修士論文

題目

SIMDデータパス向き高効率動き検出アルゴリズムの研究

指導教員

近藤 利夫 教授

2016年度

三重大学大学院 工学研究科 情報工学専攻
コンピュータアーキテクチャ研究室

箕浦　祐貴 (415M516)

内容梗概

高精細動画像の高圧縮符号化の要と期待される符号化復号化規格 H.265 が，2013 年に標準化された．この H.265 は，従来比 2 倍の圧縮性能を備えているものの，処理量が大幅に増えたことでその高い圧縮性能を十分に引き出すことが非常に困難になっている．特に，専用ハードウェアによる高速化が利用できないソフトウェアエンコーダでは，符号化処理の大半を占める動き検出の処理量を，精度を犠牲にしても抑えざるを得ない状況となっている．当研究室では，このような状況の解消に向け，汎用プロセッサの既存の SIMD データパスの置き換えを目指した高効率動き探索対応の SIMD データパスの開発を進めている．このデータパスは近年の高効率動き検出アルゴリズムの探索量低減の要の追跡型探索において繰り返される一定形状の小範囲複数点探索を高効率に実行する SIMD 命令を備えている．しかし，未だこのデータパスの動き探索機能を十分に引き出せる動き検出アルゴリズムは実現できていない．そこで，本論文ではこの動き探索用 SIMD データパスの探索機能を活かせるように，レジスタ内に納められる範囲の広い縮小画像の生成と，SIMD 命令を多用して，その範囲内での追跡型探索を繰り返す高効率の動き検出アルゴリズムを提案した．また，それを HEVC 参照ソフトウェア HM10.0 に組み込んで，HEVC テストシーケンスの符号化を行い，HEVC 参照ソフトウェア標準の TZSearch と比較し，圧縮率と画質の劣化を抑えつつ，参照画像のベクトルレジスタへの読み込み回数を平均で約 74%，演算回数を約 81%低減できることを示した．

Abstract

High efficiency video coding standard H.265 doubles the data compression ratio compared to H.264, however it is difficult to fully bring out its compression performance due to the expense of increased computational complexity, especially, in the motion estimation occupying most of the encoding time. A current practical encoder has to sacrifice coding efficiency because it cannot meet both the speed and accuracy requirements. Therefore, we had designed SIMD Data path with highly efficient local search function for motion estimation in our laboratory. However, the motion estimation algorithm for the SIMD Data path has not developed yet. This paper proposed highly efficient motion estimation algorithm frequently using SIMD instructions for small search range. As a result, the number of loading operations decreases by about 74% and the amount of absolute difference operations for block matching decreases by about 81% as compared to the motion estimation algorithm TZSearch used in the HEVC reference software.

目 次

1 まえがき **1**

2 動き探索，SAD 演算と動き探索用 SIMD データパス **3**
2.1 動き探索と SAD 演算 3
2.2 動き探索用 SIMD データパス 3

3 従来法適用上の問題点 **6**
3.1 従来の探索法 6
3.1.1 拡大型ダイヤモンドサーチ 6
3.1.2 8 点スクエアサーチ 7
3.2 サブサンプリングの問題点 8

4 提案手法 **10**
4.1 水平方向に拡張したクロスパターンによる追跡 10
4.2 複数の縮小画像を用いたブロックマッチング 10

5 実装 **15**
5.1 実装アルゴリズム 15
5.2 ベクトルレジスタシミュレータ 15

6 評価 **18**
6.1 評価方法 18
6.2 評価結果と考察 18

7 あとがき **21**

謝辞 **22**

参考文献 **22**

図 目 次

2.1 MPSADBW 命令 4
2.2 ベクトル用のレジスタファイル 5
2.3 ベクトル用のレジスタファイル 5
3.4 (A) 拡大型ダイアモンドサーチ (B)8 点スクエアサーチ . . 7
3.5 マッチングで間引く画素が異なる場合 9
4.6 SUC サーチの探索点 11
4.7 縮小画像を使って最適化 13
4.8 水平に一行ずつ間引いた縮小画像の生成 13
4.9 垂直に一列ずつ間引いた縮小画像の生成 14
4.10 格子状に間引いた縮小画像の生成 14
5.11 垂直方向のリプレイス発生例 16
5.12 水平方向のリプレイス発生例 17

表 目 次

3.1 TZSearch に 8 点スクエア探索を適用した結果 8
4.2 各縮小画像を用いた結果 12
6.3 各シーケンスの評価結果 20
6.4 評価項目の評価結果の平均 20

1　まえがき

近年 4K，8K などの動画像高精細化が進んでおり，フレームあたりの画素数が増加してきたことに伴って，圧縮符号化の負荷が大幅に増大している．その結果，動画像の圧縮に多大な時間を要したり，劣化無しでの圧縮が困難になっている．それによって，動画コンテンツの生成と編集が滞ると，動画の放送や配信の大きな制約となるため，圧縮符号化時間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処理量が膨大になる動き検出処理である．

しかし，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た結果，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．実際，MPEG-2 以降で用いられている前後の画像を参照する双方向動き予測や，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[1][2] での更なるブロックサイズの多様化等は，大幅な処理量増大の要因となってきた．その結果，ソフトウェアエンコーダでは，動き検出の処理量が，符号化処理の大半を占めるにもかかわらず，専用ハードウェアによる高速化が利用できないため，動き検出の精度を犠牲にしても処理量を抑えざるを得ない状況に陥っている．当研究室ではこのような状況の解消に向け，汎用プロセッサの既存の SIMD データパスの置き換えを目指した高効率動き探索に対応する SIMD データパスの開発を進めている [3]．このデータパスは，SIMD のベクトルレジスタ内に収まる 24x12 画素の小範囲（サブサンプリング無

しの画像）内で複数点のブロックマッチングを効率的に実行する機能を備えている．しかし, 拡大型の探索パターンを用いる UMHexagonS[4] や EPZS[5]，さらにラスター探索も組み合わせる TZSearch[6] など，従来のソフトウェア向けの動き検出アルゴリズムでは，探索点が疎らになる割合の高い手法をそのまま用いるので小範囲の高速探索機能の使用頻度が上がらず，並列化の効果が十分に得られない．一方で，全探索のようなハードウェア化が容易な動き検出アルゴリズムでは，並列化の効果が大きくても，探索点の多さから，動き探索の高速化は実現できない．

そこで，本研究では縮小画像を生成することにより，ベクトルレジスタに収まる範囲の拡張をしつつ，その範囲内に探索点のバリエーションを収める制限のもと，動き探索の探索位置や演算に用いる画素の選定の見直しによって，高探索精度，低演算量と，ベクトルレジスタ内のデータの再利用率向上の 3 つの要求を満たす動き検出アルゴリズムの実現を目指す．提案アルゴリズムでは転送オーバヘッド低減のため，HEVC 参照ソフトウェア HM に標準で搭載されている整数画素精度探索の TZSearch において参照画像データ読込みが過剰になる原因を探り，データ転送量の大幅な低減を図る．評価実験では提案手法と TZSearch に対する SD 画像と HD 画像を含めた 4 種類の動画像のエンコードを行い，その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する．

2　動き探索，SAD 演算と動き探索用 SIMD データパス

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することで，画素の輝度値の差分絶対値和 (SAD) を求め，数値化された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，ブロックサイズは最大で 64x64 で，最小は 8x4 若しくは 4x8 の可変サイズが適用されている．

2.2　動き探索用 SIMD データパス

図 2.1 に示すように従来の SIMD 命令セットである MPSADBW 命令 [7] では，4x1 画素幅の参照画像を水平方向に連続する 8 点分の SAD を同時に求めることができる．しかし，ソフトウェア処理向きの動き探索アルゴリズムで用いられる追跡型の探索パターンでは，垂直方向へ連続した探索パターンであったり，水平方向に連続する探索点が 3 程度かそれ以下に収まっており，高効率な探索パターンに対応していない．そこで当研究室では，高効率な追跡型探索に対応できるように，参照画像中の 24x12 画素の範囲内の任意の 8x4 画素のサブブロックを，アクセス可能とするベクトル用のレジスタファイルを備えた SIMD データパス (64 ビットスカラプロセッサと組み合わせたスーパースカラ構成) 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，8x4 画素のタイルデータを 0 から 8 番目までの 256 ビット (32 バイト) 幅のレジスタに格納する．各レジスタはバイト毎にアドレス修飾可能な

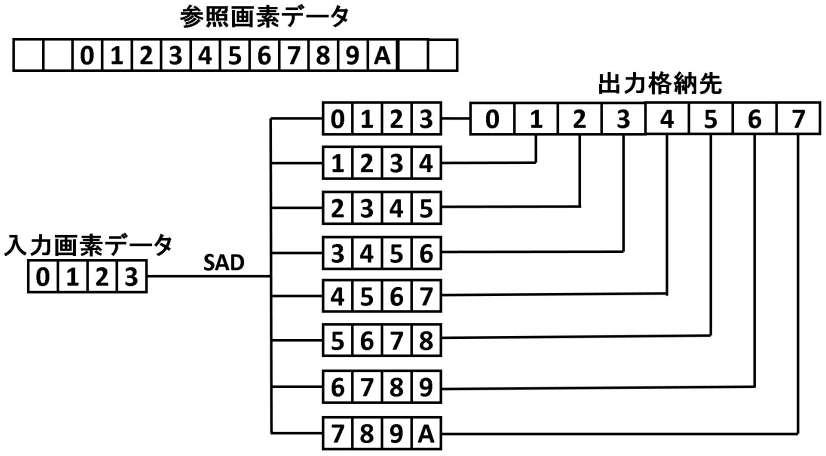

図 2.1: MPSADBW 命令

構成を採っており，適切な修飾を施すことで，図 2.3 に示すように 24x12 画素の参照画素の任意の 8x4 画素サイズのサブブロックが並列にアクセスされる．このアクセス機能により，タイル形式で格納された参照画像と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 24x12 画素の範囲内に収まる固定的な近傍探索が効率的に実行できる．また，8x4 よりも大きいブロックのマッチング処理は，8x4 画素サイズに分割したそれぞれで SAD 演算を行い，結果を足し合わせることで行う．

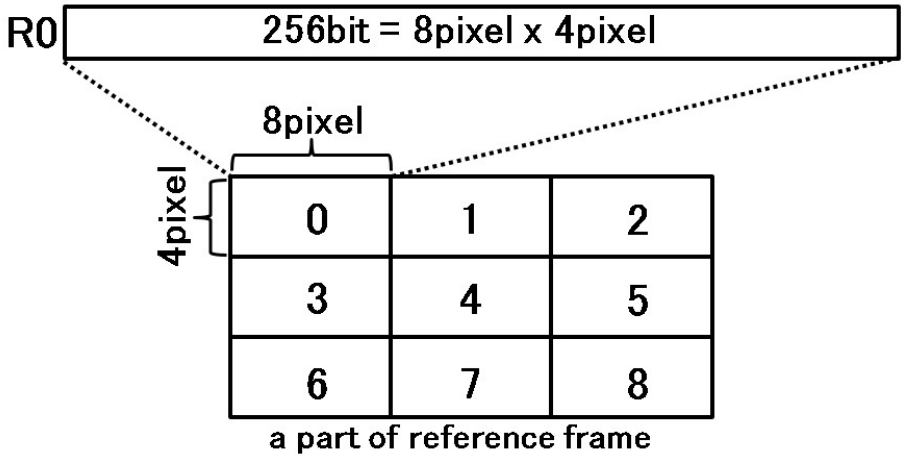

図 2.2: ベクトル用のレジスタファイル

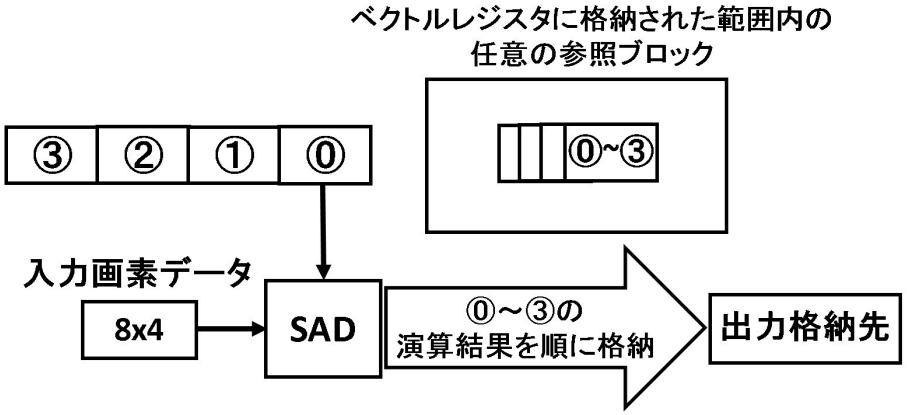

図 2.3: ベクトル用のレジスタファイル

3　従来法適用上の問題点

近年の高効率探索は追跡型探索をベースとする探索方法が広く用いられている．追跡型探索は特定の探索パターンを直前の探索の探索パターン内の SAD 最小位置 (SAD 極小位置と略記) に移動することで，探索範囲内の SAD 最小位置を検出する手法である．本節では，広範囲探索パターンである拡大型ダイヤモンド探索，小範囲探索の 8 点スクエア探索，H.265 で用いられる TZSearch アルゴリズムの問題点を示す.

3.1　従来の探索法

3.1.1　拡大型ダイヤモンドサーチ

拡大型ダイヤモンドサーチは TZSearch で用いられる速度と精度に優れたソフトウェア向けの探索方法である．図 3.4(A) に拡大ダイヤモンドサーチの探索点を示す．この探索法の問題点は，遠隔の探索点が疎ら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 データパスでは，2.2 節で扱ったベクトルレジスタの範囲である 24x12 画素の探索範囲に収まりきらず，参照画像の再利用率が低下するため，メモリからベクトルレジスタへのロードのオーバヘッドが増加し，探索効率が低下することになる．また，SAD 最小点は探索開始点の予測ベクトル近傍に存在する確率が高いため，遠隔点が SAD 最小点として検出される可能性が低くなるにも関わらず，遠隔点の探索割合が大きくなることから，探索効率低下の原因となっている．

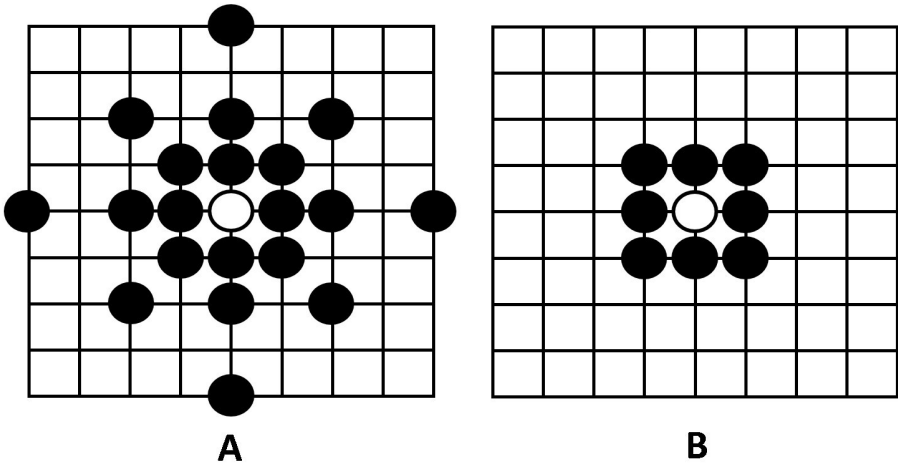

図 3.4: (A) 拡大型ダイアモンドサーチ (B)8 点スクエアサーチ

3.1.2 8 点スクエアサーチ

8 点スクエア探索は，図 3.4(B) に示すように探索中心に隣接する 8 点の探索点の SAD 極小点を次の探索中心点にして探索をする追跡型探索である．このような小範囲探索パターンは，データの再利用率が高く，転送ネックが生じ難いものの，遠隔点を探索しないため，速い動きの物体の探索に対応できず，圧縮率の低下を来す問題がある．表 3.1 は，TZSearch の拡大ダイヤモンド探索を 8 点スクエア探索に変更して，4 種類のシーケンスを実際にエンコードした際のベクトルレジスタへのロード回数 (load) と AD 演算回数 (AD) の低減率と BD-Rate の悪化率である．この結果が示すように，8 点スクエア探索は，探索点が探索中心の近傍の 8 点に集中しているので，load の低減率と AD 演算の低減率はそれぞれ平均 80.04%と 79.79%と良好である．一方で，BD-Rate の悪化率は平均で 2%を上回っており，実際に最適な SAD 最小点を捕捉できていないことが分かる．

表 3.1: TZSearch に 8 点スクエア探索を適用した結果

	load(%)	AD(%)	BD-Rate(%)
平均	80.0	79.8	2.2
最低	71.9	71.6	2.7

3.2 サブサンプリングの問題点

ブロックマッチングでは，SAD 演算で使用する画素を間引くサブサンプリングを用いて，演算量を低減できる．さらに，ベクトルレジスタファイルに格納できる画素の範囲を，間引いた分広く格納できる．しかし，間引いた画素がブロックマッチングの座標ごとに異なると，図 3.5 が表すようにマッチングの度に再度ベクトルレジスタのリプレイスが発生してしまい，その結果ロード回数が悪化する恐れがある．

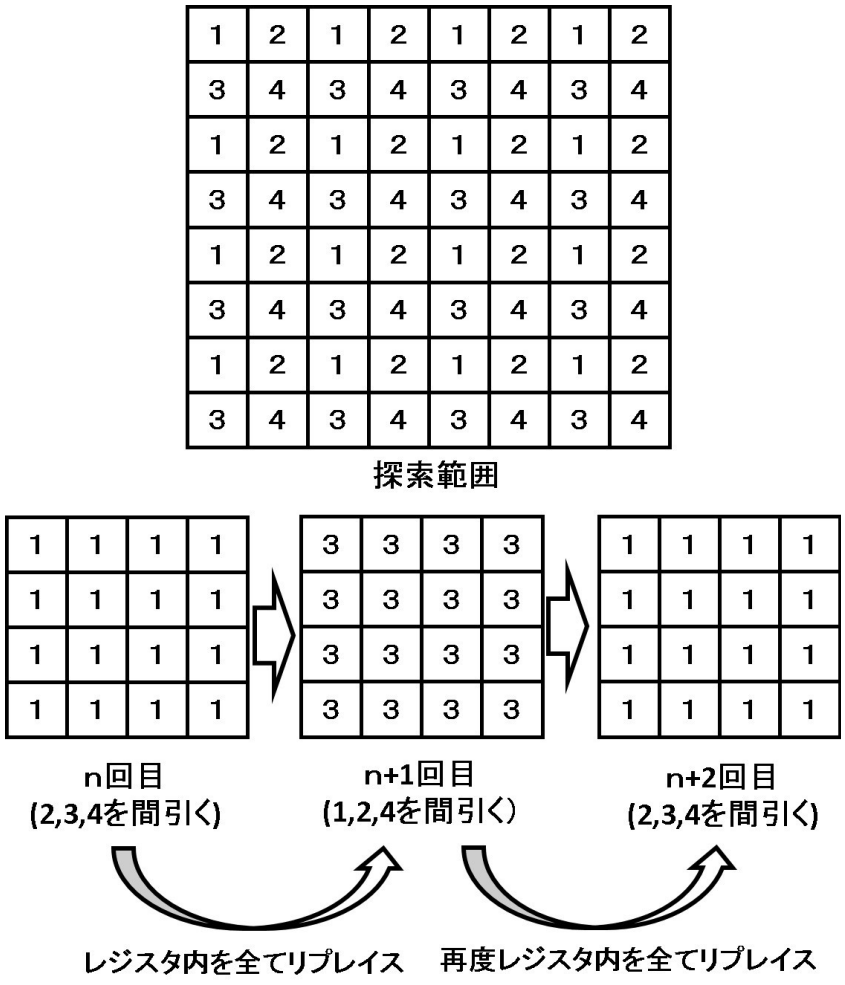

図 3.5: マッチングで間引く画素が異なる場合

4　提案手法

３章では従来法が引き起こす２つの問題点を述べた．１つ目は，従来の探索パターンが引き起こす演算回数とベクトルレジスタへのロード回数の増加，２つ目はサブサンプリングで間引く画素が探索点毎に異なるために引き起こされるロード回数の増加である．本章では，以上の問題を解決するために，３章で述べた問題を解決する提案法を述べる．

4.1　水平方向に拡張したクロスパターンによる追跡

拡大型ダイヤモンドサーチのような広範囲の探索パターンは，高い検出精度を得るのには効くものの，ベクトルレジスタ内に収めるには範囲が広過ぎる．一方，スクエアサーチのような小範囲の探索パターンは，ベクトルレジスタに収まるものの，精度よく検出できるのは探索開始点近傍に限られてしまう．そこで，両者の折衷的な探索パターンである SUC サーチ (Small Unsymmetric Cross サーチ) を提案する．SUC サーチを図 4.6 に示す．水平方向の探索点を垂直方向より多くしているのは，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に基づいている．

4.2　複数の縮小画像を用いたブロックマッチング

ブロックマッチングでサブサンプリングを用いる際の過剰なロードを抑えるために，参照画像の縮小画像を生成する．ブロックマッチングは，1 番目の縮小画像を使う探索点のマッチングを続けて実行してから，2 番目の縮小画像を使う探索点で演算をする．これを全ての縮小画像を用いた探索が終わるまで実行することで，3.3 で示した問題点のように，探索

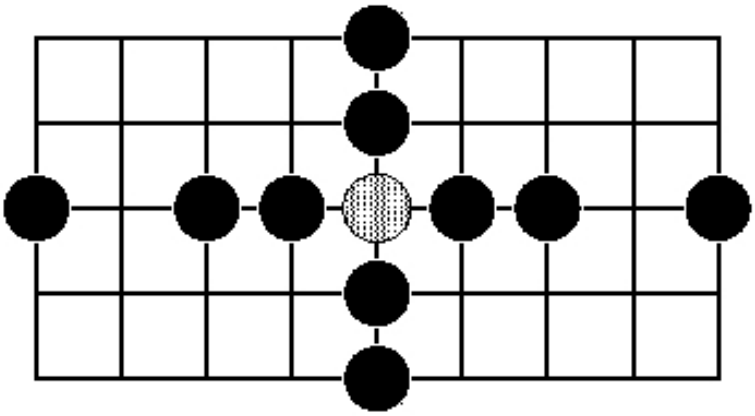

図 4.6: SUC サーチの探索点

点毎にレジスタファイルが読み込みを行うことがなくなるため，ロード回数が増加する問題を回避できる．図 3.5 で示した問題を，この手法を用いて表したものが図 4.7 である．図 4.7 では，1 の縮小画像を用いるマッチングを連続して実行することで，3.3 で発生した 18 回のベクトルレジスタのリプレイスを 9 回に抑えている．

表 4.2 は，それぞれ図 4.8，図 4.9，図 4.10 で示すように，参照画像を水平に 1 行ずつ間引いて縮小画像を生成する手法 (Hor と略記) と垂直に一列ずつ間引いた手法 (Ver と略記) と格子状に間引いた手法（Grid と略記）で実装して，4 種類のシーケンスを，SUC サーチを実装した HM で実行した結果である．評価項目は AD 演算回数の低減率 (AD)，ベクトルレジスタへのロード回数の低減率 (load)，同一画質における発生符号量の差分を評価した BD-Rate 悪化率 (BD) の 3 項目である． Hor と Ver は用いる参照フレームが確定した時点で，フレームの任意の画素を間引いた縮小画像を偶数列と奇数列（もしくは偶数行と奇数行）の 2 種類を生成する．Grid は正方形の 4 画素のうち 1 画素以外を間引くため，縮小

表 4.2: 各縮小画像を用いた結果

	load(%)	AD(%)	BD-Rate(%)
Hor	89.7	84.1	1.8
Ver	60.3	78.5	0.7
Grid	78.2	80.6	0.9

画像は 4 枚必要である．表 2 が示すように Hor は load や AD は改善するが，BD-Rate が大きく悪化することを示している．Hor は水平方向に画素を間引いて縮小画像を生成するため，ベクトルレジスタに格納される範囲が 24x24 となり，水平方向への範囲の拡張ができない．さらに 5 点以上離れた探索点を探索しないため，ラスター探索でリファインする条件も満たせないことが原因であると考えられる．Ver では水平方向に格納範囲を拡大できるため，探索精度に優れている．しかし，Grid と比較すると，BD-Rate と AD が同様の値であるのに対して，load は 78%に対して 60%程度しか低減できていない．この原因として Ver と Grid はラスター探索が実行される頻度は双方ともに変わらないが，格納範囲が Grid の方が高いことが挙げられる．SAD で使用する画素がより多く間引かれているため BD-Rate の精度は悪化しているが，僅か 0.17%の差であるので，この結果は無視できると考えて，提案手法では 4 画素を 1 画素として縮小画像を生成する Grid を採用する．

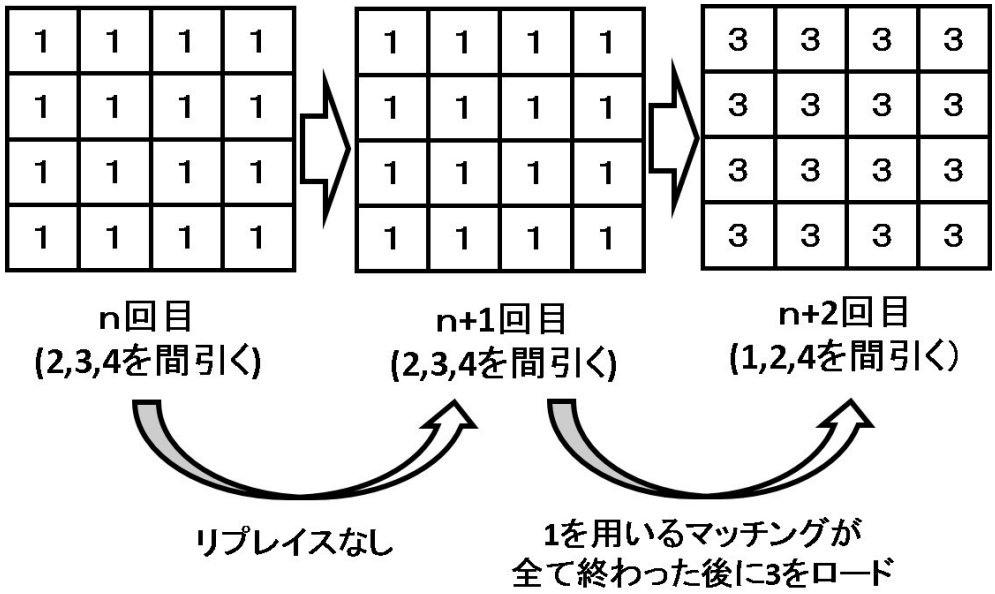

図 4.7: 縮小画像を使って最適化

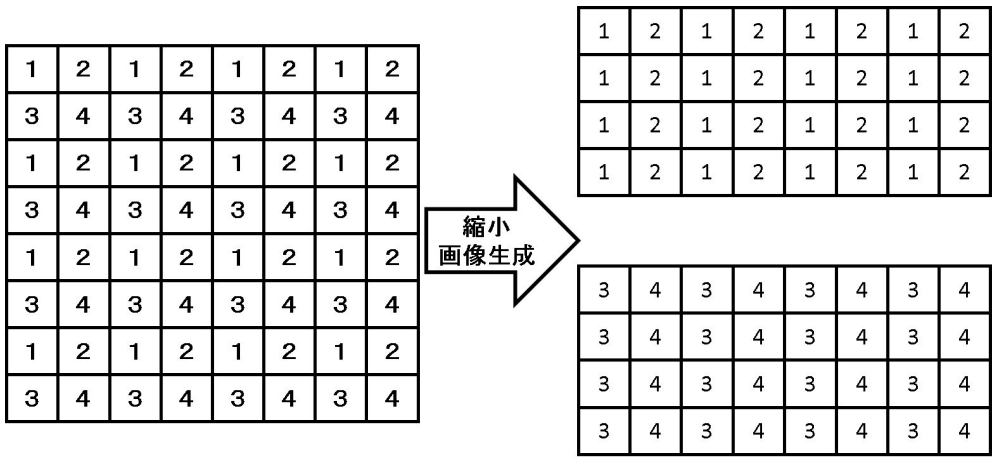

図 4.8: 水平に一行ずつ間引いた縮小画像の生成

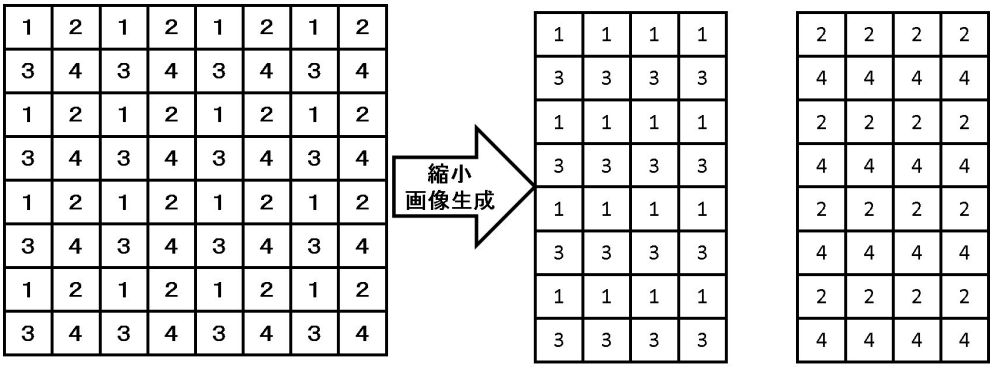

図 4.9: 垂直に一列ずつ間引いた縮小画像の生成

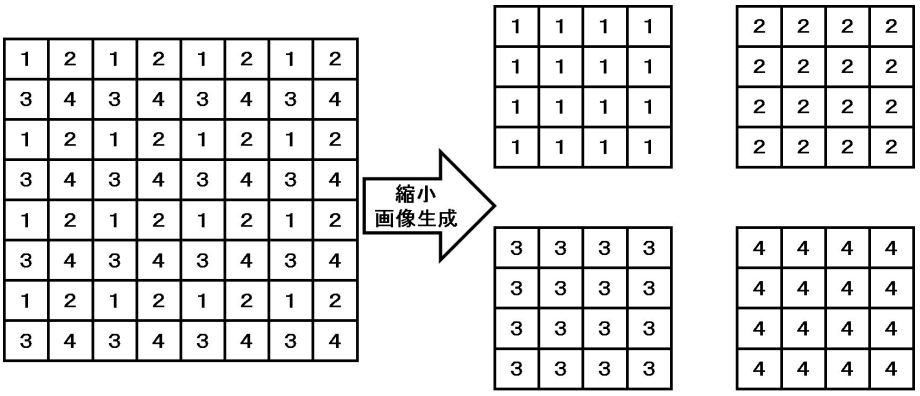

図 4.10: 格子状に間引いた縮小画像の生成

5　実装

5.1　実装アルゴリズム

本節ではHMに実装する提案手法のアルゴリズムの解説をする．提案アルゴリズムでは，動き探索に用いる参照画像が決定したとき，図4.10が示すように間引いた4種類の縮小画像が生成される．探索開始点を周囲の符号化済みブロックから予測して，SUCサーチを実行する．SUCサーチの探索点でサブサンプリングを用いるときに，1の箇所の画素を用いる点からブロックマッチングを行うために，図4.10の左上の縮小画像を読み込む．1の箇所のみを用いる探索点のマッチングを終えると続けて，2〜4のみで構成された縮小画像を用いて同様の処理をする．このアルゴリズムではベクトルレジスタに格納する範囲は，縮小画像を用いると24x48に拡張されるため，SUCサーチの探索点を水平方向に± 8の座標に2点追加している．この点によりラスター探索が発生しないことによるBD-Rateの悪化と過剰なラスター探索の発生を抑えられる．ラスター探索を実行した後は，結果のリファインのため，ラスター探索で検出された点から，再度SUCサーチを実行して，探索を終了する．もし初回の探索で探索中心の隣接した近傍点がSAD最小点となった場合，TZSearchと同様に，未探索の隣接点を探索する2点探索を実行して探索を終える．

5.2　ベクトルレジスタシミュレータ

従来手法と提案手法を用いたときの，ベクトルレジスタへのロード回数を定量的に評価するため，ベクトルレジスタシミュレータをHM10.0に実装した．このシミュレータはSADに用いる参照ブロックの位置とサイ

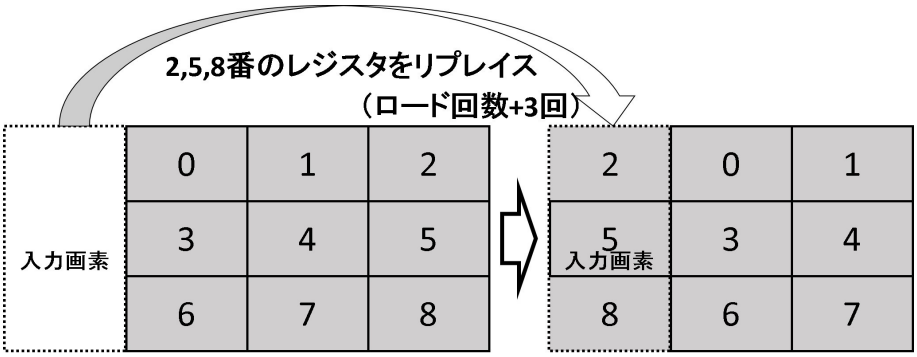

図 5.11: 垂直方向のリプレイス発生例

ズから，ベクトルレジスタに読み込む画素を算出する機能を備えた関数である．シミュレータの動作は各探索点で，縮小画像を用いた 24x12(現画像における 48x24) の範囲内に，SAD 演算で用いる参照ブロックが収まっている場合と収まっていない場合で処理が異なる．収まっている場合は，レジスタに格納された画素のリプレイスは発生しないため，ロード回数はカウントされない．一方で，参照ブロックがベクトルレジスタに格納された 24x12 の範囲内に収まっていない場合は，図 5.11， 5.12 に示すように探索範囲内に収まるように動的に格納する画素を読み込み，リプレイスが発生したレジスタの本数を数える．エンコード中にリプレイスが発生した合計回数を比較・評価に用いる．

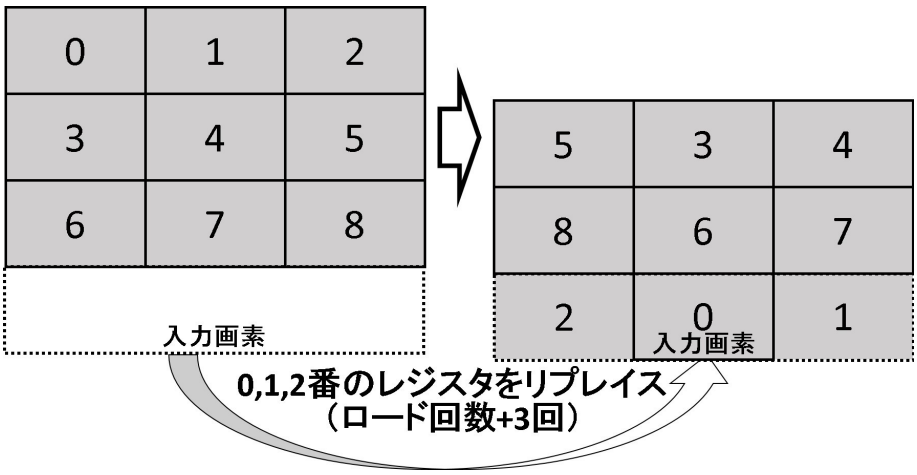

図 5.12: 水平方向のリプレイス発生例

6　評価

6.1　評価方法

比較は拡大型ダイヤモンドサーチを用いる TZSearch を基準にして，TZSearch の拡大型ダイヤモンドサーチを SUCSearch に変更した手法 (SUC)，SUC に正方形の 4 画素中の左上の画素のみを用いるサブサンプリングを適用した手法 (SUB)，４画素中の左上の画素で構成した縮小画像１種類のみを用いた手法 (Solo)，そして 4 章に述べた提案手法を HM10.0 に実装した手法 (Proposed) を用いる．TZSearch を比較に用いるのは，H.265 の標準ソフトウェアエンコーダに組み込まれているように検出精度の高さに定評があるからである. 評価は HEVC テストシーケンス画像 BasketBallDrill(BBD)，BQMall(BQM)，ParkScene(HDP)，Kimono(HDK) の４種類で比較する．それぞれ全フレームで量子化パラメータ 22，27，32，37 の値でエンコードして，AD 演算回数の低減率 (AD)，5.3 で述べたシミュレータを用いて結果を算出したベクトルレジスタへのロード回数の低減率 (load)，同一画質における発生符号量の差分を評価した BD-Rate 悪化率 (BD) の 3 つの項目で比較を行う.

6.2　評価結果と考察

表 6.3 は 4 種類のシーケンスの評価結果を，表 6.4 は各手法の評価結果の平均値を表している．提案手法では BD-Rate の悪化率は最も大きいもので 1.0%，平均で 0.8%と良好である．表 6.3 で BBD と BQM での悪化率が他の 2 つと比較して大きいのは，BBD の人とボールの動きの複雑さと，BQM の人ごみの細かい動きが，予測する際の画素を間引いたこと

で，うまく追跡されなくなったためであると考えられる．実際に，Solo における悪化率は 3%を超えており，間引いた上に縮小画像を 1 種類しか使わないことで，過剰に画素を間引くことになり，大きく BD-Rate を悪化させていることが分かる．表 6.4 から，ベクトルレジスタへのロード回数は Sub と比べて，縮小画像を用いる Solo で改善率が上がっているが，BD-Rate が大きく悪化してしまう．提案手法は，それを回避するため，4 種類の縮小画像を生成して，探索点内でブロックマッチングをする順番をロード回数が最小限になるように最適化することで，BD-Rate の悪化を抑えつつ，ロード回数の低減をしている．参照画像を複数読み込むので，Solo と比べてロード回数は平均で 13.0%増加しているが，BD-Rate の悪化は SUB と比べて，平均で 0.1%に抑えられており，提案手法は符号化効率を保ちつつ，ロード回数を SUB と比べて 13.3%低減していることが表 6.4 より分かる．AD 演算回数はサブサンプリングを適用した SUC から SUB で 20%以上の改善が見られるが，縮小画像適用後は大きな改善が見られない．これは，サブサンプリングを適用した手法と縮小画像を用いた手法では，探索点毎の演算回数には変化がないからである．

表 6.3: 各シーケンスの評価結果

	Sequence	SUC	SUB	Solo	Proposed
AD (%)	BBD	50.7	76.9	79.1	79.8
	BQM	56.8	79.7	79.7	81.6
	HDP	56.6	79.5	79.1	80.0
	HDK	53.0	78.3	79.1	79.5
load (%)	BBD	52.4	57.1	87.4	74.4
	BQM	58.8	62.9	87.2	74.4
	HDP	58.3	62.1	86.7	72.2
	HDK	54.3	59.9	36.0	74.3
BD (%)	BBD	0.6	0.9	2.3	0.9
	BQM	0.5	0.8	3.2	1.0
	HDP	0.3	0.5	1.8	0.6
	HDK	0.3	0.5	1.7	0.6

表 6.4: 評価項目の評価結果の平均

	SUC	SUB	Solo	Proposed
AD(%)	54.3	78.6	79.6	80.6
load(%)	55.9	60.5	86.8	73.8
BD(%)	0.5	0.7	2.3	0.8

7　あとがき

本論文は，開発中の高効率動き探索対応 SIMD データパスに向けて，従来の動き探索とそれに用いるサブサンプリング機能の問題を，複数の縮小画像を用いたブロックマッチングにより軽減することで，ベクトルレジスタへのロード回数と演算回数低減を両立する動き探索アルゴリズムを提案した．この提案手法は，開発中の SIMD データパスの高効率動き探索機能を活かすために，探索開始前に縮小画像を 4 枚生成することで，ベクトルレジスタに含まれる範囲を拡張して，そこに含まれる範囲を水平方向重視の SUC パターンとラスター探索を併用して探索している．また，探索点をマッチングする順番を，使用する縮小画像に応じて最適化することで，複数の縮小画像を用いることによるロード回数の増加を最小限に抑えている．この提案手法を HEVC 参照ソフトウェア HM10.0 に組み込んで，TZSearch と比較して，4 種類の動画像を評価した．その結果，AD 演算回数を 80.6%，ベクトルレジスタへのロード回数を 73.8%低減できることをそれぞれ示した．BD-Rate においては，BBD と HDK と HDP の 3 種類のシーケンスで 1.0%未満の悪化率となっており，物体の細かい動きに対応できていることを示している．しかし，4 シーケンスの平均の悪化率は，0.8% と十分に低く抑えられている．ただし，BQM では悪化率が 1.0%となっており，不規則な動きをする複数の物体の動きを追い切れず，正確な検出ができていない可能性がある．今後は，用いる縮小画像の枚数を動的に切り替えて，探索精度をシーケンスの特性に合わせることで，あらゆるシーケンスに対応できるように改善を図る．縮小画像の枚数を切り替える基準値は，ベクトルレジスタに格納されている

画素の分散値や周囲の符号化済みブロックの分散値を用いる．そうすることで，読み込み回数を抑えつつ各ブロックに対して適応的に探索精度を変更できるため，一部のシーケンスにおける BD-Rate の悪化を抑えられると考えられる．

謝辞

本論文の執筆にあたり，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，深澤祐樹研究員に感謝いたします．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] G.J.Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, Overview of the high efficiency video coding(HEVC) standard,IEEE Trans.Circuits Syst.Video Technol.22(12)(2012)1648-1667.

[3] Y. Fukazawa,K. Watanabe,Y. Minoura,T. Kondo and T. Sasaki,“ SIMD-based Datapath with Efficient Operation Structure, ”IEEE Proc. of ICASSP., pp.10311035, March 2016.

[4] Z.Chen, J.Xu, Y.He, J.Zheng, “ Fast integer-pel and fractional-pel motion estimation for H.264/AVC. ”Journal of Visual Communication and Image Representation 17.2(2006): 264-290.

[5] A.M.Tourapis“ Enhanced predictive zonal search for single and multiple frame motion estimation. ”Journal of Visual Communication and Image Representation 17.2(2006): 264-290.

[6] X. Tang, D.Sheng-kui and C.Can-hui, “ An analysis of TZSearch algorithm in JMVC, ” IEEE Proc. ICGCS.,pp.516-520,June 2010.

[7] Intel, Corp. (2010年4月). Intel Advanced Vector Extentions プログラミング・リファレンス [Online]: http://www.intel.co.jp/