1A-06

画像認識処理における SIMD 命令の有効性の評価

高畠 志泰†
株式会社 日立製作所†

1. はじめに

近年、自動運転の実現に向けて、歩行者や先行車などを車載カメラで検出する画像認識処理が重要になってきている。自動運転における画像認識処理の課題は、処理速度の向上と、それに伴うコスト増大を抑えることである。画像認識処理では歩行者や白線など多くの物体を認識するために相応の処理速度を必要とするが、処理速度を確保するために様々な機能を実装することはコスト増大を招くので許されない。

画像認識処理を高速に行なう方法には、高速な汎用 CPU や GPU を用いる方法と、画像認識処理専用のチップを用いる方法があり、車載機器では後者を用いることが多い[1][2][3]。その理由は、後者は画像認識処理に不要な機能を持たないので、比較的安価に製造できるからである。

画像認識処理専用チップを用いた高速化に有用な手段の一つに、データ並列性を利用した SIMD(Single Instruction Multiple Data)命令がある[4]。しかし、汎用 CPU のように SIMD 命令を豊富に実装すると、SIMD 命令用の演算器を備えることとなり、チップ面積が増えコストアップにつながる。コストアップを回避するには、画像認識処理に有効な SIMD 命令だけ実装する必要がある。しかし、近代的な画像認識アルゴリズムについて、どの SIMD 命令が有効であるかを調査した文献は見つかっていない。そこで、本稿では、画像認識処理に有効な SIMD 命令を明らかにし、不要な命令の実装を避けてコストを削減することに貢献する。

2. 評価

2.1 評価方法

SIMD 命令の有効性の評価は、自動運転で重要となる物体検出処理を対象とし、物体検出処理の実装には、画像認識ライブラリ OpenCV[5]を用いる。OpenCV は、オープンソースのライブラリであり、画像認識処理に必要な画像フィルターや色変換、物体検出処理など様々な処理を含む。また、OpenCV 3.1 では、処理の高速化のために組込み関数による SIMD 命令を利用した実装や、GPGPU(General Purpose Computing on Graphics Processing Units)を用いた実装などが含まれている。OpenCV に含まれる物体検出処理のアルゴリズムとその実装方法を表 1 に示す。これらのアルゴリズムのうち、本稿では、HOG を評価対象とした。その理由は以下である。

表 1：物体検出アルゴリズムの実装

アルゴリズム	SIMD	GPGPU
HOG (Histogram of Oriented Gradients)	○	○
LBP (Local Binary Pattern)	×	○
Haar-like	△	○
DPM (Deformable Parts Model)	×	×

- HOG は広く利用されている。
- HOG には SIMD 命令を用いた実装がある。
- LBP には SIMD 命令を用いた実装がない。
- Haar-like には SIMD 命令を用いた実装があるが、スカラ演算に利用しており並列演算を行なっていない。
- DPM には SIMD 命令を用いた実装がない。

LBP と Haar-like に SIMD 命令を用いた並列演算の実装がない理由は、OpenCV の実装では高速化できる箇所がないためと思われる。DPM にも SIMD 命令を用いた実装はないが、アルゴリズムとして内部で HOG を利用するため、実質的には HOG を高速化することで DPM も高速化可能であると考えられる。

評価にあたり、SIMD 命令の生成には、OpenCV 内の SIMD 用組込み関数を使うようにし、コンパイラによる自動ベクトル化は抑止した。また、Intel® CPU の SIMD 命令には SSE (Streaming SIMD Extension)系と AVX (Advanced Vector Extension)系があるが、本評価では SSE 系の SIMD 命令のみを選択した。なぜなら、AVX 系の SIMD 命令の扱うデータ長は 256bit や 512bit と bit 長が長く車載機器にはコストの面で不向きなためである。また、OpenCV のコンパイルには gcc 4.8.4 を、OS には、ubuntu 14.04.5 LTS を使用した。性能測定には Intel® Core™ i5-4590 の 3.30GHz を用いた。また、性能測定に利用したデータには OpenCV と一緒に配布されている “opencv_extra”に含まれるものを用いた。

Evaluation of SIMD Instructions for Image Recognition Processing
†Motoyasu Takabatake, Hitachi, Ltd.

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

2.2 評価結果

HOG を用いた物体検出処理の処理速度を測定した結果、SIMD 命令を利用することで平均 1.4 倍高速化できた。HOG を用いた物体検出処理で実行された SIMD 命令とその種別、および実行回数の比較を図 1 に示す。HOG で実行された SIMD 命令は、全実行命令数の 20%を占めていた。また、実行された SIMD 命令の型は、float 型と int 型、char 型のみであり、short 型、double 型の利用はなかった。実行された float 型の SIMD 命令は 13 種類で、整数型(int 型/char 型)が 13 種類、float 型と整数型間での型変換命令が 2 種類であった。また、実行された SIMD 命令の内訳は、88%が float 型命令で、11%が整数型命令であった。このことから、float 型 SIMD 命令が性能向上に貢献していることが分かる。また、float 型でも MAX/MIN や除算は実行頻度が低く、乗算や加算、データの並び替え、論理演算などの実行回数が多かった。

整数型 SIMD 命令と型変換命令を使わず、float 型 SIMD 命令のみを利用して HOG を実行した結果、SIMD 命令を使わない場合と比べると 1.3 倍の高速化であった。この結果より float 型の SIMD 命令を使うことで 30%の性能向上の効果があり、整数型の SIMD 命令と型変換命令の効果は 10%であることが分かる。

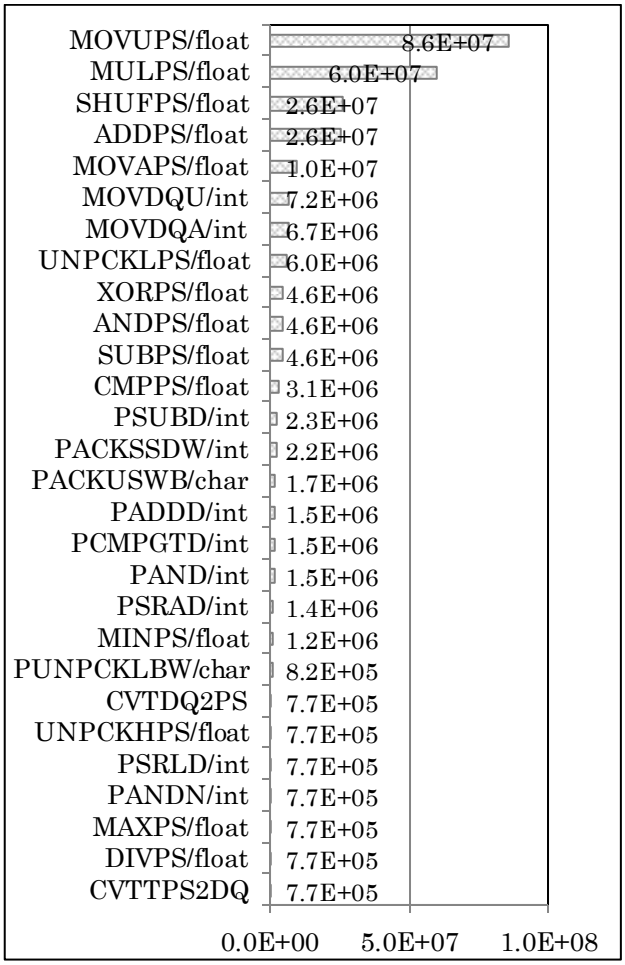

図 1：SIMD 命令の実行回数

3. 考察

評価結果より、画像認識処理専用チップにおいて、OpenCV3.1 の HOG を利用する場合、float 型と int 型、char 型の SIMD 命令の実装を行なうことにより高速化が行え、未使用の short 型や double 型の SIMD 命令の実装を省略することでコスト増大を抑えられる。また、コストをより下げる必要がある場合は、実行頻度の高い float 型の SIMD 命令のみ実装するのも一案といえる。

4. おわりに

車載カメラによる画像認識処理では、評価を行なった HOG を用いた物体検出だけではなく、白線検出や移動物体の検出、追跡など様々な処理が必要とされる。また、近年、画像認識処理においては、認識精度の高さからディープラーニングを用いた方法[6]が注目されている。現在、ディープラーニングには主に GPGPU が用いられているが、GPGPU には画像認識処理に不要な機能や命令が含まれている。今後、他の検出や追跡処理、ディープラーニングなどについても高速化に有効な機能や命令の評価を行なう必要がある。

参考文献

[1] 東芝: 画像認識プロセッサ Visconti™,東芝 ストレージ&デバイスソリューション社(オンライン), 入手先<http://toshiba.semicon-storage.com/jp/product/automotive/image-recognition.html> (参照 2016-11-11).

[2] ルネサスエレクトロニクス: R-Car H3, ルネサスエレクトロニクス(オンライン), 入手先 <https://www.renesas.com/ja-jp/solutions/automotive/products/rcar-h3.html>(参照 2016-11-11).

[3] Mobileye: Mobileye 5-Series, Mobileye (online), available from <http://www.mobileye.com/products/mobileye-5-series/> (accessed 2016-11-11).

[4] Sankaran, Jagadeesh, Hung, Ching-Yu, Kisačanin, Branislav: VE: A Flexible SIMD Coprocessor for Embedded Vision Applications, Journal of Signal Processing Systems, Vol.75, No.2, pp. 95-107 (2014).

[5] OpenCV: OpenCV 3.1, OpenCV(online), available form <http://opencv.org/>, (accessed 2016-11-11).

[6] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25, pp. 1097-1105 (2012).

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.