画像処理フィルタ作りで 処理速度数十倍を体験

高木 瞭

GPUをうまく使えば数十倍の高速化もできる

本稿は並列プログラミングの初学者を対象に、GPUを用いた高速化の基礎を習得してもらうことを目的としています。

並列プログラミングと言っても、対象となるハードウェアによってさまざまな種類のライブラリや開発環境が存在します。

例えば、OpenMPやIntel TBB (Threading Building Blocks) といったスレッド並列ライブラリ、Open MPIやMPICHといったプロセス並列ライブラリ、AVX-512 (Advanced Vector Extensions 512) やAltiVec、NEONなどのSIMD (Single Instruction Multiple Data) 命令、CUDA (Compute Unified Device Architecture) やROCm (Radeon Open Compute) といったGPU向け開発環境が挙げられます。

本稿はNVIDIAの開発するCUDA⁽¹⁾を対象としています。NVIDIAのGPUは、組み込みボードからサーバ・コンピュータに至るまでさまざまなコンピュータに搭載されています。演算器の数やメモリの種類に多少差異はあるものの、世代ごとに共通のアーキテクチャを採用しています。

CUDAプログラミングを習得することで、車載カメラの画像処理から大規模な科学技術計算に至るまで、多様な計算を高速化できます。GPUに適した処理であれば、CPUでの計算と比べ数倍から数十倍もの高速化率を達成することも可能です^{注1}。

本稿では、読者が記事と同等の環境を用意しやすいよう、比較的入手性の高いパーツで構成されたマシンを使用します。具体的には、WindowsのデスクトップPCにゲーム用GPUカードを挿したマシンで開発を行います。

● 2種類の画像処理フィルタの実装でCUDAプログラミングを解説

高速化対象として、ガウシアン・フィルタとメディアン・フィルタの2種類の画像処理を取り上げます。

これらの画像処理は画素ごとに独立した処理であり、並列化が容易であることからGPUを使った計算に適しています。

ガウシアン・フィルタでは画像処理の基本的な高速化手法を扱います。メディアン・フィルタではアルゴリズムの工夫を含めた発展的な高速化手法を解説します。最後に、エラー処理や簡易的なデバッグ方法についても扱います。

世の中に優れたCUDAの書籍⁽²⁾⁽³⁾や解説記事はたくさんありますが、ソフトウェアやハードウェアの進歩が早いため、最新の環境に対応したものは少ないです。本稿では、ここ数年で新しく導入された機能も交えて解説します。

環境構築

CUDAプログラミングを始める前に、環境構築を行います。開発環境を表1に示します。

● ハードウェア

使用したPCは、CPUとしてCore i7-8700、GPUとしてGeForce GTX 1050 Ti (NVIDIA) を搭載しています。

GTX 1050 Tiは、2016年に発売されたミドルレンジGPUですが、安価で省スペース、省電力なため扱いやすい製品です。本記事はGTX 10シリーズ (Pascalアーキテクチャ) 以降のGPUを対象としています^{注2}。

注1: AMD EPYC 7763 (CPU) は、1CPU、1ソケット当たりの理論演算性能 (倍精度) が2508.8GFLOPS、理論メモリ帯域幅が204.8Gバイト/sです。対してNVIDIA A100 (GPU) は9.7TFLOPS、2039Gバイト/sと4~10倍程度の差があります。

注2: 2021年8月現在発売されているゲーム用GPUでは、GTX 16シリーズ (Turingアーキテクチャ)、RTX 20シリーズ (Turingアーキテクチャ)、RTX 30シリーズ (Ampereアーキテクチャ) のGPUも対象となります。GTX 9シリーズ (Maxwellアーキテクチャ) 以前では、記事中で紹介するCUDAの機能やソフトウェアの一部が対応していません。