

CUDA 利用による画像減色処理の高速化

田中 宏二郎† 目出 雅之† 若谷 彰良‡

† 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

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, 今回はテクスチャメモリを利用したため, テクスチャメモリの有効性についても検証を行う.

2 CUDA

2.1 CUDA とは

CUDA とは, 2006 年 11 月から NVIDIA 社が提供している GPGPU を行うための C 言語の統合開発環境である.

2.2 テクスチャメモリ

GPU のメモリには, 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. その中のテクスチャメモリを今回は使用した.

テクスチャメモリの特徴としては 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だが, GPU の処理中, テクスチャメモリの値の変化が反映される保証がないという欠点もある. 今回の研究の場合, クラスタデータなど処理中は値が変化しないデータがテクスチャメモリに有効と考え利用した.

3 K-means 法と画像減色処理

3.1 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.

今回の減色処理では, 画像からランダムに色の値を取得し, それをクラスターとして扱い, 画像の各ピクセルの値をクラスターの値と比較し, 一番近い色をクラスターの値と置き換える. さらに同じクラスターの値に置き換えられた処理前の画像のピクセルの値の平均を求め, これをクラスターの値とし, 再び処理前の画像のピクセルの値と比較し, 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ピクセルの値をクラスターの

Evaluation of speedup of color reduction by CUDA.

† Koujirou Tanaka, Masayuki Mede and ‡ Akiyoshi Wakatani.

† School of Natural Science, Konan University.

‡ Faculty of Intelligence and Informatics, Konan University.

値と比較し、一番近い色をクラスターの値と置き換えるの部分のみである。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。

3.2 減色処理とは

画像はヘッダーファイル・ラスターデータから構成されている。このうちのラスターデータでは各ピクセルの色が保存されており、R・G・B がそれぞれ 256 段階で、最大 16,777,216 色で画像が表示されている。こういった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 法を用いて一般的なフルーツの画像を 4, 8, 16, 32, 64, 128, 256 色に減色する処理を行った。

4 実験

4.1 動作環境

今回は Atom 搭載の netbook 型ノート PC が GPU 利用によりハイエンドなデスクトップ機並みの性能を得られることを実験で確かめる。実験に用いた環境を以下に示す。

- Netbook 型ノート PC(PC1)
 - 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
- ハイエンドデスクトップ PC(PC2)
 - CPU : AMD Phenom II X4 945 (3.00GHz ×4 コア)
 - メモリ : 4.00GB
 - OS : Windows7 Ultimate
 - OpenCV : OpenCV2.1

4.2 実験結果

1)cvKmeans2 関数(PC1), 2)自作の K-Means 関数(PC1), 3)自作の K-means 関数を CUDA を用いて高速化したもの(テクスチャメモリあり), 4)CUDA を用いて高速化したもの(テクスチャメモリなし),

5)cvKmeans2 関数(PC2)をそれぞれ利用して減色処理を行い、1)を基準としたスピードアップを図 1 に示す。

この実験では GPU を利用することによって最大約 21 倍まで高速化することができ、ハイエンドデスクトップ PC 以上の性能が確認できた。また、3)において GPU での高速化の効果がみられるが 128 色と 256 色で速度が低下しているのは、2)のスピードアップが原因として考えられる。また、3)と 4)を比較し、128 色と 256 色で速度に変化があるのは、テクスチャメモリの利用回数が増え、テクスチャメモリの効果が見られたためだと考えられる。

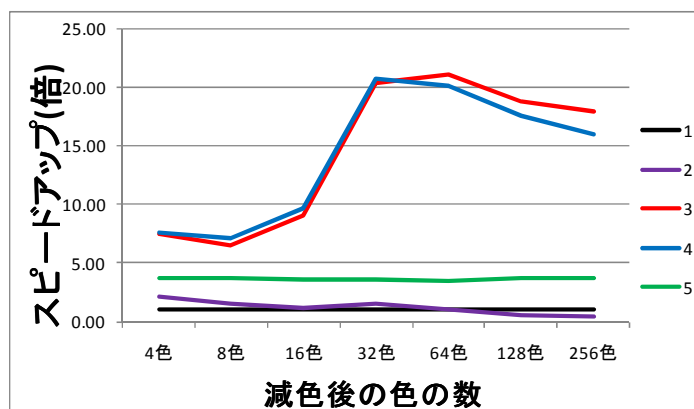


図 1.速度比較グラフ

5 おわりに

今回の実験では、ION GPU を用いた性能はハイエンドデスクトップ PC の性能を大きく上回ったが、自作の CPU プログラム部分が cvKmeans2 関数よりも速度の低下が見られたため、CUDA 利用でのスピードアップが最大約 21 倍となった。しかし、プログラムを改善し、cvKmeans2 関数により近い値またはそれを超える値を出すことができれば、さらなる速度向上が可能だと考えられる。

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>