)

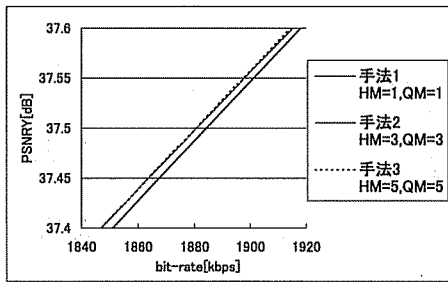


図4 小数精度探索範囲 RD 曲線 拡大 (stefan)

### 5.3 整数精度探索における高速アルゴリズム適用の評価

次に整数精度探索に高速アルゴリズムを適用した場合の小数画素 RDO 探索の有効性について評価を行う。具体的には、表7に示す手法について比較評価を行う。

表7 整数精度探索における高速アルゴリズム適用 比較手法

	整数精度探索		小数精度探索	
	評価関数	アルゴリズム	評価関数	アルゴリズム
手法1	SAD	Full	SAD	Diamond
手法2	SAD	UMH	SAD	Diamond
手法3	SAD	UMH	RDO	Diamond
		HM		QM
全手法	3		3	

QP=26 時の総演算時間を表8に、BD-Bitrateを表9に、レート歪特性を図5、6に示す。

手法2と手法3を比較すると、表8より小数精度にRDOを適用することで演算量は増加しているが、表9より1.5%~3.2%程度符号化効率率は向上しており、符号化効率の改善という点で、高速化アルゴリズムの小数精度探索にRDOを適用することは有効であると考えられる。

次に手法1に対する、手法2、3の結果を考える。表8より手法2は63.8%~70.3%、手法3は16.0%~39.9%削減しており、総演算時間に関しては両手法ともに削減できている。手法2は高速化アルゴリズムの適用により符号化効率率が低下しているのに対し、手法3は符号化効率率においても1.22%~2.97%向上しており、優れた探索精度を持つと考えられる全探索に対して実行時間を削減しつつ符号化効率の向上に成功している。

表8 整数精度探索における高速アルゴリズム適用 演算時間 [s] QP=26

	akiyo	foreman	salesman	stefan	susie
手法1	14.26	15.32	14.75	15.23	14.70
手法2	4.23	5.15	4.68	5.51	4.73
(Δ[%])	(-70.3)	(-66.3)	(-68.2)	(-63.8)	(-67.8)
手法3	8.57	11.42	10.34	12.78	10.20
(Δ[%])	(-39.9)	(-25.4)	(-29.8)	(-16.0)	(-30.6)

表9 整数精度探索における高速アルゴリズム適用 手法1に対するBD-bitrate [%]

	akiyo	foreman	salesman	stefan	susie
手法2	0.17	0.46	0.11	0.22	0.19
手法3	-1.85	-2.80	-2.51	-1.22	-2.97

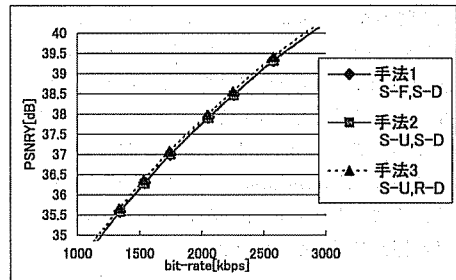


図5 整数精度探索における高速アルゴリズム適用 RD 曲線 (stefan)

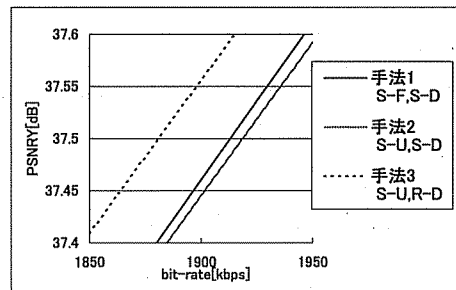


図6 整数精度探索における高速アルゴリズム適用 RD 曲線 拡大 (stefan)

## 6. まとめ

本稿では小数精度の動き検出においてRDOに基づくH.264動き検出手法を提案した。評価結果より、従来手法であるSAD方式に比べ31.5%~47.1%の実行時間の増加にて、1.71%~3.11%のビット量削減を実現した。また提案手法と整数精度の高速探索アルゴリズムを組み合わせるにより、16.0%~39.9%の実行時間を削減させると共に、1.22%~2.79%のビット量削減を実現した。

## 文 献

- [1] ISO/IEC 14496-10, "Advanced Video Coding," ITU-T Recommendation H.264,(2003).
- [2] 長島 新一, 松田 一朗, 伊東 晋, "適内挿フィルタを用いた非整数画素精度動き補償方式", 信学技報. Vol.104, No.752(20050318), pp. 43-48, (Mar 2005).
- [3] Joint Video Team(JVT), "JVT Reference Software, JM14.1", <http://iphome.hhi.de/suehring/tml/>, (Aug 2008).
- [4] J.Y.Tham, S.Ranganath, M.Ranganath, and A.A.Kassim, "A novel unrestricted centerbiased diamond search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 8, pp. 369-377, (Aug. 1998).
- [5] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation", IEEE Trans.

- on Image Processing, vol. 9, no. 2, pp. 287-290, (Feb 2000).
- [6] J.H. Lee, and N.S. Lee, "Variable block size motion estimation algorithm and its hardware architecture for H.264/AVC", in Proc. ISCAS2004, p.741-744, (May 2004).
  - [7] S.Y.Yap, and J.V.McCanny, "A VLSI architecture for variable block size video motion estimation", IEEE Trans.Circuits and Systems, vol. 51, pp 384-389, (Jul 2004).
  - [8] C.M. Ou, C.F. Le, and W.J. Hwang "An efficient architecture for H.264 variable block size motion estimation", IEEE Trans. Consumer Electronics, vol. 51, pp 1291-1299, (Nov 2005).
  - [9] T.C. Chen, Y.W. Huang, and L.G. Chen "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC", in Proc. ICASSP, pp V-9-V-12, (May 2004).
  - [10] Yu-Wen Huang, Tu-Chih Wang, Bing-Yu Hsieh, and Liang-Gee Chen, "Hardware architecture design for variable block size motion estimation in MPEG-4 AVC/JVT/ITU-T H.264," Proceedings of 2003 International Symposium on Circuits and Systems (ISCAS 03), Vol.2, pp. II-796 - II-799, (May 2003).
  - [11] 橋本 亮司, 藤田 玄, 尾上 孝雄, "H.264 符号化における演算量動的割当ての一手法", 第 21 回デジタル信号処理シンポジウム, D8-1, (Nov 2006).