

卒業論文

題目

近傍動き探索用SIMD命令を多用
する
動き検出アルゴリズムの提案

指導教員

近藤 利夫 教授

2015年

三重大学 工学部 情報工学科
計算機アーキテクチャ研究室

箕浦 祐貴 (411849)

内容梗概

近年、高精細化の動画像符号化の必要性が増加しており、従来比 2 倍の圧縮性能を備えた符号化復号化規格 H.265 が標準化されている一方で、処理量が増えることでその性能を引き出すことが非常に困難になっている。特に、符号化処理の大半を占める動き探索処理では、速度要求に応えられていないため圧縮率に大きく影響する探索精度を犠牲にせざるを得ない状況となっている。その速度要求に応えるために、当研究室では動き探索用 SIMD データパスの開発を進めている。このデータパスは高精度・高効率を両立する動き探索処理への対応を目指している。しかし、未だこのデータパス専用の動き探索アルゴリズムが開発されていないため、その性能を引き出すことが困難である。本研究ではこの動き探索用 SIMD データパスを効率よく利用する専用の動き探索アルゴリズムの実装を行った。このアルゴリズムを適用することにより、テストシーケンスの符号化において、従来の TZSearch と比較し、圧縮率と画質の劣化を抑えつつ、ベクトルレジスタ内のデータを再利用する回数を 20% 増加させ、演算回数を 80% 削減させることに成功した。

Abstract

Recently, in the need for video encoding of high definition enhancement, although video coding standard H.265 which doubles the compression ratio compared with H.264 by increasing in the amount of processing, it is difficult to fully bring out its compression performance. Therefore, in the motion estimation processing occupying most of the encode processing, a current practical encoder has to sacrifice search accuracy because it cannot meet both speed and accuracy demands. In our laboratory, SIMD Data Path for Motion Estimation has been designed to meet the two demands. The Data Path aims to realize highly efficient one. However, it is difficult to bring out the performance, because the motion estimation algorithm for the SIMD Data Path has not developed yet. This paper developed highly efficient motion estimation algorithm for the SIMD Data Path. As a result, the rate of using local data increases by about 20%, and the amount of calculation in absolute distance decreases by about 80%.

目次

1	はじめに	1
1.1	研究背景	1
1.2	研究目的	2
1.3	各章の内容	3
2	動き探索と SAD 演算, 動き探索用 SIMD データパスの概要	5
2.1	動き探索と SAD 演算	5
2.2	動き探索用 SIMD データパス	5
3	従来探索法の適用上の問題点	8
3.1	8点スクエア探索	8
3.2	拡大型ダイヤモンド探索	8
3.3	ラスタ探索	10
4	提案手法	12
4.1	Small unsymmetric cross search	12
4.2	ラスタ探索の最適化	13
4.3	サブサンプリングの適用	14
4.4	提案手法フローチャート	15
5	各動画像における性能評価	18
5.1	評価方法	18
5.2	評価結果	19
5.3	考察	19
6	おわりに	22
6.1	まとめ	22
6.2	今後の課題	23
	謝辞	24
	参考文献	24

目 次

2.1	MPSADBW 命令における並列演算が可能な探索点	6
2.2	0~8 番のベクトル用レジスタに格納される参照画像	6
3.3	8 点スクエア探索探索一回分の探索点	9
3.4	拡大型ダイヤモンド探索一回分の探索点	10
3.5	ラスタ探索の探索点	11
4.6	SUCSearch 一回分の探索点	12
4.7	サブサンプリングを適用したブロック	14
4.8	提案手法フローチャート	16

表 目 次

4.1	エンコード一回あたりの SUCSearch の実行回数の合計の比較	17
5.2	動画像のエンコード結果 1	21

1 はじめに

1.1 研究背景

近年の 4K,8K などの動画像高精細化が進んでおり [1],1 フレームに含まれる画素数が増加してきたことで圧縮符号化の負荷が大幅に増え,高性能の PC を用いたとしても,動画像の圧縮に多大の時間を要したり,劣化無しでの圧縮が困難となっている.それによって動画再生の快適性やストリーミングのリアルタイム性が損われる事は動画配信の大きな制約となるため,符号化時間短縮と圧縮率改善の両立の要求が一層高まっている.この圧縮符号化の処理量増大の主因となっているのは,参照画像内の参照ブロックと符号化対象画像の符号化対象ブロックとの間でブロックマッチングを繰り返す動き探索により膨大な処理量を要する動き検出処理である.しかし,この動き検出の膨大な処理量は,増大する一方の状況にある.圧縮率向上に向けて,従来から動き補償に様々な改善が施されてきているからである.実際,H.262 以降で用いられている前後の画像を参照する双方向用動き予測や,H.264 で導入された 7 種類の異なるブロックサイズを使い分ける可変ブロックサイズ,H.265 でのブロックサイズの多用化等は,大幅な処理量増大の要因となっている.この結果,未だに動き検出が符号化の処理量の大半を占める点は変わらず,PC によるスムーズな動画像圧縮

の実現には動き検出の高速化が必要不可欠となっている。

これに対し当研究室ではハードウェア対応の観点から、汎用プロセッサにおける符号化処理のより効率的な処理の実現をめざし、動き探索高速化に的を絞った SIMD 併用型汎用データパスの開発を進めている。このデータパスは、SIMD のベクトルレジスタ内に収まる ± 4 の範囲を効率的に探索する機能を備えている。しかし、従来のソフトウェア向け動き検出アルゴリズムのように拡大型ダイヤモンドサーチやヘキサゴナルサーチといった疎らな探索を用いるのでは並列処理の仕組みが活かされず、高速化の効果が得られない。一方、全探策やラスタ探索といったハードウェア向けの動き検出アルゴリズムでは、探索点の数が多過ぎて、並列処理が効いても、高速の動き検出は実現されない。

1.2 研究目的

そこで、本研究では動き探索の探索パターンのバリエーションを SIMD のベクトルレジスタ内に収まるものに制限する条件のもと、動き探索の探索位置や演算に用いる画素の選定などの見直しによって、高探索効率、低演算量、参照画像データの高再利用率の 3 つの要求を満す動き探索アルゴリズムの実現を目指した。提案アルゴリズムでは転送オーバーヘッド低減の

ため, HM 参照ソフトウェアに標準で搭載されている TZSearch[2] の探索点においてデータ読み込みが過剰になっている部分を調べ, 探索点の大幅な低減を図っている. 評価実験では提案手法と TZSearch に対する 3 種類の動画画像のエンコードを行い結果を比較することで, 提案アルゴリズムの有効性を明らかにしている.

1.3 各章の内容

- 1 章では近年の動画画像高精細化とそれによって生じる問題を解消する為の当研究室で行っている研究を背景とした本研究の目的を述べる.
- 2 章では本研究を行うにあたって理解が必要である知識についての説明を述べる.
- 3 章では従来の探索法を追跡型探索に適する SIMD データパスに用いた場合の問題点について言及, 続く 4 章で問題点を受けての提案手法を述べる.
- 5 章では本研究の評価方法と結果, それらを踏まえた考察を述べる.
- 6 章では本研究に関連した研究を述べる.

- 7章では研究によって得られた事で明かになった事と今後の展望と課題を述べる.

2 動き探索と SAD 演算, 動き探索用 SIMD データパスの概要

本章では動き探索と当研究室で開発を進めている動き探索用 SIMD データパスについて, 概要を述べる.

2.1 動き探索と SAD 演算

動き探索は, 符号化対象画像内の符号化対象ブロックと参照画像の参照ブロックを用いてブロックマッチングを行い, 画素の輝度値の差の絶対値の合計を求める SAD(Sum of Absolute Differences) 演算処理により, 数値化されるブロック間の類似度がより高いブロックを検出する処理である.H.265 では動画像の符号化を最大 64×64 画素から最小 4×8 画素, または 4×8 画素までの可変サイズのブロックのそれぞれに対して行う. 高効率な動き探索法とはより少ない探索点でより類似度の高いブロックを検出する探索法である.

2.2 動き探索用 SIMD データパス

従来の SIMD 命令セットの MPSADBW 命令は 4×1 画素幅の符号化対象画像に対し, 図 2.1 の 1 から 8 が示す 4×1 画素幅の参照画像を水平方向に 1 画素ずつシフトした 8 座標の SAD 演算を並列に実行出来る. しかし

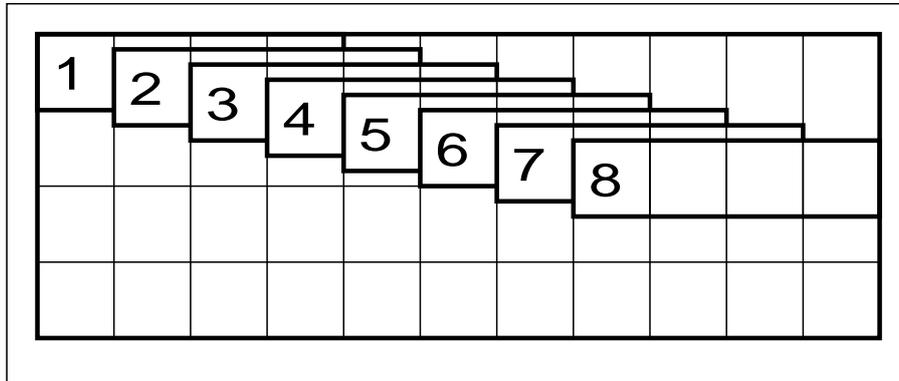


図 2.1: MPSADBW 命令における並列演算が可能な探索点

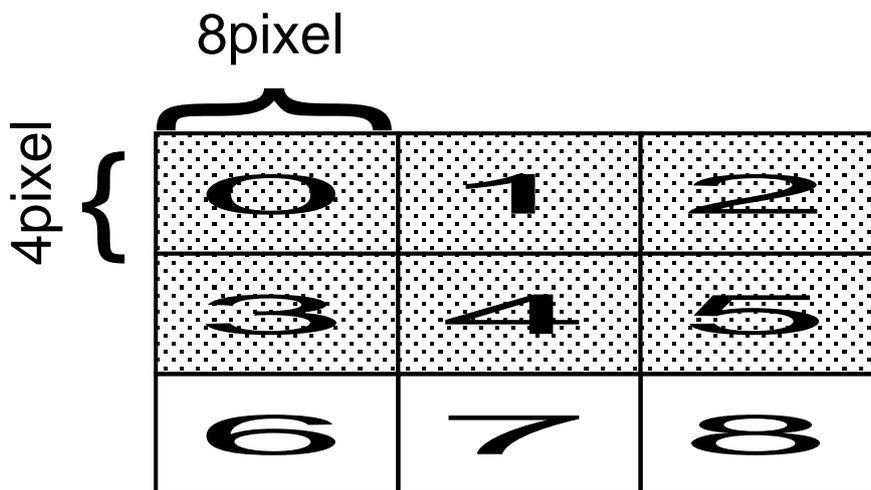


図 2.2: 0~8 番のベクトル用レジスタに格納される参照画像

水平方向に連続する探索点のみが並列に実行可能というだけでは、ソフトウェア処理向きアルゴリズムにおいて広く用いられている追跡型探索に対する適合性が低く、高効率な動き探索処理に対応できない。そこで当研究室では、この追跡型探索を高効率で実行できるように、参照画像中の任

意のマッチング対象サブブロック (8×4 画素サイズ) のアクセスを可能とするベクトル用のレジスタファイルを備えた SIMD 併用型汎用データパスの開発を進めている。そのベクトル用レジスタファイルを図 2.2 に示す。この図に示すように 0 番から 5 番に SAD 演算用の参照画像、残りの 6 番から 9 番に次の縦方向の走査時 SAD 演算に必要な参照画像を格納すると、レジスタファイル内に画像データが存在していれば 8×4 画素単位で並列アクセスが可能となる。このアクセス機能により実行可能になるベクトルレジスタに格納される参照画像と 8×4 画素の符号化対象画像との間の並列ブロックマッチングを繰り返すことで、固定的な近傍探索パターンを効率的に処理できる。また 8×4 よりも大きいブロックのマッチング処理は、ブロックを 8×4 のサイズに分割したそれぞれでブロックマッチングを行い、SAD 演算処理の結果を足し合わせる事で行う。

3 従来探索法の適用上の問題点

この章では従来の探索法を追跡型探索に適する SIMD データパスに用いた場合の問題点を挙げる。従来の探索法としては、近年用いられている代表的な探索中心点を最良点に移行する追跡型探索。そして探索中心を固定したまま範囲内の固定点を探索する探索法から選定する。

3.1 8点スクエア探索

探索中心に隣接する8点からなるスクエアパターンに対する探索を繰り返す追跡型探索である。データの再利用性が高いためハードウェア向けの探索法といえるものの、探索遠隔点の探索に対応しないため速い動きの動画像に対し圧縮率の悪化や、追跡する距離が延びることで SAD 演算量の増加といった問題がある。

3.2 拡大型ダイヤモンド探索

拡大型ダイヤモンド探索は TZSearch で用いられている速度と精度に優れたソフトウェア向けの代表的探索法で、図 3.4 にあるように、中心点から 2 のべき乗距離の位置に探索点を取るダイヤモンド状に広がる探索パターンで追跡する探索法である。この探索法の問題点は、遠隔の探索点が疎ら

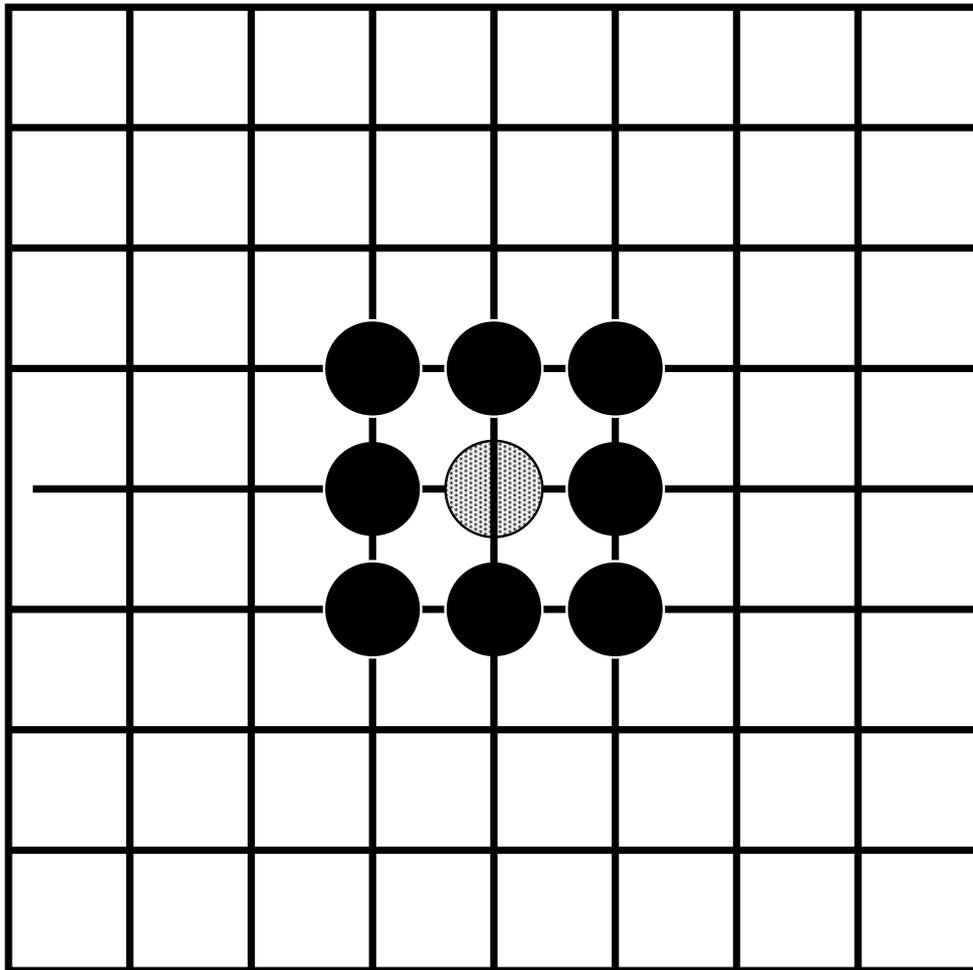


図 3.3: 8点スクエア探索探索一回分の探索点

になるためデータの再利用率が低くなることである。開発中のSIMD 併用型データパスでは、2.2節で扱ったベクトルレジスタ 8×4 画素 3×2 本分という格納範囲に収まりきらず、並列処理が不可能な上、ベクトルレジスタへの読み込みが頻発することになる。また、動き探索において検出される点は探索を開始した点の近傍に存在する割合が多いことから、遠隔点の探索

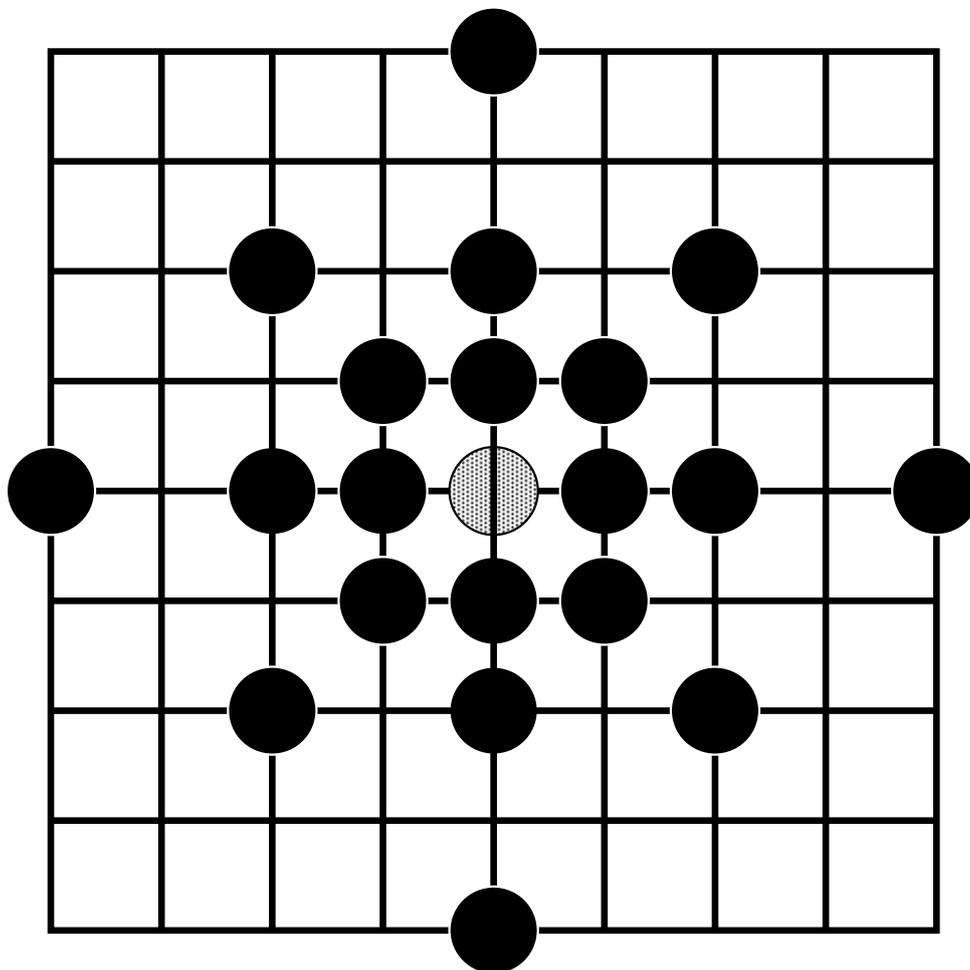


図 3.4: 拡大型ダイヤモンド探索一回分の探索点

点が多いことで探索点が過剰になる傾向にある。

3.3 ラスター探索

ラスター探索は探索範囲で定まったパターンを行う探索法として TZSearch でも用いられている探索法である。図 3.5 に  で指すように、指定され

た探索範囲を一定画素毎にラスタ順に探索してゆく。ラスタ探索には探索すべき点が抜けることによる探索精度低下を防止できる利点があるものの、データ再利用率が低い上に探索点数が多い分 SAD 演算量と画像データ読み込み回数が増加するため、探索効率が低下する問題がある。

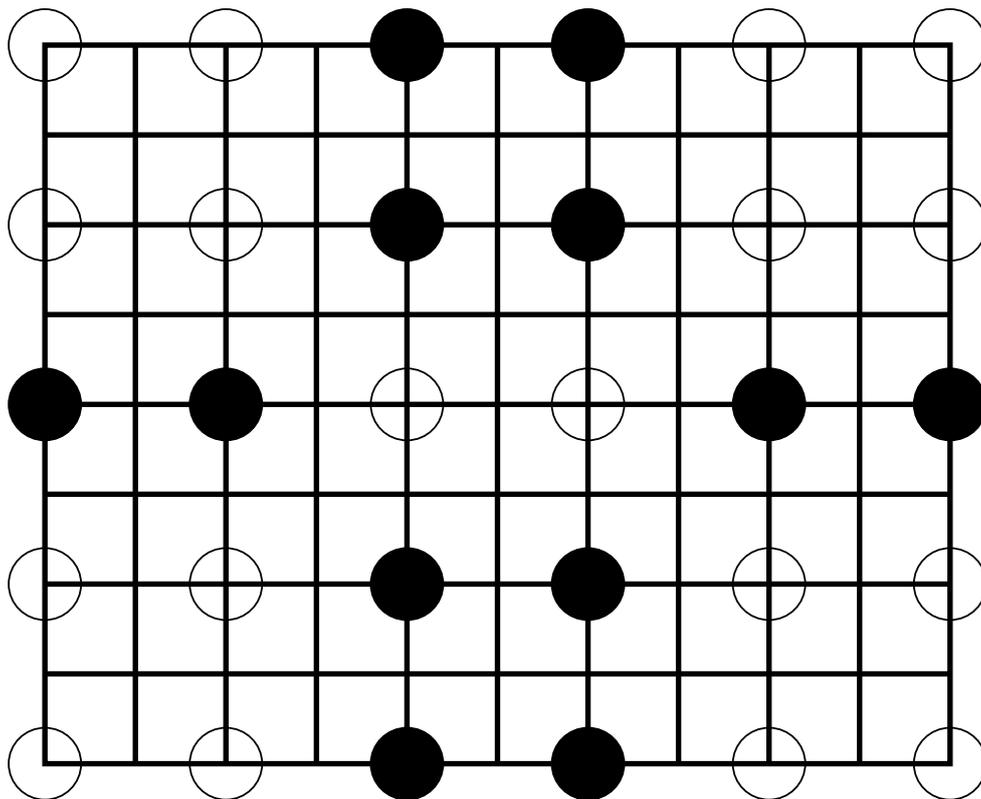


図 3.5: ラスタ探索の探索点

4 提案手法

この章では TZsearch における探索点パターン及び間引き方を 2.2 節で示した動き探索用 SIMD データパスに最適化させる方針で改良を重ねた提案アルゴリズムについて述べる。最適化は、動き探索用 SIMD 命令セットを用いた時、並列処理がより多く行える探索パターンになっているか、またその変更によって著しく圧縮率の低下、画質の劣化をしないかという指針で行う。

4.1 Small unsymmetric cross search

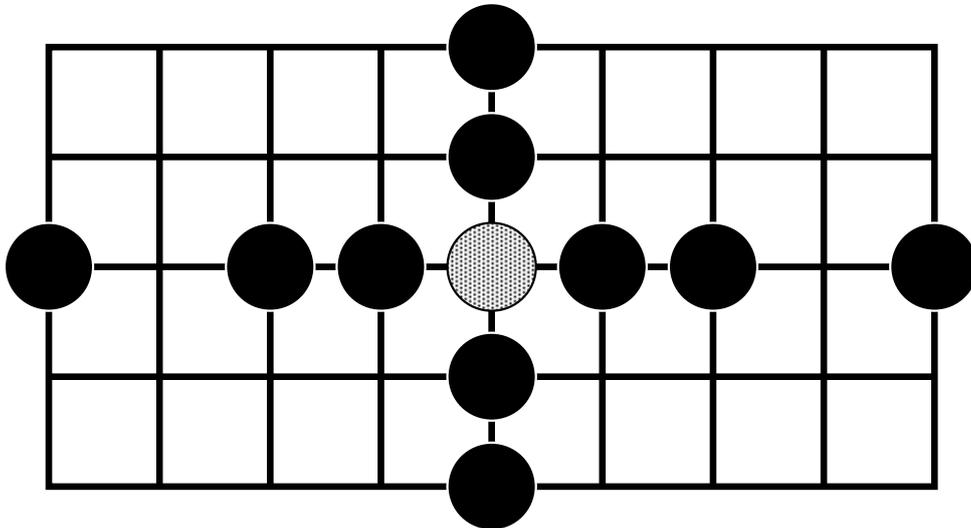


図 4.6: SUCSearch 一回分の探索点

従来の拡大型ダイヤモンド探索をベクトルレジスタ内に収めただけで

は、無駄な探索点が多過ぎる。またスクエア探索ならばベクトルレジスタに収める事は出来るが、探索開始点周辺しか検出が出来無い。それらの問題と動画像では一般的に垂直方向と比較し水平方向の動きが多いことを考慮して、横の動きに強く、かつベクトルレジスタに収まる事で並列処理可能な SUCSearch (Small Unsymmetric Cross Search) を提案する。SUCSearch は図 4.6 に示すように、拡大型ダイヤモンド探索を ± 2 まで拡大した形から斜めの点を除き、 ± 4 の水平方向の点を加えた探索パターンを用いる追跡型探索法である。初回の SUCSearch で遠隔点を検出した際に実行するラスタ探索後に SUCSearch を 4 回以上行っても符号化効率の改善が見られなかったことから、最悪実行時間の確定のため、ラスタ探索後の SUCSearch の実行は高々 4 回とする。

4.2 ラスタ探索の最適化

正確な探索を行う為の探索法であるため、ラスタ探索そのものを取り除く等の大雑把な削減は行わず、探索における重要な点の選定をする事で最適化を目指す。重要視したのは演算回数の増加とそれに応じるデータ読込回数の増加を最低限に抑制する事である。そこで SUCSearch と組合せて探索可能な部分を SAD 演算の削減対象であるとして、演算回数と圧縮

率が悪化しないよう探索点の選定を行い, 図 3.5 の のように, 探索範囲の上下左右が重点的に探索されるようラスタ探索の最適化を施した. ここで は探索点から除かれる点を示す.

4.3 サブサンプリングの適用

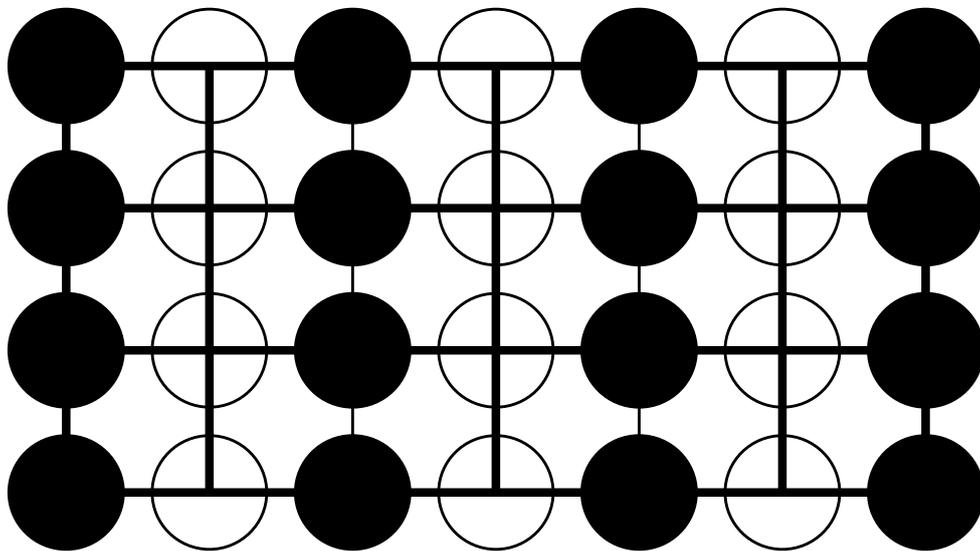


図 4.7: サブサンプリングを適用したブロック

ブロックマッチングの際に, 図 4.7 の の画素のみで画素間の差分値をとる 2 画素精度のサブサンプリングを適用する. これを適用する理由として SUCSearch はベクトルレジスタファイルのサイズ制約により, 拡大型ダイヤモンド探索と比較すると探索範囲が狭いことが挙げられる. そこでサブサンプリングにより画素を間引く事で, ベクトルレジスタに入る水平

方向の領域を2倍にし、ラスタ探索を含んだ全体の演算回数を減らしつつ探索精度を上げられるようにする。一方で水平方向ライン毎に間引くことで、垂直方向への探索範囲を広める事も可能であるが、動画像が水平方向に動く場合が多い事を考慮すると、水平方向に範囲を広める、即ち垂直方向に間引く方が良いと推定されるからである。

4.4 提案手法フローチャート

本研究で比較評価に用いる提案手法のフローチャートを、図4.8に示す。探索開始点は周囲のブロックから動きを予測した点である。その点を探索中心として一回目のSUCSearchを実行する。一回目のSUCSearchで検出した点が探索中心から距離4以上にあった場合続けてラスタ探索を実行する。これは距離の離れた点が検出された場合、さらに離れている点により良い点のありうることを考慮した結果である。ラスタ探索後はラスタ探索の検出点に探索中心を移行して、SUCSearchの実行に移る。最初の距離が4未満であった場合も、検出点が探索中心でない限り同様にSUCSearchの実行へ移る。2回目以降のSUCSearchは探索中心を検出点に移行して、探索中心が検出点になるまで繰り返し実行する。ただし初回のSUCSearchで遠隔点を検出した際に実行するラスタ探索後にSUCSearchを4回以

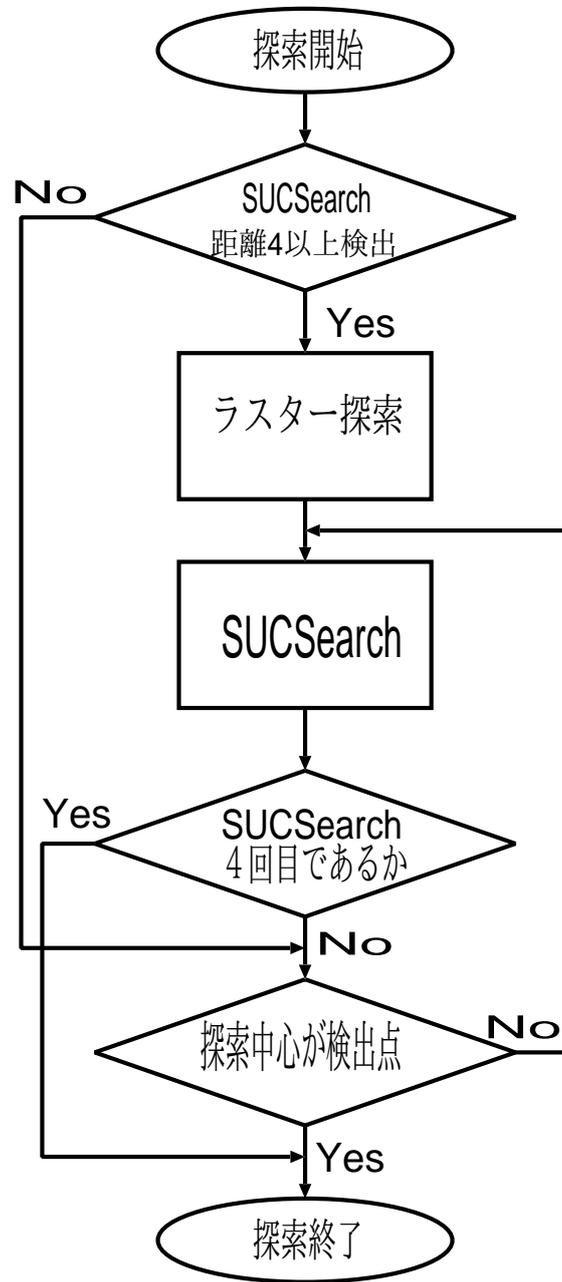


図 4.8: 提案手法フローチャート

上行っても符号化効率の改善が見られなかったことから、最悪実行時間の予測を容易にするため実行回数を高々4回としている。表 4.1 に示すように実験によって SUCSearch を繰り返す回数は 10000 回以内である。そのため全体からみて画質や圧縮率の結果に影響がないと考えた。表 4.1 で示す実験結果は HD プログレッシブ画像 park_joy(park)[3] を提案手法で 17 フレームをエンコードした場合に得られた結果である。SUCSearch によって探索中心が検出される、または 4 回目の SUCSearch が行われた時点での検出点が、その符号化対象ブロックに対する最終的な検出ブロックとなる。

表 4.1: エンコード一回あたりの SUCSearch の実行回数の合計の比較

	1 回	2 回	3 回	4 回	5 回
移動回数合計	1475269	321620	63282	14384	6034

5 各動画像における性能評価

この章では、本研究で行った実験の評価方法と評価結果を表 5.2 と共に説明と考察をする。

5.1 評価方法

拡大型ダイヤモンドサーチを用いた TZSearch(TZD) と拡大型ダイヤモンドサーチの代わりに 8 点スクエアサーチを用いた TZSearch(TZS) と、4 章で述べた探索法と改良点の機能追加を行った提案手法を HM 参照ソフトウェアである HM10.0 に実装した。比較に TZSearch を用いる理由として TZSearch は H.265 の標準ソフトウェアエンコーダで使用されており高い圧縮率と画質を両立している優れたアルゴリズムである事と、データ再利用率に優れた 8 点スクエアサーチを使用する事も可能であるので比較に適していると考えたためである。評価は、HD プロGRESSIVE 画像 7 種類 Park_Joy(park), speed bag(speed)[4], Tennis, Park_Scene, BasketballDrill(basket), Kimono, Mall のそれぞれの連続する 17 フレームをエンコードすることで行った。表 5.2 の比較内容は同一画質における発生符号量の差分を評価した BD-Rate(BD), AD 演算回数 (AD), 遠隔点の探索回数 (Far) の 3 つである。

5.2 評価結果

5.3 考察

BD-Rate は park と speed で TZSearch に対する劣化率が 1%未満と良好である。park は森の中を人物が歩き、それを水平方向に追って動く動画像である。speed は物体が高速に左右に動く動画像である。これら 2 つの動画像は細かな描写が多い一方で、動きの向きは一定のパターンがあると考えられ予測が容易であるため、提案手法による BD-Rate の悪化が少ないと考ええる。同じ理由で Park_Scene, Kimono, Mall も悪化が少ない。一方 tennis と basket では 1%以上の劣化が見られる。その理由として、tennis は人物の動きが複雑な動画像であるため動きの予測が当たり難いことが考えられる。例として映像は左に動いているが、人物の腕は左に動いている等の描写がある。最も BD-Rate の結果が優れているのは周囲のあらゆる方向を探索できる TZS であることから、提案手法では不規則な方向に動き、近傍点を検出しやすい動画像に弱い傾向があるといえる。

AD 演算回数は、探索点の選定と画素の間引きにより、提案手法を用いると各画像で約 80%の削減ができています。その理由として SUC サーチとラスタ探索時の探索点の選定、サブサンプリングの適用によるブロックあたりの演算回数の削減が挙げられる。SUC サーチと最適化したラスタ探

索では探索点を減らす事で探索回数を減らしている。サブサンプリングは水平方向 2 画素精度と 4 画素精度で実験を行ったが、4 画素精度で行った場合、探索精度の悪化から BD-Rate の結果が悪化したため、2 画素精度を適用している。2 画素精度のサブサンプリングにより、ブロックあたりの演算量は半減するため、AD 演算の結果への影響が最も大きいものであると考える。speed では BD-Rate の悪化が小さく、演算量を大幅に低減できている。このことについて、TZD での BD-Rate が最も少ないことから考察すると、提案手法は遠隔点と近傍点に検出点が散らばり、さらに水平方向への動きが多い動画像において優れているアルゴリズムであるといえる。しかし、tennis や basket のように上下方向の速い動きが多くなると動き検出の精度が低下する傾向が見られた。

遠隔点の探索数も、TZD と比べ $1/50$ 以下にまで低減できている。以上の評価結果より、演算量も各画像における PSNR 値の大きな劣化が無く演算量を大幅に低減できており、研究目標を満たせたといえる。

表 5.2: 動画像のエンコード結果 1

	case	TZD	TZS	提案手法
park	BD(%)	100.00	100.03	100.36
	AD($\times 10^7$ 回)	11836.68	10985.38	2614.90
	Far(回)	283804634	-	4211940
speed	BD(%)	100.00	100.02	100.49
	AD($\times 10^7$ 回)	29052.46	27504.10	3949.84
	Far(回)	267176371	-	4266073
tennis	BD(%)	100.00	99.44	100.37
	AD($\times 10^7$ 回)	25835.84	23696.45	3224.08
	Far(回)	317265995	-	5417234
basket	BD(%)	100.00	-	102.70
	AD($\times 10^7$ 回)	27618.34	26028.12	4959.33
	Far(回)	38104021	-	436871
kimono	BD(%)	100.00	-	100.55
	AD($\times 10^7$ 回)	17388.02	15505.52	2756.47
	Far(回)	336227218	-	4674681
park_sceane	BD(%)	100.00	-	100.91
	AD($\times 10^7$ 回)	9894.6	9188.66	2364.05
	Far(回)	172078816	-	3440409
Mall	BD(%)	100.00	-	100.49
	AD($\times 10^7$ 回)	21307.20	19917.84	4715.13
	Far(回)	35251824	-	393573

6 おわりに

この章では研究の結果の概要とその課題について今後の研究で改善するための考察を行う。

6.1 まとめ

本研究は従来の動き探索法を用いる場合の問題点を排除することによって、高効率動き探索用 SIMD 併用型汎用データパスに対しデータの再利用性と演算回数の面において適合性の高い動き探索アルゴリズムを提案した。提案手法は開発を進めている高効率動き探索用 SIMD 併用型汎用データパスを活かすのに適した、ベクトルレジスタ内に含まれる範囲を水平方向に重点的に探索する SUCSearch をベースとした探索アルゴリズムである。また多くのデータ読み込みが発生するラスタ探索の探索点を符号化効率の影響がでない範囲で削減、さらに演算量の削減と並列処理を行える範囲を拡大するために水平方向に 2 画素精度のサブサンプリングを実装した。その提案手法と従来の TZSearch を比較するために、3 種類の動画像を評価した結果、AD 演算回数とデータの再利用性において、提案手法が従来の TZSearch より優れていることを明かにした。特に park_joy と speed_bug でのビットレートの悪化が少なく、提案アルゴリズムは物体が細かい動き

や速い動きにも対応できる優れたアルゴリズムといえる。しかし tennis ではビットレートが大きく悪化しているため、物体の複雑な動きを正確に探索できていない可能性がある。

6.2 今後の課題

Tennis のような一部の画像のビットレートの悪化をどの様に抑えるかが今後の課題となる。この課題解決に向けて、サブサンプリングによる探索範囲を可変にする。垂直方向に特化した探索と水平方向に特化した探索など動画像に合わせて柔軟に使い分ける等の手法を試行し、あらゆる画像で高速に動き検出処理が可能となる SIMD 併用型汎用データパス向けの高効率動き検出アルゴリズムの実現をはかって行く。

謝辞

本研究を遂行するにあたり，日頃からご指導，ご助言いただきました近藤利夫教授，佐々木敬泰助教に感謝いたします．また様々な面でお世話になったコンピュータアーキテクチャ研究室の先輩方々や同輩方々に感謝の意を表します．

参考文献

- [1] 大久保, 鈴木, 高村, 中條, H.265/HEVC 教科書 , ISBN-978-4-8443-3468-2,2013 年 10 月.
- [2] X. Tang, D. Sheng-kui and C. Can-hui, “ An analysis of TZSearch algorithm in JMVC, ” IEEE Proc. ICGCS., pp.516-520, June 2010.
- [3] Xiph.org 「park_joy」, <https://media.xiph.org/video/derf/>,2015 年 2 月 3 日アクセス.
- [4] Xiph.org 「speed_bug」, <https://media.xiph.org/video/derf/>,2015 年 2 月 3 日アクセス.