

Master's Thesis / 修士論文

超高精細動画画像動き検出用参照画像メモリ 構成法の研究

友永, 大貴

三重大学, 2011.

三重大学大学院工学研究科博士前期課程情報工学専攻

<http://hdl.handle.net/10076/12901>

修士論文

題目

超高精細動画画像動き検出用
参照画像メモリ構成法の研究

指導教員

近藤 利夫 教授

平成 23 年度

三重大学大学院 工学研究科 情報工学専攻
計算機アーキテクチャ研究室

友永 大貴 (410M516)

内容梗概

2020年に試験放送が開始されるスーパーハイビジョンの符号化には、その高精細さから、特に動き検出に膨大な演算量をこなす必要があり、実時間処理には高並列処理が不可欠になっている。しかし、参照画像の転送ネックがその高並列実現の障害となっており、並列処理向けの参照画像メモリの開発が求められている。

本論文では、疎密階層探索において2次と3次探索にスクエアパターンサーチを用いる動き検出に対応する分散型参照画像メモリ構成を提案した。提案した分散型参照画像メモリ構成では、ライン単位でブロックマッチングアレイに供給可能な構成となっており、メモリ面積増加を抑えつつ、並列処理が可能なメモリ構成法を示した。また、参照画像メモリとブロックマッチングアレイの間にバッファを組み込み、初回以降のブロックマッチングをバッファ内のデータで行うことで、メモリユニットの稼働率を下げる低電力化構成も示した。

Abstract

Broadcast trial of High Definition Television (UHDTV) is scheduled for 2020. Coding of UHDTV needs enormous operations to meet a high accuracy requirement for motion estimation. Due to enormous operations, parallel processing is indispensable for real time encoding. But, the transmission bottleneck of a reference frame becomes an obstacle for highly parallel processing.

In this paper, I proposed a new distributed reference memory architecture that dissolved the transmission bottleneck at hierarchical motion estimation, which used square pattern search in 2nd search and 3rd search. At proposed memory architecture, it is possible to transmit pixel data by line units. Through this architecture, it suppresses an increase memory area, and can parallel processing. Also, an improvement adding a buffer between a block matching array and its local memory units is effective to reduce memory unit access rate or memory unit power consumption.

目 次

1	まえがき	1
2	スーパーハイビジョン符号化と参照画像メモリに対する要求条件	3
2.1	スーパーハイビジョンの概要	3
2.2	参照画像メモリに要求される転送レート	4
3	動き探索の概要と参照画像メモリ	5
3.1	動き探索と演算量低減手法の概要	5
3.2	追跡型動き探索	6
3.3	階層探索	7
3.4	拡張テンプレート複数併用法と疎密階層探索	7
3.5	ブロックマッチングの並列処理	8
3.6	参照画像メモリの役割	9
4	従来型並列構成とその問題点	13
4.1	従来型並列構成	13
4.2	SMP 型並列構成	13
5	分散型参照画像メモリの提案	15
5.1	分散型参照画像メモリ構成	15
5.1.1	テンプレートメモリ切り替え手法	17
5.2	メモリユニット構成	18
5.2.1	32 画素幅のメモリマクロ 1 個を用いた構成	19
5.2.2	1 画素幅のメモリマクロ 36 個を用いた構成	19
5.2.3	32 画素幅を複数のメモリマクロで分割した構成	20
6	バッファ付加による低電力化	23
7	評価	25
7.1	メモリユニット評価	25
7.2	バッファのヒット率	28
8	あとがき	30
	謝辞	31

参考文献	31
A 実験手順	32
A.1 ヒット率のシミュレーションの実装手順	32

目 次

2.1	スーパーハイビジョン	3
3.2	スクエアパターン	6
3.3	階層探索	7
3.4	拡張テンプレートで用いられる形状	8
3.5	マクロブロックの処理の流れ:単一処理	9
3.6	予測ベクトルの依存関係	9
3.7	マクロブロックの処理の流れ:並列処理	10
3.8	単一 BM アレイによる参照画像メモリ	10
3.9	32 画素幅メモリによるメモリマッピング	11
4.10	従来型並列構成	13
4.11	SMP 型並列構成	14
5.12	分散型参照画像メモリ構成	15
5.13	3x3 スクエアサーチに必要な画素	16
5.14	90 度回転した探索範囲に対するメモリユニット割り当て	16
5.15	アクセス競合	17
5.16	32 画素幅のメモリマクロ 1 個を用いた場合のメモリマッピング	19
5.17	1 画素幅のメモリマクロ 36 個を用いた場合のメモリマッピング	20
5.18	4 画素幅のメモリマクロ 8 個を用いた場合のメモリマッピング	22
5.19	メモリマッピング	22
6.20	バッファを付加した分散型参照画像メモリ構成	23
6.21	余分に転送できる画素	24
7.22	メモリ面積	25
7.23	動き検出に必要なアクセスサイクル数	26
7.24	面積当たりの実効バンド幅	27

表 目 次

5.1	切り分け単位とメモリマクロ個数とその仕様	21
6.2	余分に転送可能な画素幅	24
7.3	メモリユニット構成別の合計容量	26
7.4	バッファのヒット率 (Airplane)	28
7.5	バッファのヒット率 (Bronze)	28
7.6	バッファのヒット率 (Horse_race)	28
7.7	バッファの平均ヒット率	29

1 まえがき

近年、ビデオカメラやハイビジョン放送などとともに H.264/AVC が普及しているが、この符号化形式は MPEG-2 など従来方式のものと比較して複雑であり、ハードウェアコストの増加による製造コストや消費電力が問題となっている。現在ではスーパーハイビジョン放送等の研究が行われており、2020 年にはスーパーハイビジョンの試験放送が予定されている。スーパーハイビジョンはハイビジョンに比べて、画素数は 16 倍、動き検出演算量では 128 倍という高精細さであり、主に動き検出が原因で膨大な演算を必要とする。

動き補償では、動画像内のフレームをマクロブロック単位に分割し、その前後のフレームの比較を行う。そして、最も類似する部分を検出し、そのブロックの変移量である動きベクトルを求める。このフレーム間の類似性を利用して、データの差分値を動きベクトルに変換することで大幅にデータ量を削減することができる。この動き検出に関する手法は広く研究されており、代表的な手法としてダイヤモンドサーチなど追跡型の探索方式があり、範囲内の探索箇所を減らすことで、探索回数を減らす手法がある。さらに、既に検出された動きベクトルを利用した早期打ち切りを利用して演算量を低減した UMHexagons, EPZS などが現在ソフトウェアエンコーダを中心に利用されている。当研究室では、1 次探索に拡張テンプレートを用いる疎密階層探索を H.264/AVC に適用し、演算量低減の面は UMHexagons, EPZS を凌ぎながら、全探索並の高い精度を得るのに成功している。しかし、実時間処理には併せて高並列処理可能なハードウェアの実現が必要である。そのために、高並列実現上の最大の障害となっている参照画像の転送ネックを解消する必要がある。従来の共有メモリでは、ポート増加によるメモリ面積の増加やメモリ/キャッシュ間での転送ネックなど問題点が多く、この従来の共有メモリ構成に代わる並列処理向けの参照画像メモリの開発が求められる。

本論文では、疎密階層探索の 2 次探索と 3 次探索においてスクエアパターンサーチを用いる動き検出に対応する分散型参照画像メモリ構成を提案した。分散型参照画像メモリを構成するメモリユニットは 4 画素幅のメモリマクロを 8 個用いて構成しており、ブロックマッチングアレイにライン単位で供給可能なメモリマッピングをしている。このメモリマッピングでは、 3×3 スクエアパターンを用いて $16/8$ 画素幅サブブロックの全方向の探索を行うのに必要となる 18×1 ライン、 10×2 ラインを、1 サイクルでアクセス可能である。これにより、従来のブロック単位でのアク

セスが可能なメモリユニットと比較して、動き探索で用いるブロックマッチングに必要なアクセスサイクル数が 20%増加するもののメモリ面積を 45%削減でき、面積当たりの実効バンド幅では約 1.4 倍の改善率を得た。また、参照画像メモリとブロックマッチングアレイの間にバッファを組み込み、初回以降のブロックマッチングをバッファ内のデータで行うことで、メモリユニットの稼働率を下げる参照画像メモリの低電力化も図っている。バッファのヒット率の評価を行った結果、複数ある候補ベクトルの内、上位 3 つの 20x32 の探索領域をバッファに格納するとヒット率が 60%を超え、低電力化が可能である事が分かった。

2 スーパーハイビジョン符号化と参照画像メモリ に対する要求条件

2.1 スーパーハイビジョンの概要

スーパーハイビジョンとは現在広く研究されている超高精細映像システムであり，その解像度は従来のハイビジョンの解像度の 1920×1080 に対して縦横 4 倍の 7680×4320 の解像度がある (図 2.1)．また，フレームレートの面ではハイビジョンの 30fps に対し，スーパーハイビジョンは 60fps とハイビジョンの 2 倍ある．この精細度の高さからデータ量が 24Gbps と非常に大きくなる．この膨大なデータ量を圧縮するための演算量も膨大となっている．ハイビジョンと比較して動き探索に必要な演算量を式 1 に示す．スーパーハイビジョンの探索範囲の増加分は画像精細度が縦横 4 倍，フレーム数が 2 倍になったことでフレーム間距離が $1/2$ 倍となり，スーパーハイビジョンの探索範囲はハイビジョンに比べて 4 倍となる．また，画素数の増加分は 16 倍，フレームレートの増加分は 2 倍となるため，

$$\text{動き探索に必要な演算量} = 4 * 16 * 2 = 128 \quad (1)$$

となり，動き探索では工夫無しの場合だとハイビジョンの 128 倍の演算量が必要になってくる．

現在までに，この膨大な演算量を大幅に削減する手法が数多く提案されているが，まだ実用なレベルでの高探索精度かつ低データ転送レートと演算量低減が両立できていない．

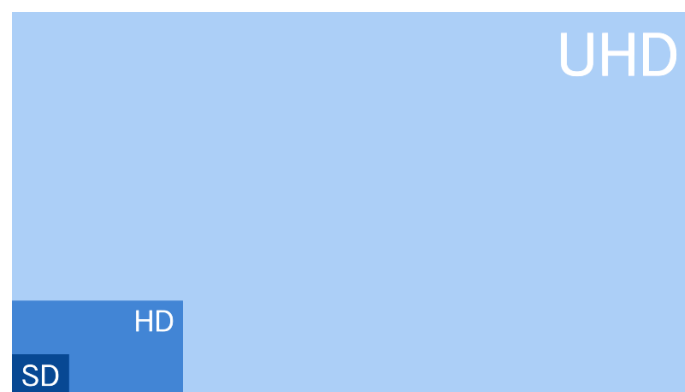


図 2.1: スーパーハイビジョン

2.2 参照画像メモリに要求される転送レート

ハイビジョンにおける 16×16 のマクロブロックによる動き探索に必要な転送量が1フレームにつき少なくとも2GBという結果を現在得ている．前で述べた通り，スーパーハイビジョンはハイビジョンに比べて画素数が縦横4倍，フレームレートが60fpsとなっている．そのため，少なくとも1920GB/sの転送レートの確保が必要となってくる．また，H.264/AVCでは可変ブロックサイズでの動き探索を行っているため，この可変ブロックサイズを考慮すると約7680GB/sの転送レートが必要になるという結果を得ている．このことから，参照画像メモリには1920～7680GB/sの転送レートの確保が必要となる．

3 動き探索の概要と参照画像メモリ

3.1 動き探索と演算量低減手法の概要

動き探索ではマクロブロック、あるいはサブブロック単位でブロックマッチングを行い、参照画像の中で符号化対象ブロックと最も似ている位置を探し出す。この位置を探し出すために1つずつの画素値の差の絶対値の和を使い、SAD(差分絶対値和)が最小となるブロックまでの変移量を動きベクトル(MV)とする。符号化対象画像をX、参照画像をYとしたとき、NxNブロックごとの画素値の差分は以下ようになる。また、(u,v)はブロックの変移量を表す。

$$SAD(X, Y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |X(i, j) - Y(i + u, j + v)| \quad (2)$$

これにより、画面内の冗長情報の削減ができるため、画素値をそのまま符号化するのと比べてデータ量を大幅に削減することができる。しかし、全ての点を探索(全探索)すると演算量が膨大になってしまうため、動き検出処理には様々な演算量低減手法が提案されている。特定の探索パターンを用いて、局小SADを追跡し最小点を見つけることにより探索点を省く方法がソフトウェアエンコーダでは一般的に主流となっている。

探索パターン内の局小点に中心を移動する探索を中心にくるまで探索を繰り返す追跡型動き探索がある。この動き探索手法では、探索パターン上の中心とその周囲のSADを調べ、SADが最も小さい箇所を次の中心点として同様の処理を行い、中心点が最も小さくなるまで処理を繰り返す。追跡型動き探索手法は、探索点を減らすことで、大幅に演算量を低減している。しかし、この手法は探索開始点に大きく影響されるため、常に高い精度が得られるわけではない。これは、周辺ブロックよりも小さいSADを持つブロックが存在することは珍しいことではなく、そのような局所的にSADが小さくなる箇所の誤検出を回避することができないためである。このため、最初にSAD最小の箇所を予測して、そこを探索開始点として利用することが必要になる。

最初にSAD最小の箇所を予測する手法として、疎密階層探索がある。探索範囲内の総点を探索対象としている疎密型の階層探索手法は、解像度を下げたサブサンプリング画像を用いて探索を行うことにより、詳細に探索を行う必要がある箇所を特定する方法である。このため、探索開始点に影響されることがないので、局所的なSAD最小点を誤検出してし

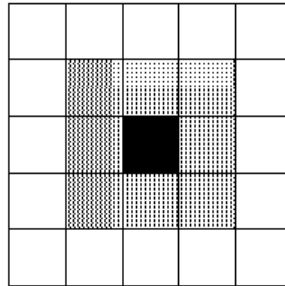


図 3.2: スクエアパターン

まう問題を回避することができる。ただし，粗探索での精度があまり高くないため，本来の解像度で探索し直しても十分な精度が得られない場合もある。

以下では，主な追跡型動き探索手法であるスクエアサーチ，疎密階層探索の改善手法として当研究室で提案している拡張テンプレート複数併用法を説明していく。また，並列処理におけるマクロブロックの流れとブロックマッチングでの参照画像メモリの役割を述べていく。

3.2 追跡型動き探索

追跡型動き探索には，ダイヤモンドサーチ，ヘキサゴンサーチやスクエアサーチがある。

スクエアサーチの動き探索手法は 3×3 点に位置する 9 個のブロックの中から，SAD が一番小さいものを探す (図 3.2)。続いて，その最小点を中心として新たな 3×3 点上のブロックを探索する。この最小点を追跡する探索を繰り返し，SAD の最小点が 3×3 の中心点に来たら探索を終える手法である。スクエアサーチでは，動き探索で探索位置を移動し，新たな探索範囲を探索する際，前回との共通部分のメモリアクセスが不要となり，データの再利用が高ければメモリアクセス数を少なくできる。また，探索点が互いに隣接しているので，読み込んだ参照画像データの再利用がしやすく，探索点が多いため並列処理の効果が高いことが期待できる。

追跡型動き探索は，局小 SAD を追跡して動き探索を行うことで，演算量を低減している。しかし，探索精度は探索開始点に大きく影響され，誤検出をする可能性があり，誤検出を防ぐためには，最初におおまかな SAD 最小の箇所を予測し，そこを探索開始点に動き探索をする必要がある。

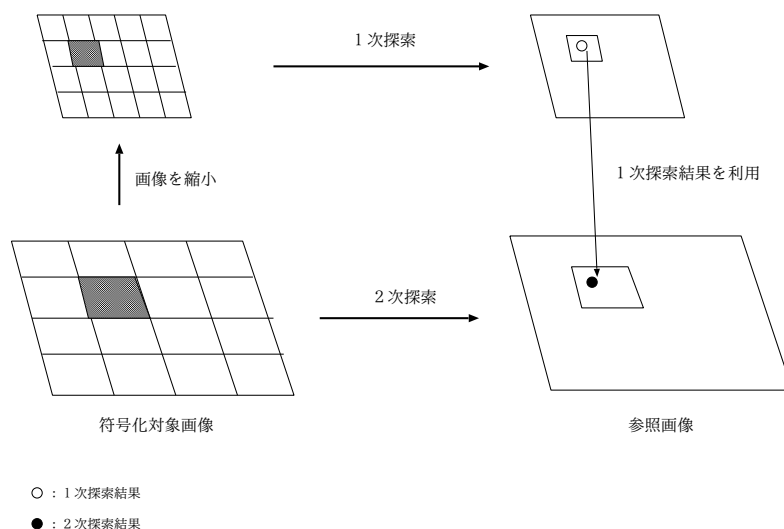


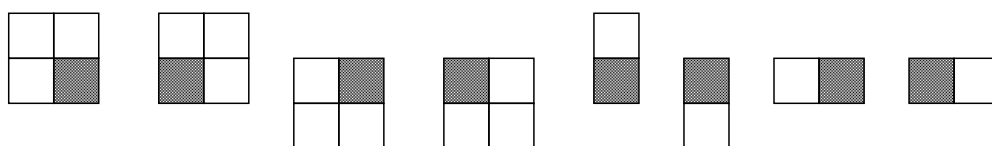
図 3.3: 階層探索

3.3 階層探索

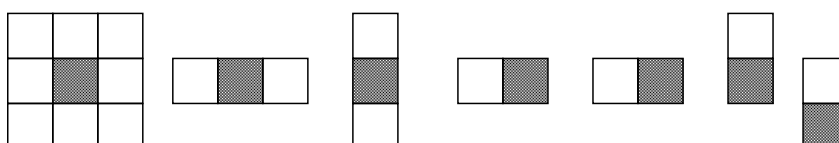
階層探索とは、まず画像を縮小し、縮小画像で最初の動き探索を行い、その探索結果から探索区域を限定して、細かな探索を行う動き探索手法である(図 3.3)。追跡型動き探索手法では、探索開始点によって探索精度が左右されたが、この階層探索手法は、1次探索において探索区域内の全点を探索対象としているため、予測はずれとなってしまうような事が少なく、無駄な探索をすることが少ないという性質がある。しかしながら、単画素精度での詳細な探索に比べて、4画素精度での探索は画素数が16分の1になるため、情報量が欠落することで探索精度が低下してしまう問題点がある。

3.4 拡張テンプレート複数併用法と疎密階層探索

3.3節で述べたとおり、階層探索では画像を縮小して探索を行うため探索精度が低下してしまう問題点がある。そこで、当研究室ではこの欠点を補うために拡張テンプレート複数併用法を提案している。従来ブロック単位で行っていたブロックマッチングを隣接ブロックを組み合わせた拡張テンプレート単位で行う。拡張テンプレート単位で行うことにより、探索対象画素が増えるため、探索能力が向上し誤検出の可能性を減らす



2x2の拡張テンプレート形状



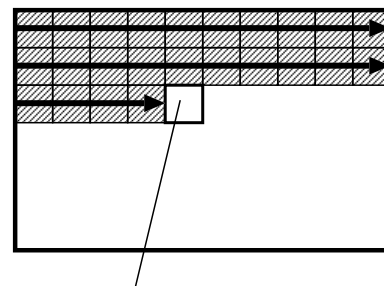
3x3の拡張テンプレート形状

図 3.4: 拡張テンプレートで用いられる形状

ことができる．この拡張テンプレート複数併用法では，複数の異なる組み合わせ方(図 3.4)で拡張し，それらを用いて 1 次探索の精度を向上させている．1 次探索に拡張テンプレートを用いる疎密階層探索を H.264/AVC に適用したところ，大幅に演算量を低減させつつ全探索並の高い精度を得ている．演算量低減の面は，既存の代表的な手法である UMHexagons，EPZS を凌ぎつつ，全探索並の高い精度を得るのに成功している．しかし，スーパーハイビジョン符号化の実時間処理には並列処理が必要であり，それに伴い並列処理可能なハードウェアも必要となる．

3.5 ブロックマッチングの並列処理

動き探索を逐次処理で行うと演算時間が伸びてしまうため，スーパーハイビジョンの実時間処理には並列処理が必須である．通常，ブロックの符号化をしていく順は，左上から右下の順のラスタ走査順で符号化を行っていく(図 3.5)．これは，あるブロックを符号化する際に，そのブロックの上，右上，左のブロックの符号化結果を元に符号化するからである(図 3.6)．そのため，マクロブロックの並列処理を考慮すると，この予測ベクトルの依存関係を無視して並列処理を行うことはできない．そこで，この予測ベクトルの依存関係を考慮した並列処理を考える．まず図 3.7 の



符号化対象ブロック

図 3.5: マクロブロックの処理の流れ:単一処理

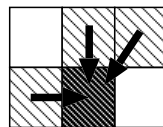


図 3.6: 予測ベクトルの依存関係

(a) の 1 ライン目のブロック 2 つは通常通り符号化する．次に，ブロック B は 1 ライン目の上，右上の 2 つが符号化済みなので，符号化可能である．よって，2 ライン目のブロック B と 1 ライン目のブロック A は同時に符号化できる．次に，図 3.7 の (b) の 2 ライン目のブロック D は上，右上，左のブロックが符号化済みなので符号化可能である．よって，1 ライン目のブロック C と 2 ライン目のブロック D は並列に処理できる．このように並列に処理できるブロックを考えていくと，現在のブロックから 2 つ左，1 つ下に位置するブロックを並列に処理することができることが分かる．そして，図 3.7 の (c) のブロック A, B, C, D のようにマクロブロックの並列処理をしていくことを考えていく．ただし，図 3.7 では 4 並列と仮定している．

3.6 参照画像メモリの役割

参照画像メモリとは BM アレイ (ブロックマッチングアレイ) に参照ブロックを供給するメモリである．参照ブロックはブロックマッチングを行う際に必須であり，実時間処理を考慮するとこの参照ブロックを効率よく供給することが重要になる．単一の BM アレイによる参照画像メモリ

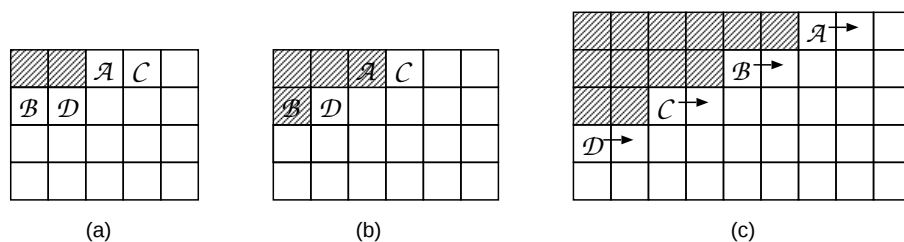


図 3.7: マクロブロックの処理の流れ:並列処理

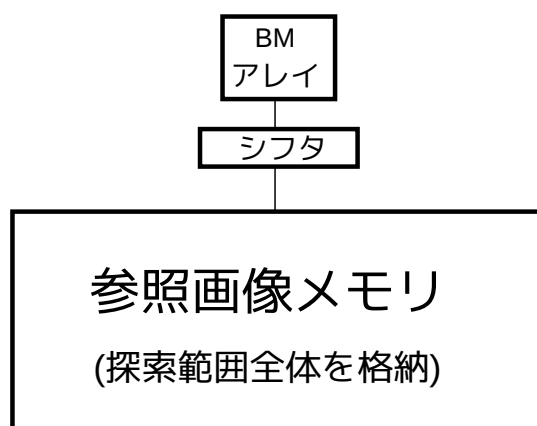


図 3.8: 単一 BM アレイによる参照画像メモリ

を示す(図 3.8)．BM アレイとはテンプレートブロック(符号化対象ブロック)と参照ブロックの SAD を算出し、動き検出を行うものである．参照画像メモリから参照ブロック、テンプレートメモリからテンプレートブロックをそれぞれ BM アレイに供給し SAD を算出し動き検出を行う．

参照画像メモリは探索範囲を格納し、BM アレイに参照ブロックを転送する．参照画像メモリに対する探索範囲の格納方法は図 3.9 のように、探索範囲に対してメモリのアドレス番号を割り当て、参照画像メモリに格納する．また、図 3.9 に 32 画素幅のメモリに対するメモリマッピングである．このように探索範囲を格納して、参照画像メモリから参照ブロックをライン単位で BM アレイへ供給する．このメモリマッピングではメモリアドレス 1 番地と 2 番地にまたがって存在するラインをアクセスする際、メモリアドレス 1 番地と 2 番地の 2 つをアクセスしなければならない．これは、メモリの整列化制約のためである．参照ブロックを供給

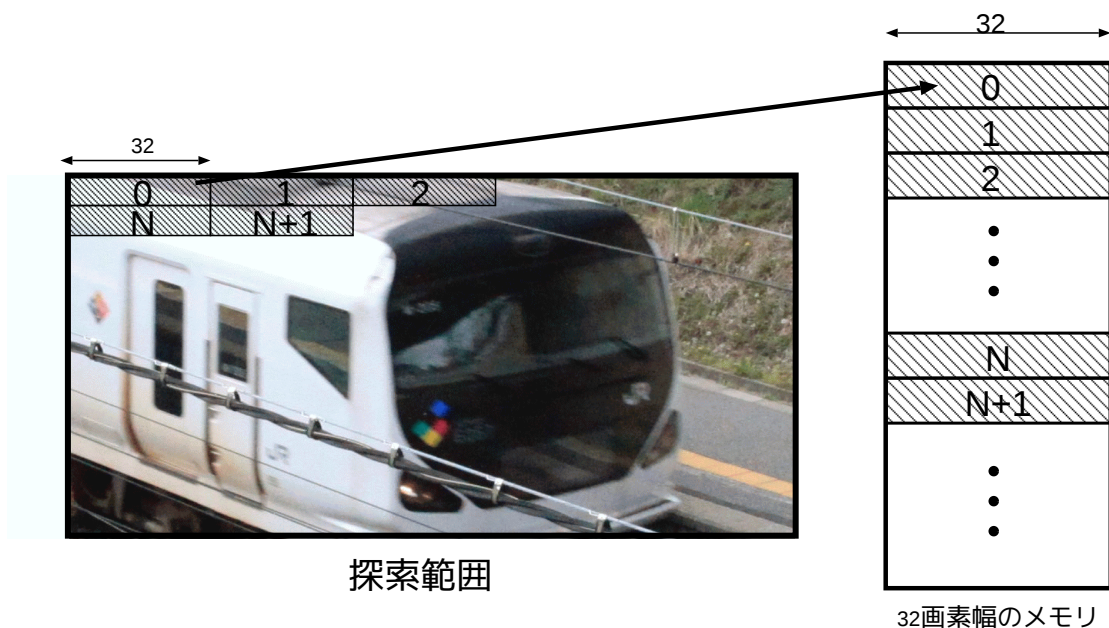


図 3.9: 32 画素幅メモリによるメモリマッピング

するには任意の位置のアクセスが必要であり，参照ブロックを効率よく BM アレイに供給するには，この整列化制約の問題点を解決する必要がある．

3.3 節で述べたとおり，1 次探索で拡張テンプレートを用いた疎密階層探索で演算量低減と高い精度を得ている．疎密階層探索の 2，3 次探索ではスクエアサーチを用いて細かな探索を行っており，1 次探索と比較して 2，3 次探索に要する演算量が増大している．よって，2，3 次探索には非常に高い演算速度が要求されている．動き検出を行うには，参照画像の供給が不可欠であり，現状の参照画像メモリの転送ネックにより，要求される演算速度が得られていない．また，参照ブロックの供給には高速なランダムアクセスが必要であり，高速なランダムアクセスが可能なメモリでは，メモリ面積が増加してしまい，コスト増加の原因になってしまう．そのため，メモリ面積増加を抑えつつ，ランダムアクセスが可能なメモリ構成が求められている．ランダムアクセスを可能にするには，先に述べた整列化制約の問題点を解決する必要がある．ここでの整列化制約の問題点は，動き検出を行うのに必要な任意のラインを 1 サイクルでアクセスをすることができないことである．提案手法でのメモリユニッ

ト構成では、この任意のラインを1サイクルでアクセスが可能なメモリマッピングをしている。

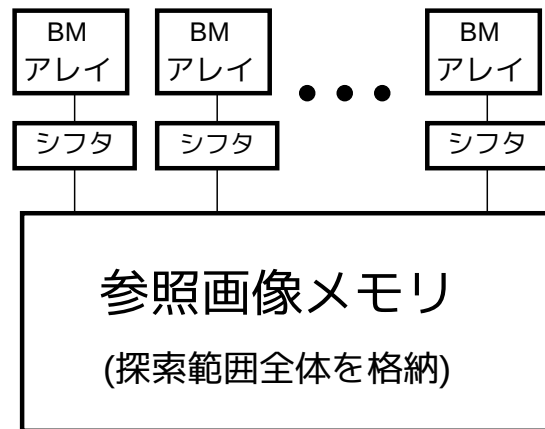


図 4.10: 従来型並列構成

4 従来型並列構成とその問題点

スーパーハイビジョンはその高精細さ・演算量の高さから実時間処理には並列処理が不可欠と考えられる．並列処理を可能にするには，その並列処理に適応した参照画像メモリが必要になる．しかしながら，一般的な並列構成に様々な問題点が生じる．一般的な並列構成では，主に単一のメモリに複数の BM アレイを接続する構成を採っている．

4.1 従来型並列構成

単一メモリと複数の BM アレイを繋ぐ従来型並列構成について説明する (図 4.10)．このような従来型構成の問題点は，複数ポートで単一メモリと複数の BM アレイを接続しているため，ポート数に比例してメモリ面積が増加してしまうことである．メモリ面積が増加してしまう事は，コスト増加になってしまう．

4.2 SMP 型並列構成

対称型マルチプロセッシング (SMP) 方式による並列構成においても問題点が生じる．ここでの SMP 方式は，単一の参照メモリを複数の BM アレイが共有し，メモリ/キャッシュ間をバスを通して接続する方式を採っている (図 4.11)．このキャッシュは各ブロックがブロックマッチングを行

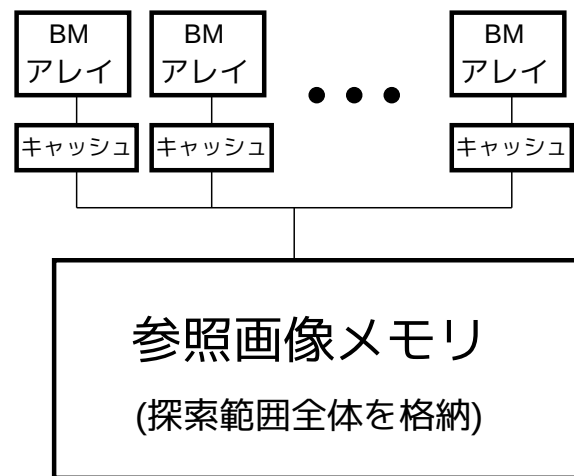


図 4.11: SMP 型並列構成

うのに必要な画素のデータを格納している．このデータ転送の際に動きの激しい動画像の場合，メモリ/キャッシュ間で転送ネックが起こってしまい，処理に支障を来す問題が生じてしまう．

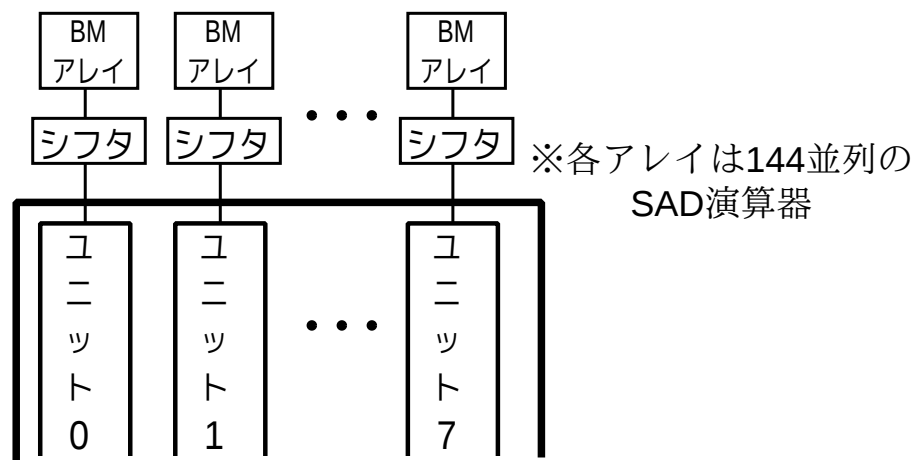


図 5.12: 分散型参照画像メモリ構成

5 分散型参照画像メモリの提案

5.1 分散型参照画像メモリ構成

一般的に、参照画像メモリは単一メモリによる共有メモリで構成されている。この共有メモリ構成では、4 節で述べたポート数増加によるメモリ面積増加、メモリ/キャッシュ間での転送ネックなどの問題点が存在する。そこで、本論文では分散型参照画像メモリを提案した。この分散型参照画像メモリでは、複数のユニットの各々を BM アレイに接続することで、バンド幅をかせぐ並列アクセス構成を採っている (図 5.12)。また、本論文で提案する分散型参照画像メモリは 8 並列を仮定している。3x3 のスクエアサーチの並列処理を行うのに必要な BM アレイを示す。図 5.13 のブロック A の 3x3 スクエアサーチを行うには画素 A と画素 B を含む周囲 8 つの画素が必要となる。同様に、画素 B の 3x3 スクエアサーチを行うには、画素 A を含む画素 B と周囲 8 つの画素が必要となり、画素 C の 3x3 スクエアサーチを行うには、画素 C と周囲 8 つの画素が必要となる。このように考慮すると、3x3 スクエアサーチを行うには 1 つの画素につき、9 並列の SAD 演算器が必要となる。マクロブロックの最大ブロック単位は 16x16 であるため、

$$3x3 \text{ スクエアサーチ} * 16x16 \text{ のサブブロックの 1 辺} = 9 * 16 = 144 \quad (3)$$

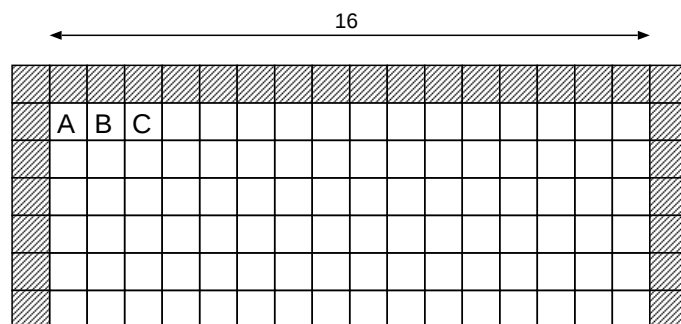


図 5.13: 3x3 スクエアサーチに必要な画素

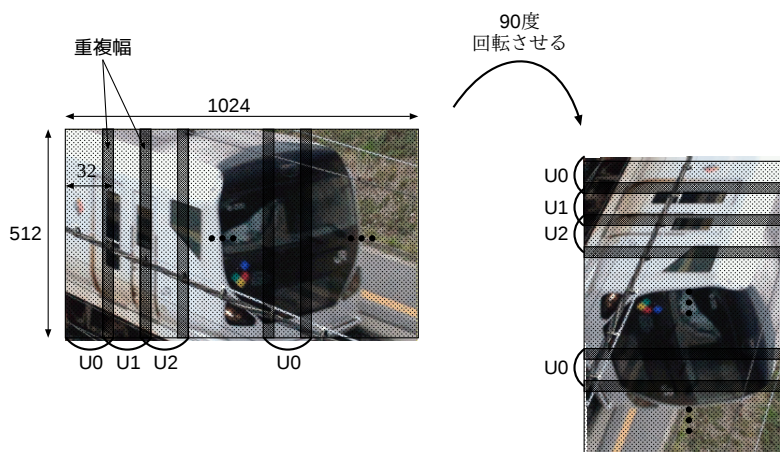


図 5.14: 90 度回転した探索範囲に対するメモリユニット割り当て

となり，3x3 スクエアサーチの並列処理を行うには，144 並列の SAD 演算器が必要になる．参照画像メモリへの探索範囲の格納方法を示す．32 画素幅ごとに 1 つの参照ブロックをアクセスすることを前提に，スーパーハイビジョンの動き検出に要求される 1,024x512 の探索範囲を 32 列に分割し，その n ， $n+8$ ， $n+16$ ， $n+24$ 列を 90 度回転し，ユニット n に格納する (図 5.14)．ただし，各列は $32 + \alpha$ 画素とし， α だけ重複させている．重複する幅はユニットを構成しているメモリマクロ構成に依存される．探索範囲を 90 度回転させて参照画像メモリに格納しているのは，重複する幅をライン単位で設定できるようにするためであり，ライン単位で重複幅の設定が可能なメモリマクロ構成法を採っている．メモリマクロ構成の詳細は 5.2 節で説明する．重複させるのは，アクセス競合が発生し，並

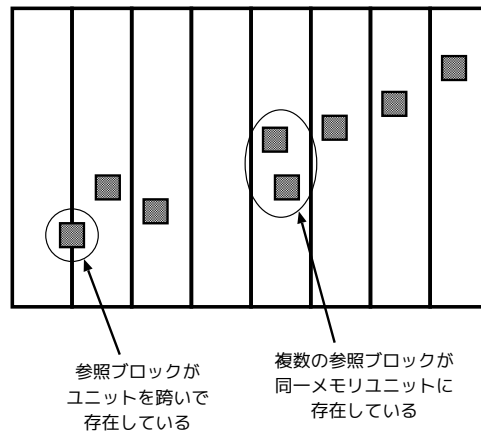


図 5.15: アクセス競合

列処理の効率が低下してしまうのを防ぐためである．アクセス競合が発生した場合，1 サイクルのアクセスで BM アレイに供給すべきデータを 2 サイクルのアクセスが必要になってしまい，効率良く BM アレイに参照ブロックが供給できなくなってしまう．図 5.15 にアクセス競合が発生している場合の参照ブロックの位置を示す．アクセス競合が発生するのは，複数の参照ブロックが同一メモリユニットに存在する場合，参照ブロックがメモリユニットをまたいで存在する場合が考えられる．このようなアクセス競合を防ぐために，隣接ユニットとの重複もしくは隣接ユニットとの接続が必要になる．しかし，このアクセス競合回避機構はメモリ面積増加やメモリユニットと BM アレイ間のネットワーク規模増大といった問題点も生ずることが考えられる．

5.1.1 テンプレートメモリ切り替え手法

テンプレートメモリにはブロックマッチングに必要なテンプレートブロックを格納している．通常のテンプレートメモリは，ユニット 0 に接続されるテンプレートメモリはテンプレートブロック 0，ユニット 1 に接続されるテンプレートメモリはテンプレートブロック 1 のように各メモリユニットに対応しているテンプレートブロックのみのブロックを格納している．

提案する参照画像メモリにおけるテンプレートメモリには，メモリユニット 0 に接続されるテンプレートメモリに全てのテンプレートブロッ

クを格納する．そして，ブロックマッチングを行う際に，テンプレートブロックを切り替える手法を採る．例えば，メモリユニット 0 から BM アレイへテンプレートブロック 1 に対応する参照ブロック 1 を供給する場合，通常のメモリではメモリユニット 0 に接続されているテンプレートメモリにはテンプレートメモリ 0 のデータが格納されておらず，テンプレートメモリ 1 が接続されている BM アレイへメモリユニット 0 から参照ブロック 1 を供給する必要がある．テンプレートブロックを切り替える手法を用いると，テンプレートメモリに全てのテンプレートブロックのデータが格納されているため，各々に接続されている BM アレイに参照ブロックを供給すれば，ブロックマッチングが可能である．この手法により，参照ブロックと対応するテンプレートブロックがテンプレートメモリ内に格納されていないという問題点が解消される．これは，提案した分散型参照画像メモリは図 5.14 で示しているように 32 画素幅ごとに切り分けて，折り返しユニットに格納しているため，各ユニットに必ずしもそのユニットに対応する参照ブロックが存在しているわけではないからである．

5.2 メモリユニット構成

提案した分散型参照画像メモリは複数のメモリユニットで構成されており，そのメモリユニットは複数のメモリマクロで構成されている．メモリユニットに格納されている参照画像 (探索範囲) には，それぞれメモリマクロが割り当てられ，そのメモリマッピングによって，BM アレイに供給が可能な範囲が決められる．また，メモリユニット構成はメモリ面積にも影響し，メモリマクロを多く使用した構成法などはメモリ面積が大きくなってしまう問題点が生じる．これは，メモリマクロには格納領域の他に入出力装置など周辺機器が必要であるため，その分のメモリ面積が増加してしまうからである．BM アレイに供給可能な範囲は，メモリマッピングに依存される．そのため，メモリ面積増加を抑えつつ，要求される動き探索アルゴリズムに適応したメモリマッピングを考えなければならない．

- 32 画素幅のメモリマクロ 1 個を用いた構成
- 1 画素幅のメモリマクロ 36 個を用いた構成
- 32 画素幅を複数のメモリマクロで分割した構成

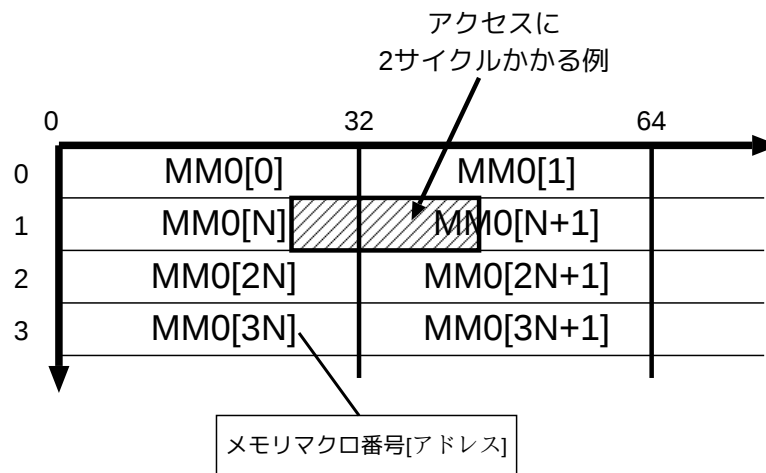


図 5.16: 32 画素幅のメモリマクロ 1 個を用いた場合のメモリマッピング

5.2.1 32 画素幅のメモリマクロ 1 個を用いた構成

32 画素幅のメモリマクロを 1 個用いたメモリユニット構成を示す．これは 3.6 節で述べた 32 画素幅のメモリと同様なもので，探索範囲に対するメモリマッピングも 3.6 節で述べたメモリマッピングと同様である．図 5.16 にメモリマッピング図を示す．この構成で用いてるメモリマクロは 1 個なので，図 5.16 で使用しているメモリマクロは MM0 のみである．また，このメモリマッピングは一般的なメモリマッピングであり，3.6 節で述べた整列化制約による問題が起きてしまう．整列化制約の問題が起こる場合は，動き検出に必要となる任意のラインが異なるアドレスに格納されている場合である．図 5.16 では，MM0 の N 番地と $N+1$ 番地に必要な任意のラインが格納されているため，この任意のラインを BM アレイに供給するには，2 サイクルかかってしまう．しかし，用いるメモリマクロは 1 個であるため，メモリ面積は小さくなると考えられる．

5.2.2 1 画素幅のメモリマクロ 36 個を用いた構成

1 画素のメモリマクロ 36 個用いたメモリユニット構成を示す．この構成は文献 [2] で用いられている構成である．文献 [2] では， 4×4 の任意のブロックを 1 サイクルで BM アレイへ供給可能にしているが，スクエアパターンサーチの並列処理を考慮するとサブブロックの最小単位である

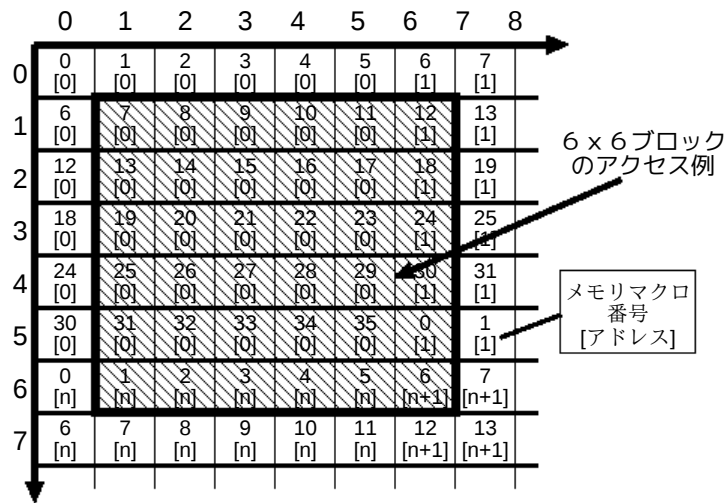


図 5.17: 1 画素幅のメモリマクロ 36 個を用いた場合のメモリマッピング

4x4 のサブブロックを並列で処理をするために 4x4 のサブブロックの周囲 1 画素を含んだ合計 6x6 を BM アレイへ供給することが求められる．そこで，今回は文献 [2] で用いられている 4x4 のアクセスを拡大した 6x6 の任意のブロックを 1 サイクルで BM アレイへ供給可能な構成としている．図 5.17 にメモリマッピング図を示す．メモリマクロを 36 個用いて図 5.17 のようにメモリマッピングを行い個別にアドレスを指定することで，任意の 6x6 ブロックが 1 サイクルで BM アレイに供給が可能である．

5.2.3 32 画素幅を複数のメモリマクロで分割した構成

5.2.1 節では 32 画素幅を 1 個のメモリマクロを用いた構成でのメモリマッピングを示した．ここでは，その 32 画素幅を複数のメモリマクロを用いてメモリマッピングをしている．本論文では，この 32 画素幅を複数のメモリマクロを用いて分割することで，整列化制約を緩和している．分割する幅は 1 画素幅，2 画素幅，4 画素幅，8 画素幅，16 画素幅単位で分割した．表 5.1 に切り分ける単位と 1 個のメモリユニットに用いられるメモリマクロ个数，そのメモリマクロの仕様を示す．また，表 5.1 で示している面積はメモリマクロ 1 個当たりの面積である．図 5.18 は 4 画素幅単位で分割したメモリマッピングの例で，4 画素幅のメモリマクロ 8 個用いたメモリユニット構成である．図 5.19 は，4 画素幅以外のメモリマッピン

グを示している．この分割したメモリユニット構成は1画素，2画素，4画素単位で分割した構成は整列化制約を緩和することができ，16/8画素幅サブブロックのスクエアサーチに必要な任意の18x1ライン，10x2ラインを1サイクルでBMアレイに供給することができる．1画素，2画素幅単位で分割する構成では，32画素幅の分割を行わず，18x1ライン，10x2ラインをそれぞれ1サイクルでBMアレイへ供給可能な構成としている．また，8画素，16画素単位で分割した構成は整列化制約を緩和することができていない．8画素分割では，10x2ラインを供給する際に2サイクル要する場合がある．同様に16画素分割では，18x1ライン，10x2ラインを供給する際に2サイクル要する場合がある．表5.1では，切り分け単位を小さくする毎にメモリマクロ個数が増加している．メモリマクロ個数が増加すれば，メモリ面積が増大することが考えられる．しかし，切り分け単位を大きくすると先に述べたように整列化制約のために，所要サイクル数が増加してしまう．そこで，メモリ面積増加を抑えつつ，整列化制約を緩和できる分割幅を決めなければならない．

表 5.1: 切り分け単位とメモリマクロ個数とその仕様

切り分け単位	メモリマクロ個数	(ビット幅, ワード長)	面積 (nm^2)
1 画素	20 個	(8bit, 3296)	0.01876
2 画素	12 個	(16bit, 2752)	0.02444
4 画素	8 個	(32bit, 2048)	0.03226
8 画素	4 個	(64bit, 2048)	0.05882
16 画素	2 個	(128bit, 2048)	0.10655

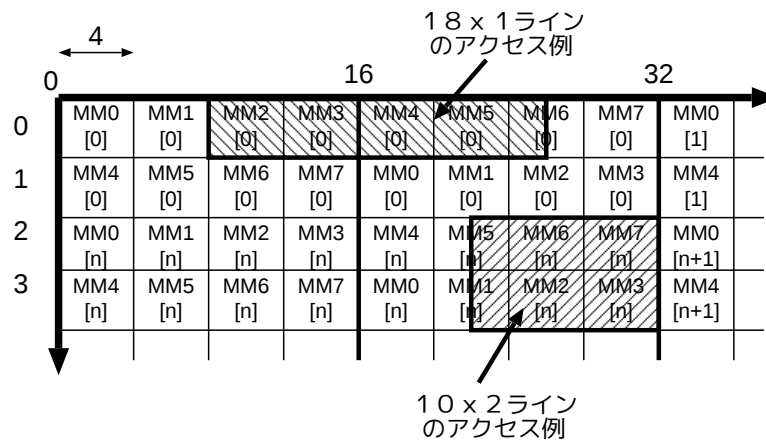


図 5.18: 4 画素幅のメモリマクロ 8 個を用いた場合のメモリマッピング

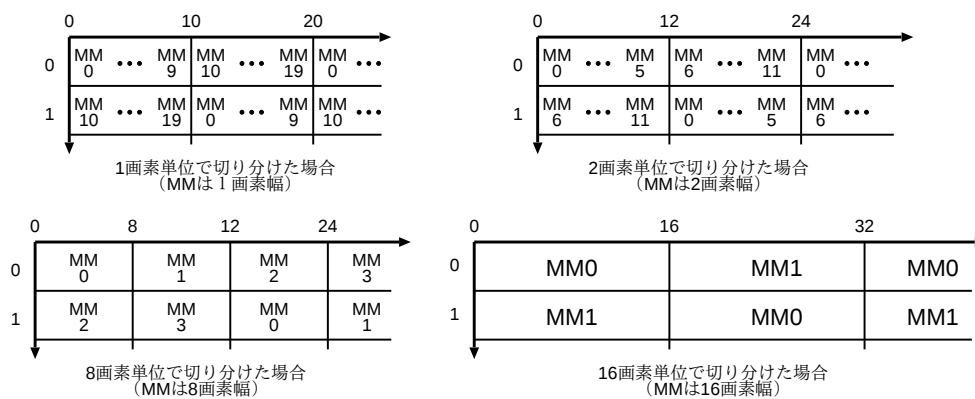


図 5.19: メモリマッピング

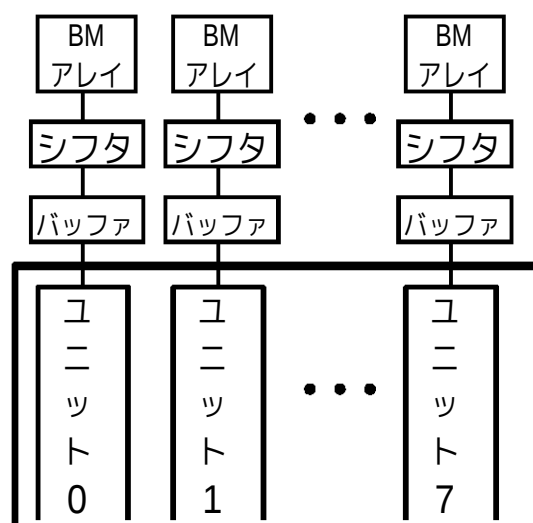


図 6.20: バッファを付加した分散型参照画像メモリ構成

6 バッファ付加による低電力化

メモリユニットと BM アレイ間にバッファを設け，メモリユニットの稼働率を下げる低電力化構成を検討する (図 6.20)．また，前提としてメモリユニット構成は 7.1 節で面積当たりのバンド幅が最善である 32 画素幅を 4 画素幅のメモリマクロ 8 個で分割する構成で，バッファはメモリユニットよりも低電力なものであるとする．メモリユニットからバッファに画素を転送する際に，ブロックマッチングに必要な 18x1 ライン，10x2 ラインに加えて周囲の画素も同時に転送し，その周囲の画素もバッファに格納することで，ヒット率向上を図る．余分に転送できる画素の例を図 6.21 に示す．スクエアサーチは 2 次探索では最大 3 回，3 次探索では最大 4 回で打ち切られ，初回のスクエアサーチで参照画像を余分にバッファに格納することにより，2 回目以降のスクエアサーチで必要になるメモリユニットからのアクセスをかなりの割合で省くことができる．32 画素幅を 4 画素幅のメモリマクロ 8 個で分割するユニット構成の合計アクセス幅は 32 画素幅である．よって，14 画素幅を余分に転送できる．5.1 節で分散型参照画像メモリは 90 度回転した探索範囲を格納しているため，縦方向に余分にデータ転送が可能となる．縦方向に余分に転送できる画素幅を表 6.2 に示す．表で示す数値の負は上方向，正は下方向に余分に転送できる画素である．バッファに格納する画素は初回のスクエアサーチに

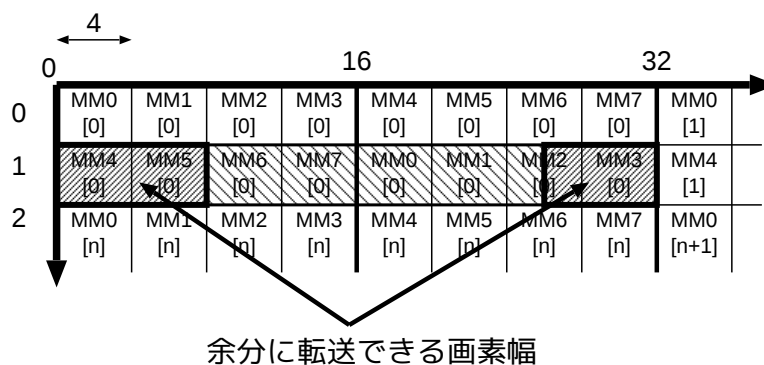


図 6.21: 余分に転送できる画素

表 6.2: 余分に転送可能な画素幅

18x1 ライン	[-7, 7]	[-6, 8]	[-8, 6]	[-9, 5]	[-5, 9]
10x2 ライン	[-3, 3]	[-4, 2]	[-2, 4]	[-1, 5]	[-5, 1]

用いられる画素幅である 16x16 のマクロブロックのブロックマッチングに必要な 18x1 ラインとその周囲の画素をバッファに格納する．この画素を格納して，バッファのヒット率を測定する．

7 評価

7.1 メモリユニット評価

5.2 節で提案した3つのメモリユニット構成のメモリ面積の評価を示す．今回，40nmCMOSのメモリマクロジェネレータを用いて評価を行った．探索範囲(1024x512)を5.1 節で提案した分散型参照画像メモリに格納し，メモリユニット8 個，アクセス競合回避機構に用いられる重複はしないことを前提に面積評価を行った．また，1 画素幅のメモリマクロ36 個を用いた構成では探索範囲外にもメモリマクロが割り当てられるが，その探索範囲外にも割り当てられるメモリマクロも加えることとする．探索範囲外に割り当てられるメモリマクロは

$$\text{探索範囲 (縦)}/1 \text{ 辺のアクセス幅} = 512/6 = 85.3333$$

のように探索範囲内にメモリマクロが切りよく割り当てられるわけでないため，この探索範囲外に配置されるメモリマクロもメモリユニットに含むこととする．この場合では，探索範囲の縦幅を割った値が85.333 となるので $6 * 86 = 516$ ，横幅では 170.666 となるので $6 * 171 = 1026$ となるため，探索範囲 1026×516 に1 画素幅のメモリマクロ36 個をメモリマッピングする．3つのユニット構成の合計メモリ面積，合計容量の調査を行った結果をそれぞれ図7.22, 表7.3 に示す．

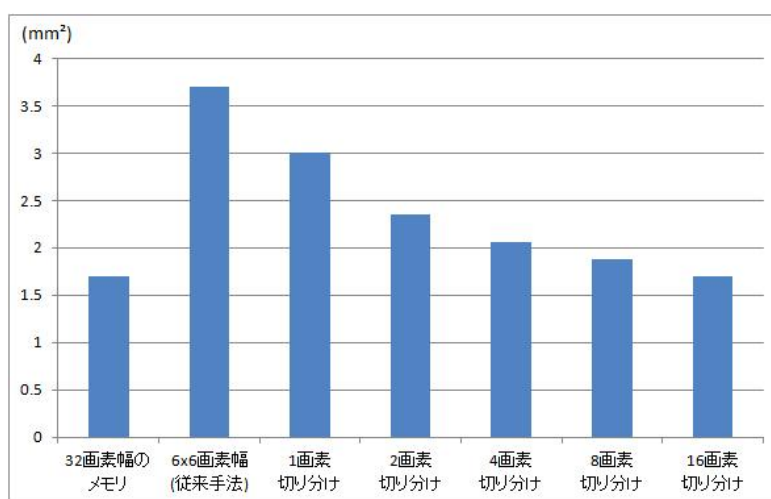


図 7.22: メモリ面積

表 7.3: メモリユニット構成別の合計容量

ユニット構成	容量 (KB)
32 画素幅のメモリ	512
1 画素幅のメモリマクロを 36 個用いる構成	528
32 画素幅を複数のメモリマクロで分割する構成	512

32 画素幅のメモリが最小，1 画素幅のメモリマクロを 36 個用いる構成が最大，複数のメモリマクロで分割する構成が分割幅が大きくなるごとにメモリ面積が小さくなっている．これは，メモリマクロ個数に比例してメモリ面積が増加しているためである．最小の 32 画素幅のメモリでは 1 個のメモリマクロを使用しており，1 画素幅のメモリマクロを 36 個を用いたメモリ構成はメモリ面積は 1 番大きくなっている．また，メモリ容量では 1 画素幅のメモリマクロを 36 個用いる構成が探索範囲外の画素を格納しているため，容量が増大している．

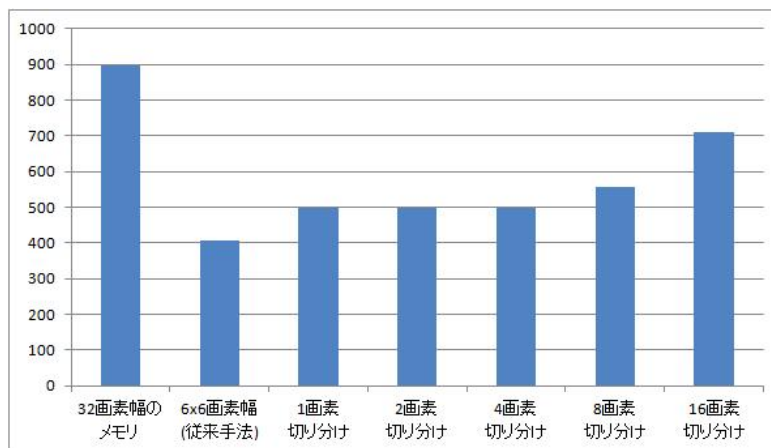


図 7.23: 動き検出に必要なアクセスサイクル数

動き検出を行うのに必要なアクセスサイクル数を図 7.23 に示す．32 画素幅のメモリは整列化制約の影響を受けるため，所要サイクル数が増大している．1 画素幅のメモリマクロを 36 個用いる構成では，1 サイクルで BM アレイに供給できる画素が他の構成を比較して多いため，所要アクセスサイクル数が少ない．複数のメモリマクロを用いる構成では分割幅が 8 画素幅より大きくなると，整列化制約の影響を受け，一部で BM アレイ

に供給するのに2サイクル要するため、所要サイクル数が増大している。

今回、このメモリ面積と所要アクセスサイクル数を考慮して、面積当たりの実効バンド幅の評価を行った。面積当たりの実効バンド幅は

$$\frac{1}{\text{メモリ面積} \times \text{所要アクセスサイクル数}}$$

で計算を行い、32画素幅のメモリを1で正規化している。結果を図7.24に示す。この結果より、32画素幅を4画素幅のメモリマクロを8個用いる構成が面積当たりの実効バンド幅を最も優れていることが分かる。4画素幅で分割する構成は、整列化制約の問題を受けずに18x1ライン、10x2ラインを1サイクルでBMアレイに供給することができ、メモリ面積の増加も抑えられているためと考えられる。1画素幅のメモリマクロ36個用いた構成と比較すると、所要アクセスサイクル数は20%程増加するものの、メモリ面積は40%削減することができ、面積当たりのバンド幅では約1.4倍の改善を得ることができた。面積当たりの実効バンド幅が大きければ、メモリ面積を抑えつつ、性能を高めることができると考えられる。よって、複数のメモリマクロを用いてメモリユニットを構成すれば、同様のメモリ面積のメモリよりも性能を上回るメモリが作成できる。

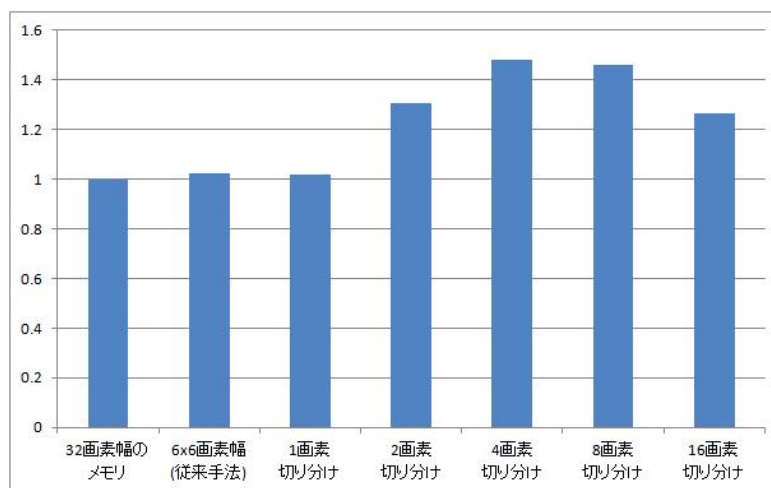


図 7.24: 面積当たりの実効バンド幅

表 7.4: バッファのヒット率 (Airplane)

格納領域数	$[\pm 7]$	$[-6, 8]$	$[-8, 6]$	$[-9, 5]$	$[-5, 9]$
1 領域	39.3%	40.2%	37.5%	36.2%	41.2%
2 領域	85.3%	87.1%	83.2%	83.0%	89.0%
3 領域	89.3%	91.2%	87.0%	86.8%	93.6%

表 7.5: バッファのヒット率 (Bronze)

格納領域数	$[\pm 7]$	$[-6, 8]$	$[-8, 6]$	$[-9, 5]$	$[-5, 9]$
1 領域	54.9%	56.0%	53.9%	53.5%	57.0%
2 領域	75.3%	76.8%	73.9%	73.4%	78.1%
3 領域	80.7%	82.2%	79.0%	78.6%	83.8%

7.2 バッファのヒット率

6 章で提案したバッファ付加による低電力化構成のバッファのヒット率を示す．評価するメモリユニット構成は，7.1 章で評価を行ったメモリユニット構成の中で面積当たりのバンド幅の値が高かった 32 画素幅を 4 画素幅のメモリマクロ 8 個用いるメモリユニット構成である．今回，H.264 の参照ソフトウェアにバッファを付加したと仮定して，18x1 ラインとその周囲の画素をバッファに格納する場合のバッファのヒット率のシミュレーションを行った．使用した参照ソフトウェアは，当研究室で用いられている h.264 拡張 MEv2.5 である．バッファに格納する探索領域として，複数ある候補ベクトルの内，上位 3 つの 20x32 の探索領域を格納する．シミュレーションの結果のヒット率を表 7.4，表 7.5，表 7.6 に示す．また，平均ヒット率を表 7.7 に示す．このシミュレーション結果より，バッファに上位 3 つの候補ベクトルの探索領域を格納すれば，動きの激しい horse_race

表 7.6: バッファのヒット率 (Horse_race)

格納領域数	$[\pm 7]$	$[-6, 8]$	$[-8, 6]$	$[-9, 5]$	$[-5, 9]$
1 領域	25.8%	26.2%	25.2%	25.0%	26.7%
2 領域	51.0%	51.8%	50.1%	49.8%	52.7%
3 領域	62.7%	63.8%	61.5%	61.1%	64.9%

表 7.7: バッファの平均ヒット率

格納領域数	Airplane	Bronze	Horse_race
1 領域	39.0%	55.1%	25.8%
2 領域	85.6%	75.5%	51.1%
3 領域	89.7%	80.9%	62.8%

でもバッファのヒット率が 60%以上になる．よって，メモリユニットからのアクセスを半分以上省くことができ，メモリユニットの稼働率を下げる
ことが可能となり，参照画像メモリの低電力化が図れることが分かった．

8 あとがき

本論文では、疎密階層探索における2次と3次探索にスクエアパターンサーチを用いる動き検出に対応する分散型参照画像メモリ構成を提案した。

提案した分散型参照画像メモリは3x3スクエアパターンサーチを用いて16/8画素幅サブブロックの全方向の探索を行うのに必要な18x1ライン、10x2ラインをそれぞれ1サイクルで供給可能な構成となっている。この構成により、従来のブロック単位でのアクセスが可能なメモリ構成と比較して、動き探索で用いるブロックマッチングに必要なアクセスサイクル数が20%増加するもののメモリ面積を45%削減でき、メモリ面積当たりのバンド幅は約1.4倍の改善率を得た。参照画像メモリとブロックマッチングアレイの間にバッファを組み込み、初回以降のブロックマッチングをバッファ内のデータで行う低電力化構成ではヒット率は60%を超える結果を得た。これにより、メモリユニットの稼働率を下げ、参照画像メモリの低電力化が可能な事が分かった。しかし、メモリユニット構成ではメモリ面積は低減したが、ブロックマッチングに必要なアクセスサイクル数が増加した問題点がある。これは、スーパーハイビジョン符号化の実時間処理には膨大な処理量を要するため、その分ハードウェアコスト増加は避けられず、ハードウェアコストと処理速度がトレードオフの関係になっているためである。そのため、提案したメモリユニット構成以外のメモリマッピング法やメモリ面積を抑えつつ、アクセスサイクル数を低下させる構成法を考えていく必要がある。また、提案した分散型参照画像メモリにおいてアクセス競合を起こさずに並列処理が行える重複幅の設定や隣接ユニットへのアクセスを考慮した参照画像メモリ構成の比較、メモリユニット実装による消費電力評価なども今後の課題となる。

謝辞

本研究を遂行するにあたり，日頃から御指導，御助言を頂きました近藤利夫教授，大野和彦講師，佐々木敬泰助教に感謝いたします．また，研究に協力していただいた計算機アーキテクチャ研究室の方々に感謝の意を表します．

参考文献

- [1] 近藤, 小林, 平松, 佐々木, 大野, “照合用拡張テンプレートを複数併用する階層型動き探索方式”, Proc. 電子情報通信学会総合大会論文集, pp22, March, 2005.
- [2] Tung-Chien Chen, Yu-Han Chen, Sung-Fang Tasai, Shao-Yi Chien, and Liang-Gee Chen, “Fast Algorithm and Architecture Design of Low-Power Integer Motion Estimation for H.264/AVC”, IEEE Transactions on Volume 17, PP 568 - 577, Issue 5, May 2007
- [3] Jarno Vanne, Eero Aho, Timo D. Hamalainen, and Kimmo Kuusilinnä, “A Parallel Memory System for Variable Block-Size Motion Estimation Algorithms” IEEE Transactions on Circuits and Systems for Video Technology, Vol.18, No.4, April 2008
- [4] 小林, 平松, 佐々木, 大野, 近藤, “拡張テンプレートを複数併用するHDTV 用 4 画素精度探索動き検出器の構成”, 電子情報通信学技術研究報告, pp.19-24, Dec, 2005
- [5] 大久保 (監修), 角野, 菊池, 鈴木:改訂版 H.264/AVC 教科書,(2006)

A 実験手順

A.1 ヒット率のシミュレーションの実装手順

バッファのヒット率のシミュレーションは H.264 拡張 MEv2.5 に対して行った。

- mv-search.c にある関数 SquareSearch1_kon のスクエアサーチ開始位置で buf_right_x, buf_left_x, buf_up_y, buf_down_y に, ex_center_x, ex_center_y の上下左右の位置を記録する。このとき, buf_right_x, buf_left_x, buf_up_y, buf_down_y は, バッファに格納される探索の左上, 右上, 左下, 右下の座標位置が設定される。ex_center_x, ex_center_y の値はスクエアサーチ開始点のマクロブロックの左上の座標位置である。
- スクエアサーチの 9 点の探索を終える度に, 探索結果である座標 ex_cand_x, ex_cand_y を左上とするマクロブロックに対して, 次のスクエアサーチに必要な周囲 1 画素を含む探索領域がバッファ内に格納されているか判定を行う。