B-033

メモリ選択を考慮した画像減色処理の CUDA 実装

CUDA implementation of color reduction with effective memory selection.

田中 宏二郎† 目出 雅之† 若谷 彰良‡
Koujirou Tanaka Masayuki Mede Akiyoshi Wakatani

1 はじめに

現在,CPU より高性能な GPU での演算処理を行う GPGPU (General Purpose Graphics Processing Unit)が注目され，CUDA (Compute Unified device Architecture) [1]が普及してきている.

OpenCV [2]も例外ではなく，K-means 法を使いクラスタリングを行う cvKMeans2 関数やモーション履歴画像の勾配方向を求める cvCalcMotionGradient 関数など，パラメータの与え方によって処理時間が大きくなる関数も含まれているため，GPU での演算が注目され，GPU による高速化版の開発がすすめられている[3]．しかし，今まで OpenCV での高速化の面では OpenMP や TBB での高速化が行われてきたが, GPU を利用しての高速化はまだあまり行われていない. そこで本研究では, OpenCV と CUDA を組み合わせた場合どの程度の高速化を行うことができるのかを検証する.

OpenCV を GPU で実行するには，1) 1 つの関数をすべて GPU メモリで行う方法と 2)CPU でデータを管理し，GPU を利用し計算する方法がある. 今回は,既存の OpenCV アプリケーションの継続性を考慮し，2)の方法をとる．本論文では，現時点での OpenCV の高速化を GPU で行うことを目的とし，1 つの方法のもとで実行時間の長い関数のみを GPU で高速化することとする．そこでまず，cvKMeans2 関数に注目し，GPU による高速化を行い，K-means 法を利用したカラー画像の減色処理を行うアプリケーションを対象として評価する．また，メモリ利用により性能の向上が見られるため[4]，今回はグローバルメモリ・シェアードメモリ・コンスタントメモリ・テクスチャメモリを用いてそれぞれのメモリの比較を行った.

† 甲南大学大学院 自然科学研究科 情報システム工学専攻
‡ 甲南大学 知能情報学部 知能情報学科

2 CUDA のメモリ利用

CUDA で利用できるメモリには，グローバルメモリ，シェアードメモリ，コンスタントメモリ，テクスチャメモリなどの記憶領域が存在する.

グローバルメモリは多く使われる GPU メモリである. デフォルトで CPU のメモリから GPU のメモリに書き込み時に利用されるメモリで，コアレス通信等を利用することで高速に通信することが可能になる．シェアードメモリは容量約 16KB でそれぞれのプロセッサごとに存在するため，他のスレッドブロックからは参照できない．オンチップメモリで高速にアクセスが可能である．コンスタントメモリは容量約 64KB でキャッシュにより高速なアクセスが可能で，読み込み専用となっている．テクスチャメモリは 1～3 次元の次元をもち，キャッシュにより高速なアクセスが可能で，読み込み専用となっている.

3 K-means 法と画像減色処理

3.1 画像減色処理

24bit フルカラー画像を 8bit パレット画像に変換するように画像を少ない色で表示することを画像減色処理という. 今回の実験では K-means 法を用いる例として, 一般的なフルーツの画像を 4～1024 色に減色する処理を行った.

3.2 K-means 法

K-means 法とは K 平均法とも呼ばれ，クラスタリング手法の 1 つである．K-means 法では，画像からランダムに色の値を取得し，それをクラスタとして扱い，画像のそれぞれのピクセルの色の値をベクタとして扱う．まず各ベクタの値をクラスタの値と比較し，一番近い色をクラスタの値と置き換える．さらに同じクラスタの値に置き換えられた処理前の画像のベクタの値の平均を求め，これをクラスタ

値とし，再び処理前の画像のベクタの値と比較し，近い値と置き換える．この作業を繰り返し，クラスタの値が変化しなくなれば処理を終了するというように用いた．

これは cvKMeans2 関数においても行われている作業であるが，今回は CUDA を利用するため，新たに K-means 法の関数を作成した．また，GPU で高速化を行ったのは画像の各ベクタの値をクラスタの値と比較し，一番近い色をクラスタの値と置き換えの部分のみである．この部分は他の部分に比べて計算量が多く，また他の部分は if 文を利用することが多く，SIMD 計算機である GPU にはあまり向かないと考えたためである．GPU で処理する部分の中で，クラスタの値は各ピクセルにとって共通の値であり、また読み込みのみである．よってテクスチャメモリやコンスタントメモリに向いていると考えられるが、本論文では，メモリ選択による性能の比較を行う．

4　実験

4.1　動作環境

今回は Atom 搭載の netbook 型ノート PC が GPU を用いることによってどの程度の処理速度を実現できるかを検証する．実験に用いた環境を以下に示す．

- CPU：Intel Atom 330 (1.60GHz×2 コア)
- GPU：NVIDIA　ION
- メモリ：2.00GB
- OS：Windows7 Home Premium
- CUDA：CUDA3.0, OpenCV：OpenCV2.1

4.2　実験結果

図 1 は，減色処理に対する，1) グローバルメモリを用いた GPU 性能，2) シェアードメモリを用いた GPU 性能，3) コンスタントメモリを用いた GPU 性能，4) テクスチャメモリを用いた GPU 性能を示す．GPU 性能は，PC1 で実行した cvKmeans2 関数の実行時間を基準としたスピードアップ値とする．

一番高速に処理を行えたのが，コンスタントメモリでキャッシュにより高速にアクセスが行えたためだと考えられる．次に高速に処理を行えたのがシェアードメモリで，オンチップメモリで処理が行えるがグローバルメモリからデータを読み込む必要があるのでその分コンスタントメモリの方が高速に処理できたのではないかと考えられる．三番目に高速だったのはテクスチャメモリで，64 色まではグローバルメモリの方が高速に処理を行えている．これはテクスチャメモリの利用時に何らかのコストがかかっているためだと考えられる．そのため 64 色以降ではテクスチャメモリの利用回数が増え，テクスチャメモリ利用の効果が見られたためと考えられる．

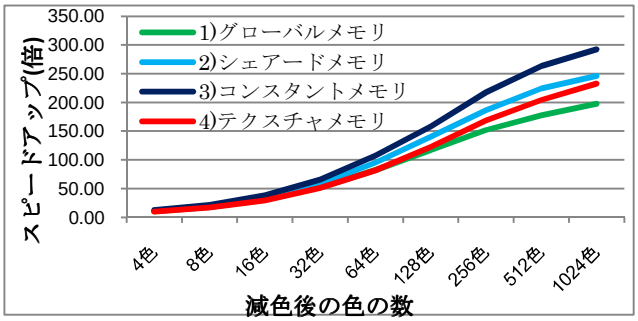

図 1 .速度比較グラフ

5　おわりに

今回のメモリ比較では、コンスタントメモリがグローバルメモリの最大約 1.5 倍高速に処理を行うことが出来たという結果になり，64KB 以内で書き込みがない場合はコンスタントメモリが有効であるが，書き込みがある場合はシェアードメモリが一番有効であると考えられる．このように場合によってメモリの特長から選択を行うことが重要だと考えられる．

参考文献

[1] 青木尊之,額田彰, "はじめての CUDA プログラミング." 株式会社工学社,2009.

[2] http://opencv.jp/

[3] http://www.nvidia.co.jp/object/cuda-gpu-acceleration-opencv-application-press-20100923-jp.html

[4] B. Jang and P. Mistry, “Exploiting Memory Access Patterns to Improve Memory Performance in Data-Parallel Architectures,” IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 1, pp. 105-118, 2011.