

組み込みプロセッサを用いた動体検出システムの構築と評価

鳥田 貴史[†] 酒井 智也[†] 北村 俊明[†]

近年、モバイル製品においても高品質の動画処理などが要求される一方、稼働時間の延長や、設置型のシステムにおいても電源装置の小型化等、プロセッサの高性能化・低消費電力化が重要な課題となっている。本論文では、組み込み用プロセッサのアーキテクチャとして、シンプルなパイプライン・VLIW・スーパスカラを採用した場合について、性能と消費電力の観点で評価を試みている。このため、ARM アーキテクチャのシンプルなインプリメントである XScale, VLIW 方式を採用した FR-V プロセッサなどに動体検出システムを構築し評価を行った。スーパスカラ方式の組み込み向けプロセッサは適当なものが見当たらなかったため、Pentium4 を用いて測定し、これを元に組み込み向けのスーパスカラプロセッサの性能などを推定した。

その結果、処理の実行時間と消費電力を乗じた消費電力量の面では、VLIW 方式はスーパスカラ方式の 38 %、シンプルなパイプラインでは 58 % に抑えられるという結果が得られた。処理性能と低消費電力性の両立が要求される組み込みプロセッサへの VLIW 方式の高い適性を示すことができた。

The Construction and Evaluation of Moving Object Detection System by Embedded Processor

TAKASHI SHIMADA,[†] TOMOYA SAKAI[†] and TOSHIKI KITAMURA[†]

It is a important point at issue that high performance video processing, extension of operating time, and miniaturization of the power-supply unit on small embedded products.

This paper describes an evaluation of several processor architectures for the embedded systems. In order to evaluate the processor architectures, we developed moving object detection system on several processors. We picked up three processors as a typical architectures; One is XScale which is simple pipeline implementation of ARM instruction set. Second is FR-V which is media-enhanced VLIW architecture. Last is pentium4 which is super scalar architecture. Of course, this is not a embedded processor, but we couldn't find out suitable super scalar embedded processor, so we decided to assume the performance and power consumption from this measurement.

As a result, On the aspect of performance power consumption product, it is obtained that VLIW method become 38 %, simple pipeline method become 58 % of super scalar method. The result shows that VLIW method is most superior to the others for aptitude to the embedded processor demanded coping both energy and performance.

1. はじめに

現在、計算機の急速な性能向上や、ビデオキャプチャカードの普及等により、パソコンを用いた動画再生、画像処理の環境は、一層身近なものになってきたと言える。

しかし、そのような計算機の高性能化に伴い、消費電力の増大が大きな問題ともなっており、プロセッサの高性能化と低消費電力化の両立は重要な課題になっている。このような問題の打開策として、低消費電力性、チップサイズなどの要因をより重視した、様々な組み込み用プロセッサの開発が進められている。

そこで今回、富士通が開発した FR-V ファミリプロセッサの最上位プロセッサコア FR550¹⁾ と、Intel 社が開発した ARM プロセッサ XScale²⁾ を搭載した評価ボード、加えて Intel 社の Pentium4³⁾ を用いて動体検出システムの構築を行い、その性能評価を行った。

FR-V は、VLIW アーキテクチャを採用しており、低消費電力性と、メディア演算命令を用いた最適化による高性能化を両立した仕様が可能である。

ARM は、シンプルなパイプライン構成を持ち回路規模が小さく、低消費電力性に優れている。

Pentium4 は組み込みプロセッサではないが、スーパスカラ方式を採用しており、多段パイプラインによる高速処理が可能である。

本論文では、各プロセッサを用いて動体検出システムの構築を行い、その処理速度、消費電力についての

[†] 広島市立大学
Hiroshima City University

評価を行う。

これは、VLIW アーキテクチャ採用の FR-V、シングルパイプライン構成の ARM、スーパーカラアーキテクチャ採用の Pentium4 を用いて性能評価を行うことで、どのようなアーキテクチャが組み込み用プロセッサに適しているのかを考察することを目的としている。

以下、2章ではシステムの概要として、各プロセッサのスペック、画像処理アルゴリズムについて述べる。3章では、システム全体の構成について述べ、4章では、メディア演算命令を用いたシステムの最適化処理について述べ、5章では、処理速度と消費電力についての評価、考察を行う。最後に6章では、まとめと今後の課題について述べる。

2. システム概要

2.1 本研究に用いる各種プロセッサ

2.1.1 FR-V プロセッサ¹⁾

FR-V プロセッサは、富士通が開発した高性能メディア処理向けマイクロプロセッサである。特色としては、メディア演算命令の搭載によるマルチメディア処理への高い適性などが挙げられる。今回用いた FR550 の簡単なスペックを以下に示す。

- アーキテクチャ： 8 命令同時並列実行 VLIW
- クロック周波数： 400MHz
- キャッシュ： 命令 32KB/データ 32KB

2.1.2 XScale プロセッサ²⁾

XScale プロセッサは、Intel 社のマイクロプロセッサアーキテクチャである。特色としては、1 命令でシフト演算とその他の演算が実行できるという点が挙げられる。XScale の簡単なスペックを以下に示す。

- アーキテクチャ： 固定長 32bit 命令 RISC
- クロック周波数： 400MHz
- キャッシュ： 命令 32KB/データ 32KB

2.2 画像処理アルゴリズム⁴⁾⁵⁾

2.2.1 オプティカルフローの検出

動画像中における移動物体を抽出するために、オプティカルフローの検出を行う。

オプティカルフローとは、フレーム中の各画素間の速度ベクトルを求めたものである。オプティカルフローを求めることによって、その移動物体の各部分が実際にどのような方向にどれだけの速度で動いているかを求めることができる。

今回、オプティカルフローを検出するための手法として、ブロックマッチングアルゴリズムを用いた。これは、注目している画素の周囲の画素値と対応点の周囲の画素値はほぼ同じであるという特徴を用いたアルゴリズムである。

オプティカルフロー検出までの手順を以下示す。

例として、1つの球体が画面の左側から右側へと転

がっていく状況を考える。

- (1) まず、動体領域の抽出を行う。図1の上段の三枚のフレームは1つの球が画面の左側から右側に転がっていく様子を表している。まず、一枚目と二枚目のフレーム間で画像間差分検出・二値化処理を行い、二枚目と三枚目のフレーム間でも同様の処理を行う。その後、二値化処理が施された二枚のフレーム間で論理積演算を行うことによって中間フレームにおける動体が抽出される。
- (2) 次に、テンプレートマッチングによる輪郭抽出を行う。これは、次の手順(3)のブロックマッチングアルゴリズムを用いた計算において、この輪郭抽出された部分のみを計算の対象にすることによって全体の計算量を削減することを狙いとしている。

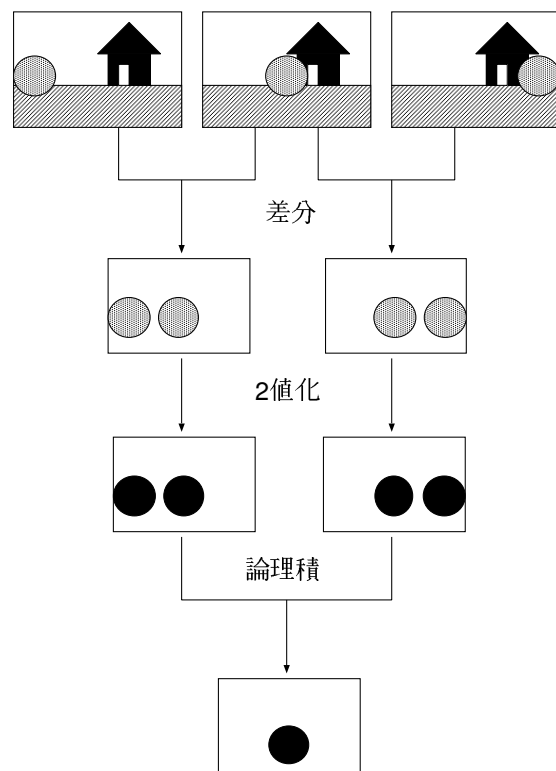


図1 動体領域抽出の流れ

- (3) 最後に、フレーム間で対応点の計算を行う。ここで、前節で述べたブロックマッチングアルゴリズムを用いるが、このアルゴリズムの問題点である膨大な計算量を削減するために、このフレームにおいて先程の手順(2)で輪郭抽出した部分にのみアルゴリズムを適用することによって計算時間の短縮を計っている。

3. 実 装

3.1 システム構成

本研究全体のシステム構成を図2に示す。

FR-V 評価ボードには PCI スロットがあり、ビデオキャプチャカードを直接実装することは可能であったが、これを制御するためのドライバの作成が困難であったこと、ARM の評価ボードには PCI スロットが無く、同一の構成を持ったシステムにした方が良いと判断し、既存の PC でキャプチャした画像データを Ethernet 経由で評価ボードに転送するシステム構成を採ることとした。

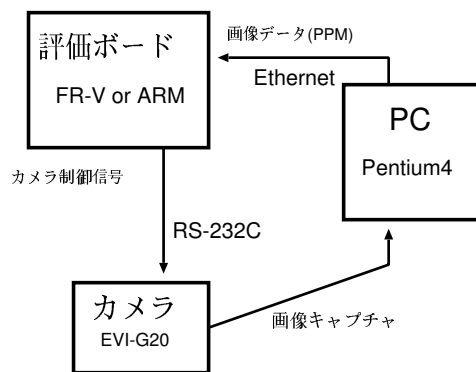


図2 システムの構成

3.2 プログラムの実装

本研究では、移動物体検出システムとして“画像キャプチャプログラム”、“画像データ送受信プログラム”、“オプティカルフロー検出プログラム”、“カメラ動作制御プログラム”の4種類のプログラムを評価ボード (FR550, XScale)、PC (Pentium4) に実装している。

4. メディア演算命令を用いた最適化

FR-V, Pentium4 のシステムにおいては、オプティカルフロー検出の処理時間の大部分を占める、フレーム間の比較処理を行っている関数において、メディア演算命令を用いた最適化も行った⁽⁶⁾⁷⁾。これはコンパイラによる最適化のみではメディア演算命令を使用した最適化処理は行われず、この命令を用いることで大幅な高速化が望めそうであったためである。一方、ARM では同箇所に対して効果を発揮できそうなメディア演算命令が搭載されていないため、ハンドコーディングによる最適化は行っていない。後述するように、ARM の命令セットの特長を生かした最適化はハンドコーディングで行わなくてもコンパイラの-O2 オプションで充分行われていることを確認している。

4.1 FR-V プロセッサにおける最適化⁷⁾

この関数での処理は、7×7のテンプレート・フレー

ムを用いて候補となるフレームとの差の絶対値を求めるといものである。FR-V のメディア演算命令では、画素値の各8ビットの差の絶対値を直接求めることはできない。そこで、メディアクワッド飽和付き加算/減算命令である MQADDHUS/MQSUBHUS (Quad Add/Subtract Unsigned Halfword with Saturation) 命令を利用することによって、符号を判別することなく画素間の差の絶対値を求めることができることを利用した。これによって、条件分岐を省略することができ、処理速度の高速化を実現している。7×7のテンプレート・フレームの1行に対する比較処理を、メディア演算命令を使用してハンドコーディングを行った。図3に、1ピクセル毎の比較処理の流れを示す。まず、1ピクセル毎のRGB値に対して、8ビットのダミーを入れることによって、1ピクセルのデータを4バイトのパウダリに揃える。次に、左右のフレームから1ピクセル分の32ビットデータを、各8ビットのRGB値を各々16ビットに拡張する。そして、R成分、G成分、B成分それぞれについて MQSUBHUS (メディアクワッド飽和付き減算) 命令を用いて飽和減算を行い、各々の成分の差の絶対値を求める。さらに、これらの和を求めるために MQMACHS (メディアクワッド積和演算) 命令を用いることによって、テンプレートフレーム中の画素差の総和を求めることができる。

これらの演算実行レイテンシを隠蔽するために7重にアンローリングして命令スケジューリングを行った。

4.2 Pentium4 プロセッサにおける最適化³⁾⁷⁾

Pentium4 においても同じ箇所に対してメディア演算命令である SSE2 命令セットを用いて最適化を行った。SSE2 を用いたカーネルコードを図4に示す。左右のフレームの4画素分のデータを、MOVDQU 命令によってメモリから XMM レジスタ XMM1, XMM2 にそれぞれロードし、PSADBW 命令によって各画素毎の絶対差を求め、これらを加算した結果を XMM2 に格納する。その後、PADDD 命令によりアキュムレート用レジスタ XMM3 に累積していくことによって、絶対値の総和を求めている。

4.3 XScale プロセッサにおける-O2 オプションによる最適化²⁾⁸⁾

ARM では差の絶対値を直接求める命令が搭載されていなかったためハンドコーディングを行わずに、最適化はコンパイル時のオプション-O2 で行った。最も計算時間が必要である差の絶対値を求める部分がコンパイラの最適化によりどのように最適化されているのかを述べる。図5の上段に示す計算をする場合、最適化前は図5の中段のような記述であった。ここでレジスタ r1, r3 の初期値には、それぞれ tmp_image_R, tmp_image2_R の値が入っている。

asr(算術右シフト)によるマスクを利用することで、rsb(逆減算)の結果の絶対値を求めることができる。もし rsb を終えた時点で r3 が正の数であれば r3 の

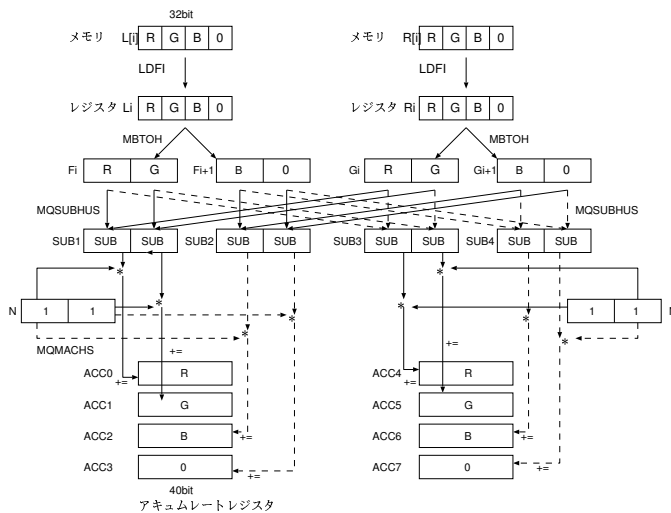


図 3 フレーム間の比較処理の流れ

```

movdqu L[i], L[i+1], L[i+2], L[i+3]
      xmm1
movdqu R[i], R[i+1], R[i+2], R[i+3]
      xmm2
psadbw xmm1, xmm2   xmm2
padd  xmm2, xmm3   xmm3

```

図 4 SSE2 を用いたカーネルコード

31bit 算術右シフトの結果は 0 になるので eor を行っても r1 へは r3 の値がそのまま格納され、つぎの sub でも r1-0 となり r1 の値は変わらない。rsb を終えた時点で r3 が負の数であれば r3 の 31bit 算術右シフトの結果は -1 となり、r3 との eor を行うと r1 へは r3 のビット反転したものが格納されることになる。r3 は 2 の補数表現なので、これは $r1 = -r3 - 1$ に等しい。sub では同じように $r1 = r1 - (-1)$ となり、絶対値の計算が可能になっている。

最適化後は図 5 の下段のような記述になった。ARM プロセッサでは rsblt のようにフラグに応じて命令の条件実行が使えるので、分岐命令を使用せずに記述できる。

```

abs( tmp_image_R[i+y_tmp][j+x_tmp]
    -tmp_image2_R[i+y_tmp][j+x_tmp]);

```

差の絶対値の計算部分.....

```

rsb r3, r1, r3
eor r1, r3, r3, asr#31
sub r1, r1, r3, asr#31

```

最適化前の記述.....

```

rsb r3, r1, r3
cmp r3, #0
rsblt r3, r3, #0

```

最適化後の記述

図 5 XScale における最適化

5. 評価

5.1 結果

FR-V の評価には、プログラムを FR-V 用クロスコンパイラ SOFTUNE で最適化してないものと、最適化レベル 4 で最適化したもの、さらにメディア演算命令を用いて最適化したものの 3 種類を用いた。ARM の評価には、プログラムを最適化してないものと gcc -O2 オプションで最適化したものの 2 種類を用いた。Pentium4 の評価には、プログラムを最適化してないものと gcc -O2 オプションで最適化したうえで SSE2 を用いてさらに最適化したものの 2 種類を用いた。各々のプロセッサ、最適化状況に応じてオプティカルフロー検出に要した時間と消費電力、消費電力量を表 1 に示す。評価に用いた消費電力は、プログラム実行中に電流計を用いて、マザーボードや CPU ボードへのコア用電源供給線の電流値を計測し、電圧を乗じることによって求めている。従って、DC-DC コンバータの効率などプロセッサコアの消費電力そのものに対して多少の誤差を含んでいると考えられ、実際のプロセッサ単体の消費電力は、測定値よりも若干少なくなるものと見込まれるが、比較評価としては、十分な精度を保持していると考えている。また、Pentium4 の動作周波数は 3000MHz であるが、同等のプロセッサアーキテクチャを持つ組み込み向けプロセッサでの性能・消費電力を測定するため、他のプロセッサと同等の 400MHz に正規化を行った。ここで、CPI を考慮しなければならぬが、一番の変化要因であるメモリレイテンシについては、ほとんどのデータがキャッシュに載り、キャッシュミスは着目ループでほとんど無視し得るので動作周波数の正規化は周波数値の単純な比率をもとに算出することで、十分な精度の正規化が行え

ると考えている．また，消費電力量については，リーク電流による消費電力は周波数比例しないため過小評価してしまう可能性が高いが，低クロックにすることで，低速ではあるがよりリーク電流の少ないトランジスタを使用する可能性を見込んで，あえてそのままとした．

5.2 評価

5.2.1 処理速度

表 1 から抽出したプログラムの実行時間のグラフを図 6 に示す．最適化による実行時間の削減率の割合は Pentium4 がおよそ 71 %，ARM がおよそ 52 %であったのに対し，FR-V はおよそ 90 %もの削減率を示すという結果が得られた．これは，FR-V は VLIW 方式を採用しているため最適化を行わないと全く VLIW に命令の詰め込みが行われなため，並列実行の無い 1 命令ずつの逐次実行となっているためである．表 1 に示すとおり，各プロセッサ間でそれほどの処理速度の差異は無いと言える．

5.2.2 消費電力

表 1 に示す通り，FR-V，ARM と Pentium4 の消費電力値の間には大きな開きが存在し，FR-V，ARM 双方の低消費電力性を示す結果となっていることが分かる．Pentium4 と比較すると，FR-V のシステムではおよそ 15 %程度，ARM ではおよそ 16 %程度の消費電力に抑えられるという結果が得られた．改めて組み込み用プロセッサの低消費電力性を示す結果が得られたと言える．また，FR-V と ARM では，若干ではあるが FR-V を用いて構築したシステムが低消費電力性に優れているという結果が得られた．

5.2.3 消費電力量

表 1 から抽出した消費電力量のグラフを図 7 に示す．FR-V，ARM，Pentium4 の比較は，VLIW 方式採用アーキテクチャ，シンプルパイプライン構成アーキテクチャ，スーパスカラ方式アーキテクチャの比較でもある．スーパスカラ方式の多段パイプライン構成は，高クロック化に伴って高速な処理が可能である反面，消費電力が大きくなるという欠点がある．このため，Pentium4 の消費電力量は大きくなっている．これは，他の組み込み向けプロセッサの設計ポリシーと異なり，Pentium4 が低消費電力よりも性能向上に力点を置いた設計を行っているためと言う側面も考えられるが，スーパスカラ方式であると言う点に起因するものと判断している．ARM は，処理能力は Pentium4 に劣るものの，シンプルなパイプライン構成なので消費電力はそれほど大きくない．消費電力量は Pentium4 よりも優れている．VLIW 方式の FR-V は，消費電力が最も小さかった．また，メディア演算命令が効果を発揮し，処理能力においても優秀な結果が得られた．消費電力量において最も優れているのは FR-V であった．また FR-V では，最も使用頻度の高いループのみをメディア演算対応にし，全体アルゴリズムの改良によ

てこの対象ループの呼び出し回数を削減させたため，先行研究⁷⁾ に比べて効果が減っているが，他にも候補となるループが存在するので，それらを最適化することによって処理性能の一層の向上が見込めると考えている．

結果として，処理能力と低消費電力性の両立が求められる組み込み用プロセッサへの，VLIW アーキテクチャの高い適性を示すことができたと言える．

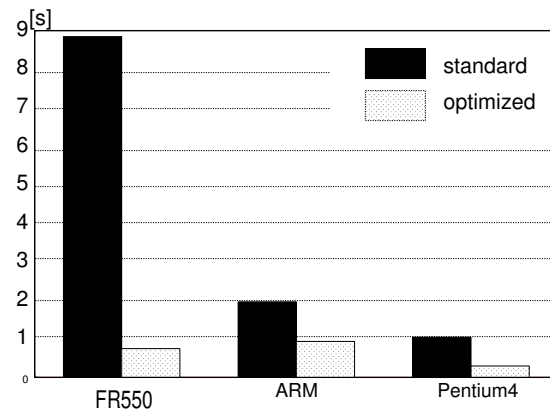


図 6 実行時間

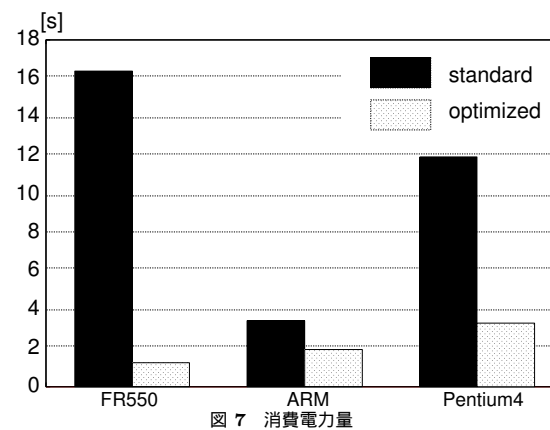


図 7 消費電力量

6. まとめ

本論文では，FR-V，ARM，Pentium4 の各プロセッサを用いて動体検出システムを構築し，その処理速度，消費電力の観点で評価を行った．ARM の優れた命令効率，FR-V のメディア演算命令を用いた命令の並列化など，命令セットの相違も含んだ評価になってしまっているため，純粋なアーキテクチャ間の比較にはなっていないと言える．それでも，FR-V，ARM 両プロセッサの低消費電力性や，Pentium4 の高い処理能力など，各プロセッサの特色を示すような測定結果が得

表 1 実行時間と消費電力 (括弧内は正規化後の数値)

	電圧 [V]	周波数 [MHz]	実行時間 [s]	消費電力 [W]	消費電力量 [Ws]
FR-V	3.3	400	8.85	1.85	16.37
FR-V -O4	3.3	400	0.95	1.85	1.76
FR-V メディア	3.3	400	0.75	1.68	1.26
ARM	5.0	400	1.96	1.75	3.43
ARM -O2	5.0	400	0.94	2.05	1.93
Pentium4	12.0	3000	0.14(1.05)	85.20(11.36)	11.93
Pentium4 SSE2	12.0	3000	0.04(0.30)	82.44(10.99)	3.30

られたので、比較の妥当性は十分に満たしていると考えている。

結果としては考察でも述べたように、優れた低消費電力性を保ちながら高い処理能力が要求される組み込み用プロセッサには、FR-V プロセッサのような VLIW 方式採用アーキテクチャが高い適性を持つという結論が得られた。それは、低消費電力性、メディア演算命令による処理能力の向上、ハードウェアの簡略化による回路規模の小型化といった VLIW の特色が、組み込み機器に求められる要素と一致しているためであると言える。

FR-V のコンパイラである SOFTUNE で生成される命令列では、メディア演算命令を用いたハンドコーディングによる命令列の効率性には遠く及ばない。ハンドコーディングには非常な手間がかかるため、効率の良い命令列を自動生成する新たなコンパイラの開発が望まれているとも言える。

今後の課題としては、前述したように、FR-V のプログラムにおいて他に最適化の候補となっているループ部分で、メディア演算命令を用いて最適化を施した上でのシステムの性能評価や、FR-V、ARM 以外の組み込み用プロセッサ用いたシステムでの性能評価、あるいはモバイル機器への適用を考慮した上でのチップ面積コストの評価などが挙げられる。また、XScale のパイプラインを元にスーパースカラ化したプロセッサを作成して、Pentium4 での代用による評価のあいまいさを無くしていきたいと考えている。ARM においては、version6 でメディア拡張命令がサポートされており、SSE2 と同程度の処理速度の向上が得られると仮定すれば、優秀な評価結果が期待できる。

参 考 文 献

- 1) FUJITSU JOURNAL
<http://journal.fujitsu.com/>
- 2) intel XScale Technology
<http://www.intel.com/design/intelxscale/>
- 3) intel Pentium 4 Processors
<http://developer.intel.com/design/intarch/pentium4/pentium4.htm>
- 4) 飯尾 淳: Linux による画像処理プログラミング, オーム社, 2000.
- 5) 安居院 猛, 長尾 智晴共著: C 言語による画像処理入門, 昭晃堂, 2000.
- 6) 富士通: FR-V Family FR550 Series Instruction Set Manual, 2002.
- 7) 清水 雄歩: 情報処理学会論文誌 Vol44. 「距離画像生成処理におけるメディアプロセッサの評価」, 2003.
- 8) Intel Architecture Software Developer's Manual, 2000 .