GPU実装によるJones matrix OCT用 maximum a-posteriori 複屈折推定器

久保田敦[1)]，巻田修一[2)], 安野嘉晃[2)]

（株）スカイテクノロジー[1)]，筑波大学 Computational Optics Group[2)]

4pP16

はじめに

Jones-matrix OCT などの偏光 OCTを用いて局所位相遅延量を計測することで試料の複屈折分布を知ることができることが知られている。しかし、一般的な生体試料の局所位相遅延量は小さく、そのため、複屈折計測の信号雑音(SNR) は低い。

この問題を解決するために事後分布最大化 (Maximum *a-posteiori*; MAP) 複屈折推定器[1)]が提案されている。この方法は複数のSNRの低い局所位相遅延の計測量から試料の複屈折値を最尤推定する手法である。この推定器は数学的に厳密な方法であるが長い計算時間を必要する。従来のCPUによる計算を用いた実装では三次元分布の複屈折データの推定に40分程の処理時間が必要であった。

本研究ではGPUを用いたMAP複屈折推定器の実装を行い、複屈折推定の速度改善を行った。さらに、複数の GPUを用いて速度比較を行うことで、推定速度の制限要因を検討した。

手法

MAP複屈折推定器

MAP複屈折推定器は「ある有効SNR (ESNR) 値と複屈折値が計測された場合の真の複屈折の確率分布」（尤度関数）を利用して推定を行う。

図1に計算機によるMAP推定器の処理の流れを示す。ここでは計測に先立ち尤度関数をモンテカルロ法によってすべてのESNRと複屈折値の組み合わせに対して計算し、三次元の配列データとしてメインメモリ内に保存している。計測の際には試料の同一箇所を複数回計測する。その後の複屈折推定過程では各計測値に対応する尤度関数を三次元配列からそれぞれ参照し、それらをすべて掛け合わせることで真の複屈折の事後分布を求める。最終的に、その事後分布が最大値をとるような複屈折値をMAP推定値とする。

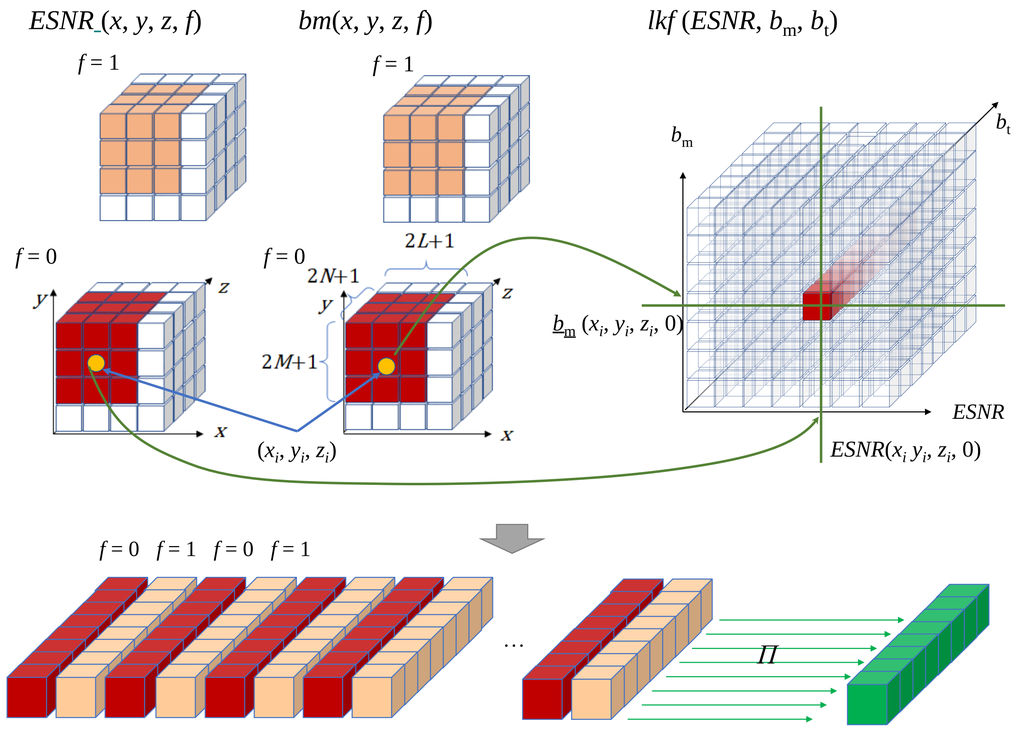

Fig. 1 Schematic diagram of data and their relationship in MAP birefringence estimate.

GPUアーキテクチャ

今回のGPU実装ではPascalアーキテクチャのGPUを使用した。Pascal アーキテクチャでは1つのstreaming multiprocessor (SM) 当たり、4基のwarpユニット、1 warp当たり32個のCUDA core、合計128個のCUDA coreが利用可能である※。また、1 つのSM内のすべてのCUDA coreは96 kBのshared memoryを共有する。

GPU演算の高速化のためには、アクセス速度は速いが小容量であるSMを効率よく使用し、より多くのthread (= CUDA core)を同時実行することが重要である。

※）例えば、Pascal アーキテクチャのGPUの１つであるGTX1080Ti (Nvidia, CA, US) には28個のSMがあり、3,584 CUDA coreが利用可能である。

Fig. 2 Architecture of streaming processor.

GPU実装

JM-OCT画像の１pixel（１点）のMAP推定の処理過程ではfloat型1,024要素の一次元配列をメモリに保存し、それに対し複数回の読み書きを行う必要がある。この配列を保存するには1,024 × 4 byte = 4 kBのメモリが必要である。今回の実装ではこの１pixelの推定処理を1つのthread (CUDA core)で処理を行い、複数のpixelの複屈折推定を並列に行う。今回のGPUでは1 SMあたり96 kBの shared memory が利用可能であることから、1 SMあたり24 pixelのMAP推定を並列処理可能である。

この並列数は1 SM当たりのCUDA core数 (128個) よりも少ない為、並列処理されるpixel数はCUDA core数ではなく共有メモリのサイズによって制限されている。

一般的なCPUでは同時実行可能な thread数は8～16個であることから、GPUでは約50倍の処理を同時実行できることになる。

結果

各点を4回ずつ繰り返し計測した三次元JM-OCT データ（512 × 640 × 256 pixel）を用いて複屈折の MAP 推定を行った。表1に複数のGPUモデル、および従来のCPU実装の推定速度の比較を示す。これによりGPUを用いることでCPUの8倍の推定速度が得られることがわかかった。

Processor	Processing time/slice[sec]	Total processing time (256 slices) [sec]	Total # of parallelly processable thread	Total # of CUDA cores
GTX 1060	0.77	195.98	240	1280
GTX 1070M	0.47	119.44	384	2048
GTX 1080Ti	0.31	78368	672	3584
CPU※	6.30	1,637	(unknown)	---

※ Core i7-7700HQ 2.8GHz Processed with python 2.7 with Numpy1.16.4 (OpenBlas)

Table 1 The processing performance of MAP birefringence estimates and GPU specs.

GPUモデルにより並列処理可能なthread数が異なる。図３に並列処理可能なthread数とOCT slice (512 × 640 pixel)の推定速度の関係を示す。並列処理可能な thread数に比例して推定速度が向上していることがわかる。

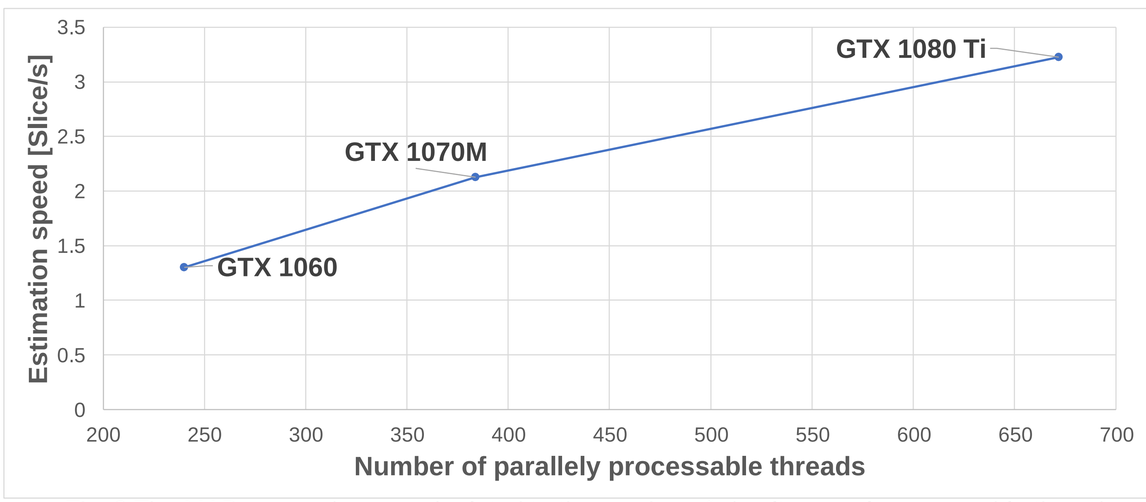

Fig. 3 The MAP processing speed related to the number of simultaneously processable threads.

結論

MAP複屈折推定処理をGPUコードで実装することにより、処理時間の大幅な短縮が可能であることがわかった。GPUモデル間の比較により処理速度は並列実行可能なthread数に対し線形に向上することがわかった。MAP複屈折推定処理の処理時間はGPU SMのshared memoryサイズにより制限されている。そのため、次の二つの方法でさらなる速度向上が見込まれる。(1) GPUアーキテクチャ改良によるSM当たりのshared memoryサイズが増加。(2) 演算アルゴリズムの改良による1ピクセル処理あたりのthread数の増加。

今回のGPU実装によりMAP複屈折推定処理がJM-OCTの画像再構成処理に占める割合が大幅に減少した。今後、MAP複屈折推定以外の処理の最適化を行うことで、さらなるJM-OCTの利便性向上が期待される。

参考文献

1) D. Kasaragod, S. Makita, Y.-J. Hong, and Y. Yasuno, *Biomed. Opt. Express* **8**, 653–669 (2017).