

SAD 演算回路における最適な比較ビット使用箇所および ブロックサイズの検討

Study on the best bits to be compared and the best block size for a SAD computing circuit

浜地 亮輔†
Ryosuke Hamaji

山下 茂‡
Shigeru Yamashita

1. はじめに

近年、画像処理をハードウェアで処理することが多く、この技術は、カメラの顔検出やステレオマッチング、動画圧縮といった場面で利用されている。このような画像処理において、特定のパターンを参照画像から検出するために、テンプレートマッチングという手法が用いられている。テンプレートマッチングを行なう上で、画像の類似度を示す指標が複数あるが、ハードウェア上で最も実装が容易で、かつ高速に動作する絶対差総和 (Sum of Absolute Difference:SAD) を用いる場合が多い。SADとは、2つの画像の画素を比較するために、それらの輝度値の差の絶対値を加算し、類似度を算出するものである。

SADを用いた場合でも、テンプレートマッチングは処理に時間がかかるため、処理を高速化するために様々な手法が考案されてきた。Hishamら[1]は、SADの加算途中結果がある閾値を超えた場合、そのブロックでの計算を中断して、次のブロックのSAD計算を始めるハードウェアを設計した。また三浦ら[2]は、ステレオマッチングにおいて、SADの途中計算の結果を再利用することで、処理の高速化を図った。

これらのアプローチの他にも、比較する輝度値（以降比較ビット）のうち、いくつかのビットを無視して計算するというアルゴリズムが存在する。この手法では、一画素の絶対差を算出し、その値の下位ビットを無視して加算するため、高速に動作する。ただし、この手法を用いた場合、最もマッチしている位置を複数検出してしまい、誤差を生じる可能性があり、利用するうえで慎重にならなければならない。これまでに、比較ビットを切り詰める手法を適用した論文がいくつか発表されているが、利用した比較ビットの数や比較画像のサイズ（以後ブロックサイズ）はそれぞれ異なる。これらの研究のアーキテクチャは参考になるが、利用用途や実装環境によって最適な比較ビット数やブロックサイズが異なるため、参考にならない場合が多い。

本稿では、利用する比較ビット数やブロックサイズについて、様々なパターンを利用してSADを計算し、どのような画像において、比較ビット数、ブロックサイズが最適であるかを検証する。そして、単純に利用するビットを減らす場合、どのビットを減らせば良いかを検討する。

本稿は全5章で構成されている。2章では、SADや関連研究について説明する。3章では、実験で用いた利用ビットパターンについて示し、それらの実験結果を示す。4章では、実験結果から得られた情報を元に、考察を行なう。

†立命館大学大学院理工学研究科, Graduate School of Science and Engineering, Ritsumeikan University

‡立命館大学情報理工学部, College of Information Science and Engineering, Ritsumeikan University

5章ではまとめと今後について述べる。

2. 関連研究

画像処理のテンプレートマッチングにおいて、テンプレート画像と比較画像（以後ブロック画像）がどれだけ似ているかを示す方法として、SAD（絶対差総和）がある。SADは、各ピクセルの輝度値の差の絶対値をとり、すべてのピクセルの総和をSAD値とする。SAD値が少なければ少ないほど、2つの画像が似ていると言える。座標 (i,j) における画像Aの輝度値を $A_{i,j}$ としたとき、サイズ $X \times Y$ のテンプレート画像Tとブロック画像IのSAD値Rは式1によって求められる。

$$R = \sum_{i=0}^X \sum_{j=0}^Y |T_{(i,j)} - I_{(i,j)}| \quad (1)$$

テンプレートマッチングでは、SADの他にもSSD（絶対差2乗和）やNCC（正規化相互相関）などいくつかあり、ソフトウェアではSADを利用する例は少ない。しかし、ハードウェアでSSDやNCCを設計すると、高速性を重視するため、組み合わせ乗算回路を実装しなければならない、ハードウェアコストが膨大になる。そのため、多くの場合において、実装しやすく且つハードウェアコストの小さいSADが利用されている。また、多くの画像処理ハードウェアは、8ビットのグレースケール画像を用いている。そのため、本稿でもこの環境を想定して実験を行なう。

ハードウェアでテンプレートマッチングを処理させる場合でも高速化や小面積化が求められ、様々な研究が進められている。Ertürkら[3]は、SAD計算を始める前にブロック画像の平均と標準偏差を求め、1ピクセルのデータを2ビットに変換してからSADの処理を行なうハードウェアを提案した。2ビットで表現することで、加算機削減や絶対差算出方法を改善でき、ある程度の信頼を保ちつつ、SAD計算速度を飛躍的に向上させた。またWujian[4]らは、SAD処理を始める前にブロック画像の分散を求め、分散値が閾値を超えた場合、上位4ビットのみを利用し、そうでない場合は8ビットすべてを利用するアルゴリズムを提案した。このアルゴリズムを用いて、あるブロックにおいて上位4ビットのみ利用してSADを求める条件を満たした場合、1度の絶対差加算で2ピクセル分の加算を行なうハードウェアを設計した。ブロック画像の輝度値に特徴的に変化がある場合、上位4ビットのみを利用してSAD値を算出する。これは、下位ビットの情報が無くても精度をほとんど失わずに、一致する箇所を探し当てることが可能であるからだと述べられている。逆に背景といった、似たような輝度値が多く占める場合、上位ビットのみの利用では

計算誤差が多発するため、全ビットを利用して SAD 値を算出するというアプローチをとっている。

3. 利用ビット箇所の検討と実験

2 章で示した通り、利用する比較ビットを減らして SAD の処理を高速化する研究はこれまでにいくつか発表されてきた。しかしこれらの手法では、前処理に時間がかかるという問題点がある。また比較ビット切り詰めアルゴリズムを用いた研究において、上位ビットが利用される場合が多く、下位ビットが使われることは少ない。これらの既存手法に対して、我々は、画像によっては上位ビットを使わずに、下位ビットを使用するほうが精度を落とさずに算出できるのではないかと考えた。もしこの仮説が正しければ、画像処理ハードウェアを設計するうえで、非常に重要な事項になると思われる。よって、様々な利用ビットのパターンを用いて SAD 処理を行ない、それらの結果を検証した。本章ではその方法と結果を述べる。

3.1 検証した利用ビット箇所の種類

本節では、実験で検証した利用ビット箇所のパターンを名称と共に表 1 に示す。利用したビットを 1 とし、利用しないビットを 0 で表現する。これらの利用ビットパターンを用いて SAD 計算するとき、利用しないビットは加算に使用せず、詰めて計算した。例えば、H1L3 の場合では、上位 1 ビットと下位 3 ビットによる 4 ビット加算を行なう。この方法を利用することで、ハードウェアを設計において加算器の面積を減らすことができ、同時に高速化が期待できる。

3.2 実験内容

実験では、3.1 節で述べた利用ビットパターンを用いた場合、SAD の結果にどのような影響を及ぼすかソフトウェアによるシミュレーションを行なった。実験では、研究用画像データベース (SIDBA)¹ のカラー画像 5 種類を 128×128 ピクセルの画像に縮小し、重み付けグレースケール化²によって変換したビットマップ画像を用いた。図 1 に、実験に使用した画像を示す。1 つの図に対して、その図から切り出せるすべてのブロック画像についてテンプレートマッチングを行ない、SAD 値が 0 になる箇所を算出した。通常、切り出した地点が最も一致する場所であり、

表 1 利用ビットパターンとその手法名

手法名	利用ビットパターン
FULL	11111111
H6	11111100
H4	11110000
H2	11000000
H1L3	10000111
H2L2	11000011
H3L1	11100001
M4	00111100
L4	00001111

¹ http://www.ess.ic.kanagawa-it.ac.jp/app_images_j.html

² RGB 成分のうち、B に重みをつけたグレースケール化手法

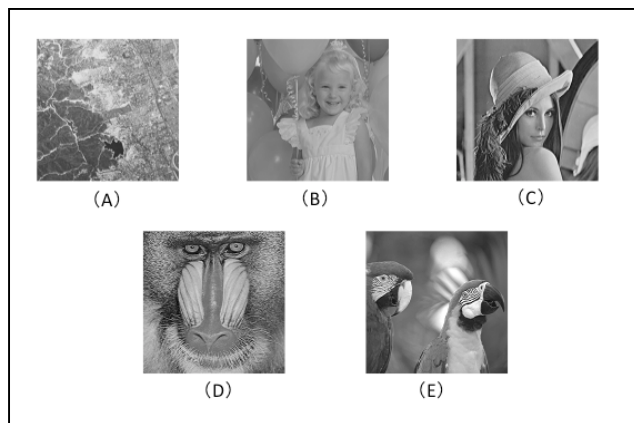


図 1 実験で用いた 5 種類の図

SAD 値が 0 になるが、ブロックサイズや利用ビットによって、SAD 値が 0 になる位置が複数箇所検出されることが考えられる。今回はブロックサイズ 2×2, 3×3, 4×4, 5×5, 6×6 を用いて検証を行なった。

3.3 実験結果

ブロックサイズ 2×2, 3×3, 4×4, 5×5, 6×6 における、利用ビットパターンごとの実験結果を表 2, 3, 4, 5 に示す。各数値は、切り出した 1 ブロック画像あたりの SAD の値が 0 になる検出数の平均値である。例えば、検出数の平均値が 1 である場合、一箇所（ブロック画像を切り出した位置）のみ SAD 値が 0 になったことを示し、他の位置（誤った位置）を検出しなかったことを意味する。つまり、この値が大きくなれば大きくなるほど、精度が悪いことを示す。

以下では、ブロックサイズごとの実験結果について述べる。ブロックサイズ 2×2 では、FULL を用いたとしても、

表 2 ブロックサイズ 2×2 でのブロック画像による検出数の平均値

画像	FULL	H6	H4
(A)	1.00062	1.21414	30.7218
(B)	1.74486	64.2509	972.206
(C)	1.03658	6.74741	206.664
(D)	1.00099	1.21638	35.4103
(E)	1.21142	21.8098	506.811
画像	H2	H1L3	H2L2
(A)	1991.31	2.05313	125.260
(B)	6046.62	4.27447	386.543
(C)	2290.17	2.35333	144.055
(D)	1437.50	1.67567	90.6819
(E)	4289.10	3.21638	269.273
画像	H3L1	M4	L4
(A)	17.2125	5.88610	1.25110
(B)	180.347	122.403	3.22791
(C)	56.8268	22.0380	1.39605
(D)	16.1659	5.85212	1.24180
(E)	96.3528	48.7884	1.78678

表 3 ブロックサイズ 3×3 でのブロック画像による検出数の
の平均値

画像	FULL	H6	H4
(A)	1.0	1.00012	1.56563
(B)	1.00176	13.8862	453.843
(C)	1.0	1.14134	64.9775
(D)	1.0	1.0	3.14562
(E)	1.00226	2.28974	232.537
画像	H2	H1L3	H2L2
(A)	691.069	1.0	2.35210
(B)	4371.55	1.00289	10.4066
(C)	1350.09	1.0	3.66754
(D)	321.394	1.0	1.61652
(E)	3073.21	1.00289	7.00881
画像	H3L1	M4	L4
(A)	1.06575	1.00012	1.0
(B)	4.38196	19.1194	1.00289
(C)	1.79402	1.24137	1.0
(D)	1.06072	1.00037	1.0
(E)	2.72789	2.83371	1.00239

表 4 ブロックサイズ 4×4 でのブロック画像による検出数の
の平均値

画像	FULL	H6	H4
(A)	1.0	1.0	1.0672
(B)	1.0	4.20921	246.784
(C)	1.0	1.00384	25.0641
(D)	1.0	1.0	1.12608
(E)	1.0	1.05580	123.172
画像	H2	H1L3	H2L2
(A)	333.964	1.0	1.00588
(B)	3221.09	1.0	1.08422
(C)	875.518	1.0	1.01433
(D)	94.5215	1.0	1.00089
(E)	2284.10	1.0	1.03456
画像	H3L1	M4	L4
(A)	1.00012	1.0	1.0
(B)	1.01779	4.86841	1.0
(C)	1.00243	1.00435	1.0
(D)	1.0	1.0	1.0
(E)	1.00844	1.06041	1.0

全ての画像において検出数の平均値は 1 にならなかった。
また、画像 (B) , (C) , (E) においては、6 ビットを利用
した H6 よりも、4 ビット利用した L4 や H1L3 の方がより
検出数の平均値が 1 に近いことを示した。4 ビット利用し
た種類の中では、L4 が最も精度がよく、H4 が最も精度が
悪かった。ブロックサイズ 3×3 では、FULL の結果に対し

表 5 ブロックサイズ 5×5 でのブロック画像による検出数の
の平均値

画像	FULL	H6	H4
(A)	1.0	1.0	1.02263
(B)	1.0	1.95993	142.846
(C)	1.0	1.0	10.1220
(D)	1.0	1.0	1.00130
(E)	1.0	1.00195	68.7722
画像	H2	H1L3	H2L2
(A)	191.294	1.0	1.0
(B)	2408.50	1.0	1.00117
(C)	577.218	1.0	1.0
(D)	32.0738	1.0	1.0
(E)	1722.90	1.0	1.00013
画像	H3L1	M4	L4
(A)	1.0	1.0	1.0
(B)	1.0	2.02822	1.0
(C)	1.0	1.0	1.0
(D)	1.0	1.0	1.0
(E)	1.0	1.00195	1.0

表 6 ブロックサイズ 6×6 でのブロック画像による検出数の
の平均値

画像	FULL	H6	H4
(A)	1.0	1.0	1.00502
(B)	1.0	1.27510	84.7530
(C)	1.0	1.0	4.01711
(D)	1.0	1.0	1.0
(E)	1.0	1.0	39.9947
画像	H2	H1L3	H2L2
(A)	113.912	1.0	1.0
(B)	1798.98	1.0	1.00013
(C)	374.458	1.0	1.0
(D)	14.1305	1.0	1.0
(E)	1307.16	1.0	1.0
画像	H3L1	M4	L4
(A)	1.0	1.0	1.0
(B)	1.0	1.28091	1.0
(C)	1.0	1.0	1.0
(D)	1.0	1.0	1.0
(E)	1.0	1.0	1.0

て、H1L3 と L4 が非常に似た結果を示した。これらの種類
による実験の結果、画像 (A) , (C) , (D) では H1L3 と
L4 の検数数の平均値が 1 となり、精度の高さを伺える。
ブロックサイズ 4×4 では、FULL, H1L3, L4 において、す
べての画像で検出数の平均値が 1 となり、誤検知がなかつ
た。H4 以外の 4 ビット利用種類のほとんどが検出数の平

均値 1 であるのに対し、H4 では、画像 (B) において検出数の平均値が 246 と依然高いことがわかる。ブロックサイズ 5×5 では、上記 3 種類に加えて、H3L1 も検出数の平均値が 1 となった。H2L2 も検出数の平均値が 1 に近く、誤検出がかなり減っていることがわかる。ブロックサイズ 6×6 では、H4、H2 以外の種類において、検出数の平均値がほとんど 1 になったが、H4 では、2 桁以上の検出数の平均値を示す画像もあり、誤検出が多いことがわかる。画像 (B) は、風船の部分が似たような輝度値であった。そのため、H4 や H2 で検出数の平均値が非常に高い値になったと考えられる。

4. 考察

実験結果より、上位ビットを利用した SAD よりも、下位ビットを利用した SAD の方が高精度であることがわかった。画像によっては、1 つのピクセルの周辺ピクセルはほとんど似た輝度値であることが多い。そのため、上位ビットはほとんど同じ値になってしまい、下位ビットに差が出てくる。よって、より下位ビットを重視した手法が高精度であると考えられ、結果からもわかるように、FULL の次の高精度であったものが L4 だった。また、6 ビットを用いた H6 よりも L4 の方が高精度であったこともわかる。これらの結果から、上位ビットを重視した SAD 回路よりも、下位ビットを重視した SAD 回路の方が、同じハードウェアコストでありながら、精度がよくなるため、今後の SAD 回路設計においてキーアイデアになると考えられる。

しかしながら、上位ビットが完全に不必要であることはなく、利用できる場面も考えられる。下位ビットだけを用いた場合、上位ビットだけを用いた場合よりも高精度になるが、誤検出時は遠い座標を参照してしまう可能性が高いと考えられる。よって、上位 4 ビットと下位 4 ビットそれぞれで SAD を計算し、最もマッチングした座標集合の積を取れば、一意に決まる可能性が高いと考えられる。このように、上位ビットは不必要ではなく、上位ビットと下位ビットを分離して利用する方法を検討する必要があると考えられる。

また、動画圧縮で用いられているブロックサイズは 4×4 が多く、そのブロックサイズでの実験結果は、基準となる FULL と H1L3、L4 が同じ結果であったため、これらの利用ビットパターンを実際の動画圧縮に用いることができると考えられる。MPEG-2 では、サイズの大きいブロックと小さいブロックを併用してマッチングする場合もある。本実験の結果より、それぞれのブロックサイズによる SAD において、利用するビット数をそれぞれ減らし、高速化を図ることも可能であると期待できる。

5. まとめと今後の課題

本稿では、SAD で利用する比較ビットの利用箇所や、ブロックサイズを変えてシミュレーションを行ない、これまで上位ビットを重視してきた既存研究よりも、下位ビットを重視した方が、状況によっては高精度であることを明らかにした。SAD 演算回路を設計するうえで、この検証は非常に有利に働き、回路の高速化や小面積化に大きく貢献できるものだと考えている。近年、HDTV の普及によって、動画圧縮に用いるブロックサイズが大きくなってきているため、処理時間が膨大になると考えられる。本稿で示した下位ビットのみを利用した SAD 演算によって、ブロック

サイズの大きい HDTV の動画圧縮に貢献できると考えている。

今後、ステレオマッチング処理のハードウェアを想定して、非常に似ているが一致しない画像での検証を行なうことを考えている。また、下位ビットのみを利用して SAD を行なった場合、上位ビットのみを利用した場合よりも、誤検出時により遠い座標を参照する可能性があるため、発生条件や誤検知場所を調査し、その場合の対策を検討したい。そして、並列化や上位ビット組み合わせ等を考慮し、より高速で、より精度の高い SAD 回路を設計し、評価する予定である。

謝辞

本研究は科研費 (23300019) の助成を受けたものである。また、本研究を進めるにあたり、数多くのコメントを頂きました立命館大学大学院の崔英鮮氏に感謝致します。

参考文献

- [1]. C. Hisham, K. Komal, Amit. K. Mishra, "Low power and less area architecture for integer motion estimation," International Journal of Electronics, Circuits and Systems, 2009
- [2]. 三浦清志, 張山昌論, 亀山充隆, "再帰的計算に基づくステレオマッチングと VLSI 化," 電子情報通信学会論文誌 C, Vol. J86-C, No.8, 2003
- [3]. Alp Ertürk and Sarp Erürk, "Two-Bit Transform for Binary Block Motion Estimation," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL 15, NO.7, 2005
- [4]. Zhang Wujian, Zhou Runde, "A High-Throughput Systolic Array for Motion Estimation Using Adaptive Bit Resolution," Proceedings of 4th International Conference on ASIC, 2001