

修士論文

題目

拡張テンプレート複数併用法を用
いる動き探索の高効率化

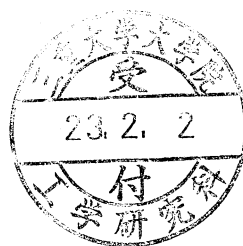
指導教員

近藤 利夫 教授

2010 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

大矢 祐希 (409M507)



内容梗概

符号化対象画像の高精細化の進展の中で，構成画素数の2乗に比例して増大する動き探索の膨大な演算量は常に符号化処理装置実現上の最大のネックとなってきた．特に，2015年にも試験放送の開始が予定されているスーパーハイビジョンへの対応は，ハイビジョンから精細度が16倍にも飛躍するため，大幅な演算量あるいはハードウェア規模の低減が必要不可欠になっている．このような状況の中，さまざまな動き探索の演算量を低減する手法が提案されてきた．

当研究室では拡張テンプレートを複数併用する階層探索法が提案し，画質を維持したまま演算量を低減できることを示した．しかし，階層探索の1次探索部分では動きの激しいシーケンスで演算量が増大してしまい，ハードウェア規模低減の障害となっている．

本論文ではこの問題の解決のために探索範囲拡大を動的に制限する演算量低減法と，動き探索に使われる画像のビット幅を切り詰める演算器規模低減の手法を提案する．これらの提案手法を適用した場合のハイビジョン画像に対する符号化効率と演算量の評価を行なった結果，階層探索部の半分程度を占める1次探索部を従来の1/5程度に演算規模を低減しながら，圧縮効率を低減前と同等に維持できることを確認した．

Abstract

While the progress of encoding an ultra-high-resolution image , the amount of the operation to motion estimation has always become the problem in the encoding processor achievement. In such a situation , the technique for decreasing the amount of the operation of the motion estimation has been proposed. In this laboratory , the hierarchical search that used Multiple Expanded templates (MET) is proposed. However, the amount of the operation increases in the first search part of a hierarchical search .

In this paper , author have proposed the technique to limits the search range for decrease the amount of operation , and to cut the width of bits of the image for decreasing hardware scale. As a result of using these two techniques, the opelation scale was decreases.

目次

1	はじめに	1
2	スーパーハイビジョン符号化の要求条件	3
2.1	スーパーハイビジョンの概要	3
2.2	符号化の演算量	3
3	従来の動き探索手法	5
3.1	追跡型動き探索	5
3.2	階層探索法	5
4	拡張テンプレート複数併用法とその問題点	6
4.1	精度低下抑止の原理	6
4.2	拡張テンプレート複数併用法の問題点	6
4.2.1	スーパーテンプレート単位での一括探索	7
4.2.2	探索範囲の適応的設定	8
4.2.3	ビット切り詰め	8
5	演算数制御による演算量低減	10
5.1	可変探索範囲の提案	10
5.2	演算回数制御	11
5.3	段階的制御を行なう演算回数低減手法	12
6	ビット切り詰めによる演算器規模の低減	13
6.1	微分画像を用いるビット切り詰め法	13
6.2	微分画像に輝度平均値を併用するビット切り詰め法	14
7	評価実験	15
7.1	評価環境	15
7.2	実験結果	15
8	まとめ	16
	謝辞	16
	参考文献	17

図 目 次

2.1	スーパーハイビジョンと従来の解像度の比較	3
3.2	3 階層探索の概略図	6
4.3	拡張テンプレートの形状	7
4.4	スーパーマクロブロックの形状	8
4.5	隣接ブロックと SAD 結果を共有する形状	9
4.6	3 種類の形状のテンプレート	10
4.7	隣接マクロブロックの動きベクトル	11
6.8	元画像 8 ビット	13
6.9	エッジ画像 2 ビット	14
6.10	輝度平均値画像 1 ビット	15

表 目 次

2.1	標準画像とハイビジョン画像の1ブロック当たりの演算量の比較	4
5.2	制御を行なわない演算量低減時のビットレート	12
5.3	段階的制御を行なった演算量低減時のビットレート	12
7.4	段階的制御を行なった演算量低減時のビットレート	16
7.5	ビット切り詰めによるビットレート比較	16

1 はじめに

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、この符号化方式は従来のものと比較して複雑であり、ハードウェアの増加による製造コストや消費電力が増大が問題となっている。特に、動き補償処理は全体の 80 % 以上を占めており、演算量の低減が求められている。また、現在ではスーパーハイビジョン放送等の研究が行なわれており、さらに動き補償処理が増えてしまうことが予想され大幅な演算量の低減が求められている。

この動き補償では、動画像内の 1 フレームを $N \times N$ サイズのマクロブロックに分割し、その前後のフレームのブロックと比較する。そして、最も類似する部分を検出し、そのブロック同士の変移量である動きベクトルを求める。これにより、フレーム間の類似性を利用して、データを差分値と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出来る。マクロブロック同士の画素を比較する時、対応画素間の差分値の総和である SAD を算出する。このブロックマッチングでは非常に大きな演算回数を必要とする。この計算を式 (1) に示す。

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

この問題に対して、演算回数を低減する手法が広く研究されている。

その代表的な手法としてダイヤモンドサーチなどの追跡型の探索方法で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。これらの方法に加え、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、さらに演算量を低減した UMHexagonS, EPZS などが現在ソフトウェアエンコーダを中心に広く利用されている。しかし、この方法は、既に求めた動きベクトルを利用して探索を行なうため、利用した動きベクトルが正しくなかった場合、探索回数が増えてしまう。そのため、利用前の探索結果によって探索効率が大きく異なる。また、これらの追跡型の探索では、参照画像の再利用が難しいため、ロードネックが発生し、並列度の向上が困難である。

一方で、階層探索は密な探索を行なっているため、隣接する画像データの再利用性が高まり、参照画像の再利用性が高い。

階層探索は、探索に利用する画像を縦横 2 分の 1 に縮小してから大まかな探索を行なった後、2 次探索と 3 次探索で細かな探索を行なうことで、演算量を低減している。ただし、1 次探索では探索に用いる画像を縮小し

ているため情報が欠落しているため、探索精度が低下してしまう問題がある。

当研究室では階層探索に拡張テンプレートを用いることで画質を改善した。一方で拡張テンプレートを用いることで演算量が増加してしまう問題がある。そこで、本論文では拡張テンプレート複数併用法を利用した階層探索を用いた動き探索の改良法を提案し、1次探索の探索精度を低下させずに演算器の規模を減らすことを目標とし、具体的には、階層探索の1次探索部の探索範囲を制御することで演算回数を低減し、さらに、探索に使われる画像の画素値を従来の8ビットから3ビット相当に低減することでハードウェア規模を低減する手法を提案する。

2 スーパーハイビジョン符号化の要求条件

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映像の解像度規格であり，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) さらにフレームレートも 60fps とハイビジョンの 2 倍ある．このため，データ量が 24Gbps と非常に大きくなってしまふ．また，この莫大なデータ量を削減するために演算数も膨大になっている．特に動き探索では工夫無しではハイビジョンの 256 倍の演算量が必要となる．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，まだ実用的なレベルで高探索精度で低データ転送レートと演算量低減を両立できていない．

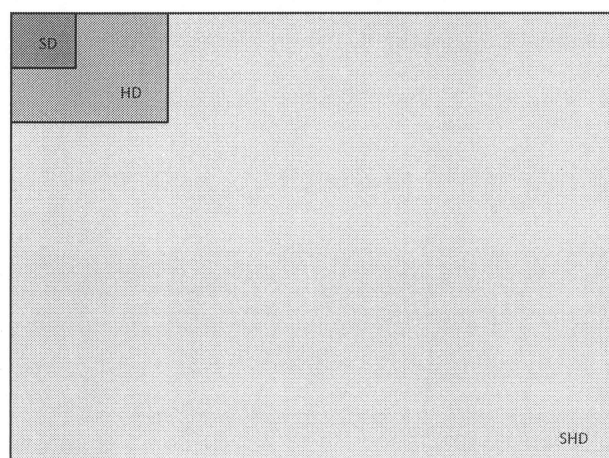


図 2.1: スーパーハイビジョンと従来の解像度の比較

2.2 符号化の演算量

今回，標準画像 (640×480) とハイビジョン画像 (1920×1080) のそれぞれの画素数と演算量の比を利用して，スーパーハイビジョンの演算量を推定する．符号化の演算量の大半を占める動き探索の演算量は一般にフレー

評価画像 解像度	HorseRace	soccer	Whale
SD	58000	49992	23244
HD	372460	329480	79132

表 2.1: 標準画像とハイビジョン画像の1ブロック当たりの演算量の比較

ム構成画素数の2乗に比例する。なぜなら、探索が必要なフレーム内マクロブロック数が画素数に比例し、さらに、マクロブロック毎の探索に必要な探索点数（探索範囲の構成画素数）が画素数に比例するからである。今回はこの推定を元にスーパーハイビジョンの演算量を推測する。表(2.1)は、1マクロブロック当たりの計算量の平均値を表している。検証に利用する画像として、ITE 標準画像の中で、動きの大きいHorse_race,Soccerと動きの少ないJapanese_roomを使った。また、検証に用いるパラメータでは探索範囲をSD画像,HD画像ともに最大384で設定し、探索範囲を動的に縮小させるようにした。表(2.1)にあるSD画像とHD画像を比較すると画像の画素数と比例して増加していることがわかる。なぜなら動画の動きはその画像の画素数の平方根と比例して大きくなり、その計算量は画素数と比例するからである。

また、このそれぞれにマクロブロックの増加量を掛けることで画素数の2乗に比例することがわかる。

ここで、HD画像の計算量からSHD画像のマクロブロック当たりの計算量を計算する。

ただし、今回のHD画像では60fpsであり従来の30fpsよりもフレームレートが高く、フレーム間距離が半分になる。そのため、探索範囲は縦横共に2分の1で良く、探索範囲は画素数と比例して16倍から4分の1し、結果4倍として計算する。また、マクロブロックの数が16倍になるので合計でHD画像の64倍の計算量が必要と想定される。このまま64倍のエンコーダを設計した場合、ハードウェア規模が膨大になり、製造コストや消費電力の面でハードウェア化が困難である。そこで、今回の研究ではスーパーハイビジョン符号化のためにハードウェアを大幅に低減することを目標とする。

3 従来の動き探索手法

3.1 追跡型動き探索

追跡型の動き探索にはダイヤモンドサーチやスクエアサーチなどがある。これらの探索方法は、探索中心と探索範囲内の SAD を調べ、最も SAD が小さい点を中心とする探索範囲内の探索を、探索範囲の端にある点での SAD が最小にならなくなるまで繰り返す。これらの探索方法では探索範囲内の探索点を極力減らすことで探索箇所が大幅に減る。さらに、これらの探索方法を利用した EPZS や UMHexagonS などの代表的な動き探索が現在主流となっているが、一方で探索点を疎らに取る探索の高効率化が、参照画像の再利用率低下によるロードデータ量増大を招き、高並列化のネックとなる問題がある。

3.2 階層探索法

これまでに説明した探索手法は画像の再利用が難しかった。一方で、階層探索は探索点を連続的に取るため、隣接する画像データの再利用ができ、ロードネックが緩和され並列化が可能である。

ここで階層探索の説明をする。階層探索では、探索に使用する画像を縦横 4 分の 1 した縮小画像を用いて大まかな探索を行う。そして、1 次探索よりも小さい範囲で縦横 2 分の 1 に縮小した画像を用いて探索を行ない、さらに元の画像でさらに小さな範囲で探索を行なう。この方法では 1 次探索で 4 画素精度を使用することで画像のサイズが 16 分の 1 になり、さらに探索範囲も 16 分の 1 になるため全探索の 256 分の 1 まで演算量を低減できる。ただし、元の画像を用いて行なう探索に比べ 4 画素精度探索では構成画素数が 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。

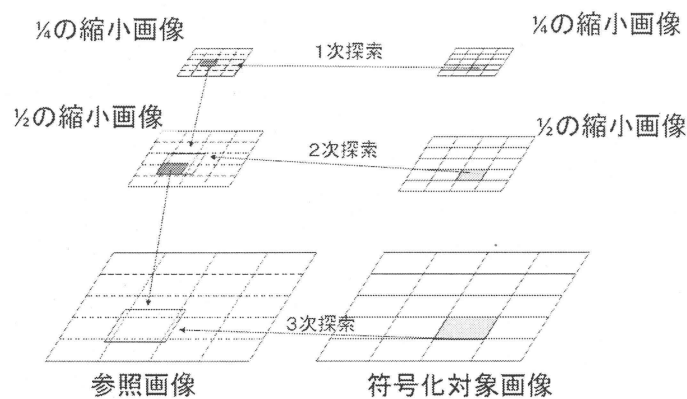


図 3.2: 3 階層探索の概略図

4 拡張テンプレート複数併用法とその問題点

4.1 精度低下抑止の原理

階層探索は、画像の細かな変化を識別できずに精度が低下する問題があった。この問題を解決するために、当研究室では階層探索の1次探索に拡張テンプレート複数併用法を提案されている。この手法では、階層探索の1次探索部で探索対象のマクロブロックの周囲の図(4.5)の様なブロックを利用して探索を行なう。拡張テンプレートを使い隣接ブロックの画素も利用することでブロックの違いを認識できる可能性が高まり精度が改善される。また、図(4.5)のそれぞれの拡張テンプレートにによる検出ベクトルを候補ベクトルとすることで、2次探索以降での探索で正しい動きベクトルを見つけ出す可能性が高くなる。

4.2 拡張テンプレート複数併用法の問題点

しかし、通常のブロックに加えて図(4.5)の8つのブロックも探索する必要があるため、演算量が増えハードウェア規模が増大してしまう。ま

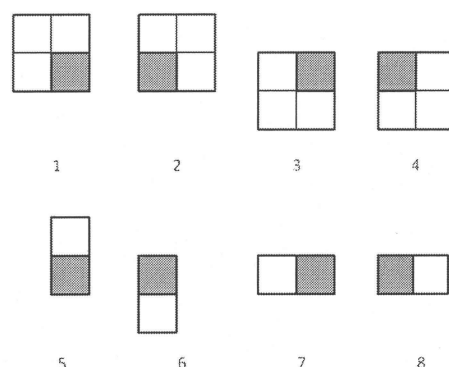


図 4.3: 拡張テンプレートの形状

た、全ての拡張テンプレート形状の計算を行なうと通常の 9 倍の SAD 計算が必要となってしまう。

4.2.1 スーパーテンプレート単位での一括探索

拡張テンプレートによる演算量の増大に対してスーパーテンプレート単位での探索によって演算量を低減している。スーパーテンプレート単位の探索では、図 (4.4) の様に縦横 5x6 の範囲で同時に探索をすることで、並列に演算する。

また、隣接ブロック同士で重複する拡張テンプレートブロック形状の SAD を共有することで計算回数を低減できる。例として、図 (4.5) の様に隣接する Tb1, Tb2, Tb3, Tb4 のそれぞれの拡張テンプレート形状がすべて重複しており、これらの SAD 計算は共有することが出来る。これらすべての拡張テンプレートが共有できた場合、図 (4.6) の 3 種類の拡張テンプレートの形状を計算することですべての方向の形状を計算することが出来る。

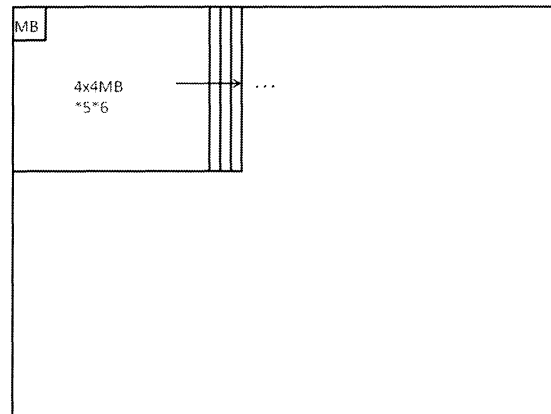


図 4.4: スーパーマクロブロックの形状

4.2.2 探索範囲の適応的設定

探索範囲を動的に変更することで演算回数を低減している。その方法は周囲の MB のすでに求められているブロックの動きベクトルを調べ、その動きの大きさに対応して探索範囲を設定する。周囲のマクロブロックは図 (4.6) にあるような左、上、右上の既に動きベクトルの求まっている MB を利用する。しかし、この方法で探索範囲を設定すると一部のシーケンスでは探索範囲が過剰に大きくなってしまい、冗長な演算が発生してしまう場合がある。そこで、本稿では探索範囲を段階的に制御することで探索の精度を維持しつつ演算量を低減する手法を 5 章で提案する。

4.2.3 ビット切り詰め

従来の動き探索では用いられている画像は 8bit で 256 階調を表現している。しかし、8bit の画像で動き探索を行なうと演算規模が大きくなりハードウェアが膨大になってしまう。そこで、動き探索の演算規模を低減するために使用する画像のビット数を低減し、ハードウェア規模を低減するビット切り詰め法が考案された。特に 2bit 以下ではハードウェア規模を大幅に低減できるが、4bit 以下では探索精度が大幅に低下してし

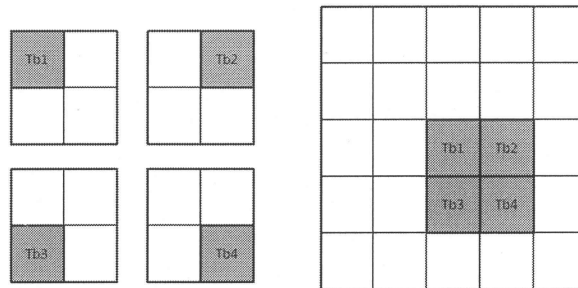


図 4.5: 隣接ブロックと SAD 結果を共有する形状

もう問題がある．そのため，精度の低下しにくいビット切り詰めを行なうには適切な閾値の設定が必要である．そこで，探索精度の低下を抑えたビット切り詰め手法を 6 章で提案する．

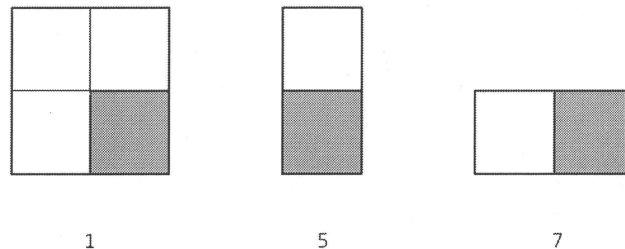


図 4.6: 3 種類の形状のテンプレート

5 演算数制御による演算量低減

5.1 可変探索範囲の提案

動き探索では一定の範囲内を探索する場合、動きの小さな部分でも一定の範囲を探索してしまうため、冗長な演算が起きてしまい、結果的に演算量が多くなってしまう問題がある。そこで、階層探索の1次探索部分では、探索範囲を可変にすることで探索精度を犠牲にすること無く演算回数を低減する手法を実装している。これにより、動きの大きな部分と小さな部分にバラツキがあるものの、平均では、演算回数を大きく減らすことが出来る。

この可変探索範囲では、探索範囲の設定に周囲の動きベクトルを用いることで動画像の動きに適した探索範囲を動的に設定し、動きの小さな部分で演算回数を減らすことが出来る。

しかし、可変探索範囲では、動きの大きな部分で大きく探索範囲を設定する。ハードウェア設計では、この動きの大きな部分の演算量をカバーできるだけの並列度のハードウェアを構成する必要がある。なぜなら、動きの大きな部分でハードウェアの演算量が足りなかった場合、動きベクトル検出に遅延が発生してしまうからである。結果としてハードウェア

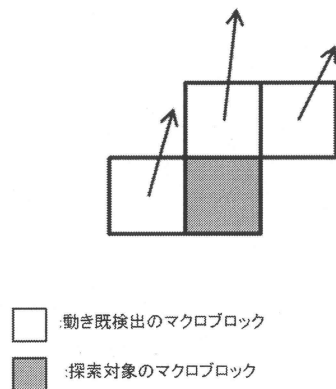


図 4.7: 隣接マクロブロックの動きベクトル

規模が大きくなってしまいう問題がある。

5.2 演算回数制御

従来の可変探索範囲では、動きの大きなシーケンスに合わせてハードウェアを設計する必要があった。そこで今回は、一定量の遅延を許容し、最大演算回数を少なく設定することでハードウェア規模を抑える。このとき、可変探索範囲の大きな部分で演算量が演算の規定値を上回り、遅延が発生してしまう。そこで、2次探索との間にバッファに挿入し、既定値を上回った部分の借りを下回った部分で返せるようにする。それでも、遅延の許容量を越えてしまった場合には、演算を途中で打ち切る。この方法により、演算回数の増減を調整することで探索精度を落とさずに効率的にハードウェア規模を落とす。

今回は、提案する演算量制御手法をソフトウェア上に実装し、圧縮効率の増減を調べた。また、今回の指標は圧縮効率に対応する符号発生ビットレートで表し、評価に使用する画像は動きの大きな画像の HDhorse, HDsoccer, そして動きの小さな HDjapanese とする。その結果として、(表 5.2) に評価値を示す。HDhorse の画像では演算回数を 80000 回に制御した場合、制御無しに比べ、ビットレートが 2.5% 増加してしまう。これは、他の画像

評価画像 最大演算回数	HorseRace	soccer	Japanese
制御無し	100.0%	100.0%	100.0%
40000 回	104.2%	101.1%	100.3%
80000 回	102.5%	100.5%	100.1%
120000 回	101.3%	100.4%	100.0%
160000 回	100.7%	100.29%	100.4%

表 5.2: 制御を行わない演算量低減時のビットレート

評価画像 最大演算回数	HorseRace	soccer	Japanese
制御無し	100.0%	100.0%	100.0%
80000 回	101.4%	100.2%	100.4%

表 5.3: 段階的制御を行なった演算量低減時のビットレート

に比べ horserace の動きが大きく、探索範囲を小さくしたときに、正確な動きを検出できないことが考えられる。今回はこの問題を解決するために、段階的に演算量を制御する手法を提案する。

5.3 段階的制御を行なう演算回数低減手法

5.2 節の手法では、遅延時間が既定値を越えないよう、演算を途中で打ち切り、演算回数を減らす。しかし、演算回数を急に停止してしまうと正確な動きが検出できず探索精度が低下してしまう。この問題に対して、累積遅延が許容量以下の閾値を超えて遅延が発生した場合、探索範囲を小さくし、演算回数を段階的に制御する手法を提案する。

この手法はバッファの空きが無くなる前に事前に探索回数を減らし急激な演算回数の低下を抑える。実際にはバッファの 1/2 以上演算量の遅延が累積したときに演算回数を 320000 回以下に抑え、遅延の増加量を減らす。さらに 3/4 以上累積したときに演算量を 160000 回以下に抑え、さらに遅延の増加量を減らす。この結果を (表 5.3) に示す。表では従来の horse_race の 2.5% 増が 1.4% 増に緩和された。また、この手法による平均演算量は horse_race で 157760 回から約 65472 回に約 60% 低減することが出来た。

6 ビット切り詰めによる演算器規模の低減

従来の 8bit256 階調の画像では演算規模が大きい。ビット切り詰めを 2 ビット以下に下げると演算規模が大幅に低減できる。ビット数を 4 ビット以下にすると探索精度が低下し圧縮率が低下してしまう。代表的な閾値の決定方法の線形量子化やメディアンカット法があるが、4 ビット以下で精度が低下してしまう。そこで、本章では探索精度を維持しつつビット数を 2 ビットに低減する手法を提案する。

6.1 微分画像を用いるビット切り詰め法

微分画像を用いるビット切り詰めは、元の画像に微分フィルタを掛けることで微分画像を作り、さらにビット切り詰めを行なう方法である。(図 6.9) 元の輝度値に微分フィルタをかけることで、画像のエッジが強調され、ビット数が 2bit 以下でも精度が維持できる。当研究室は、この微分画像をさらにメディアンカット量子化によってビット切り詰めすることでさらに精度を改善した。しかし、微分画像は輝度値の情報を削除してしまうために特有の問題がある。具体的には、画像全体の大まかな輝度の変化を識別できない模様とその輝度の変化量が同じ場合、輝度が違っていても同じ形の物体の識別が出来ないという問題である。その結果として一部の画像においては探索精度が低下してしまう。



図 6.8: 元画像 8 ビット

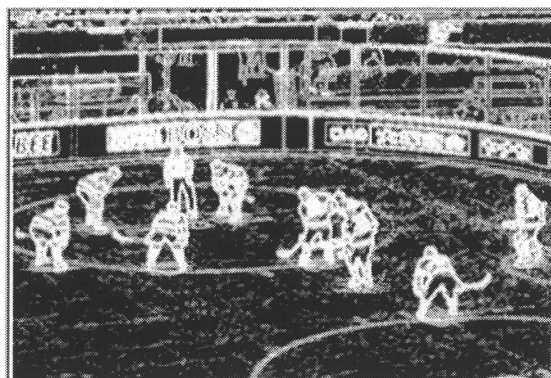


図 6.9: エッジ画像2ビット

6.2 微分画像に輝度平均値を併用するビット切り詰め法

微分画像を用いた動き探索では，探索精度が落ちる画像があった．そこで，当研究室では微分画像に輝度平均値を併用するビット切り詰め法が試された．これは，微分が増が輝度の変化を識別できないという欠点を補うもので，輝度平均値とは横4画素の平均値を取り，それを微分画像と併用して探索を行なうものである．この方法は元の画像から2bitの微分画像と4bitの輝度平均値画像を作り，それぞれのSADを加算したものを動き探索に用いる．(図6.10)

これにより，特定の画像でも精度を改善することが出来た．しかし，輝度平均値を追加する事によって演算器規模が増大する問題があった．そこで輝度平均値を4bitに据置く一方で，サブサンプル率を下げて画素当たりのビット量が増えないようにする．

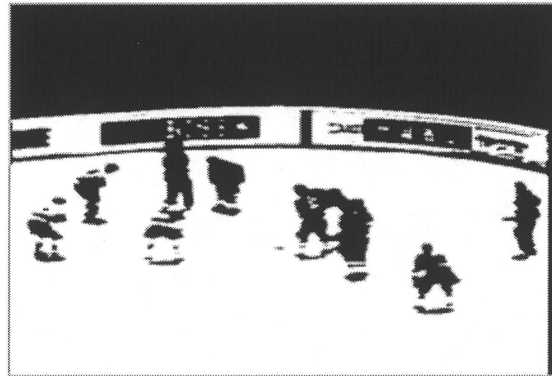


図 6.10: 輝度平均値画像 1 ビット

7 評価実験

7.1 評価環境

H.264 のソフトウェアエンコーダ (JM10.1) に今回の提案手法を実装した。また、量子化パラメータは 28 に設定し、圧縮効率評価を行なった。今回の実験に使用する画像は動きの激しい画像の horse, soccer と動きの少ない JapaneseRoom の合計 3 つの画像を使用し、探索精度の評価はビットレートにより符号の発生量の変化を評価に使う。

7.2 実験結果

表 (7.4) は、演算回数の段階的制御による圧縮率の比較の評価値である。結果を見てわかる通り、動きの大きな HorseRace では圧縮率が若干低下し、1%程度ビットレートが増加してしまっている。しかし、それ以外のデータでは圧縮率は元のデータと同じになっている。この結果より、段階的な演算数制御が適切に機能し、範囲設定の絞りこみが行なえたことがわかった。

表 (7.5) は、ビット切り詰めによる圧縮率の比較の評価値である。表 (7.5) より、ビット切り詰めをしたことによって最大で HorseRace では探索精度が 1%程度低下している。しかし、他の画像ではほとんど探索精度は低下しておらず、探索精度が維持されることがわかった。

評価画像 制御回数	HorseRace	soccer	Japanese
制御無し	100.0%	100.0%	100.0%
80000 回	101.4%	100.2%	100.4%

表 7.4: 段階的制御を行なった演算量低減時のビットレート

評価画像 ビット数	HorseRace	soccer	Japanese
元画像	100.0%	100.0%	100.0%
輝度 4 ビット併用	101.1%	100.2%	100.4%

表 7.5: ビット切り詰めによるビットレート比較

8 まとめ

本研究では動画像符号化にの演算量とハードウェア規模を低減することを目的とし、演算回数の段階的な制御と動き探索に使用する画像のビットを低減することで演算規模低減を行なう手法を提案し、実装と評価を行なった。H.264 のソフトウェアで評価を行なった所、また、演算回数の段階的制御ではビットレートの増加量は 1 % 程度に抑えつつ、平均演算回数を最大 6 割程度低減し、ビット切り詰めではビットレートの増加量を 1 % 未満に抑えることを確認した。演算回数の段階的制御では、比較的動きの大きな horserace でのみビットレートが 1 % 程度増加してしまっているが、これは動きの激しい部分が多く、演算回数を制御して探索範囲を小さくした範囲を一部のシーケンスの動きが上回ってしまったことが原因と考えられる。さらに精度を高めるためにも、動きの大きなシーケンスの範囲を広げ、その動きに適した探索範囲の設定方法を見付けることも今後の課題の 1 つである。

謝辞

本研究を遂行するにあたり、様々なご指導を頂きました近藤利夫教授、大野和彦講師、佐々木敬泰助教に深謝致します。また、本研究に御協力して頂いた計算機アーキテクチャ研究室の皆様に感謝いたします。

参考文献

- [1] 片野 智健, 異形状拡張テンプレート併用型4画素精度ビット切り詰め探索ユニットの設計, 電子情報通信学会技術研究報告. ICD, 集積回路, Vol.107, No.382, pp. 71-76, 20071206
- [2] 大矢 祐希, 佐々木敬泰, 大野和彦, 近藤利夫:”輝度値とその微分値を併用するビット切り詰め型動き探索法”: 平成 21 年度電気関係学会東海支部連合大会 (9 月 10-11 日)(O-014)