High-Speed Maximum A Posteriori Birefringence Estimator for Jones Matrix Optical Coherence Tomography by GPU Implementation

Atsushi Kubota [1], Shuichi Makita [2], Yoshiaki Yasuno [2]

Sky technology Inc., Tsukuba, Ibaraki, Japan. [1], Computational Optics Group, University of Tsukuba, Tsukuba, Ibaraki, Japan. [2]

11228-87

Background and Purpose

It is known that the birefringence distribution of a sample can be known by measuring the amount of local phase retardation using polarization OCT such as Jones-matrix OCT. However, the amount of local phase retardation of a general biological sample is small, and therefore, it generally has a low signal-to-noise ratio (SNR).

Maximum *a posterior* (MAP) birefringence estimation [1] has been proposed to solve this problem. The maximum likelihood estimate of birefringence has been obtained from multiple measured local phase retardation values with a low SNR. This estimator is mathematically exact, however, it requires a long calculation time. The previous implementation of MAP birefringence estimation is based on CPU processing, and it took 40 min for the birefringence estimation of an OCT volume.

In this research, we implemented a MAP birefringence estimator using GPU and improved the speed of birefringence estimation. Furthermore, the speed limiting factor was examined by comparing the speeds using multiple GPUs.

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

Method

MAP birefringence estimator

The MAP birefringence estimator performs estimation using a "probability distribution of true birefringence when a certain effective SNR (*ESNR*) value and a birefringence (b_m) value are measured" (likelihood function).

Fig.1 shows a processing flow of the MAP estimator. Here, prior to the measurement, a likelihood function is calculated for all combinations of ESNR and birefringence values by the Monte-Carlo simulation and stored in the main memory as a three-dimensional array $L(ESNR, B_m, b_t)$.

Multiple (*f*-times) measurements at the same location of the sample is performed.

In the subsequent birefringence estimation process, the likelihood function corresponding to each measured value is referred to from the three-dimensional array L, and all of them are multiplied to obtain a true posterior distribution of birefringence.

Finally, the birefringence value at which the posterior distribution takes the maximum value is used as the MAP estimation value.

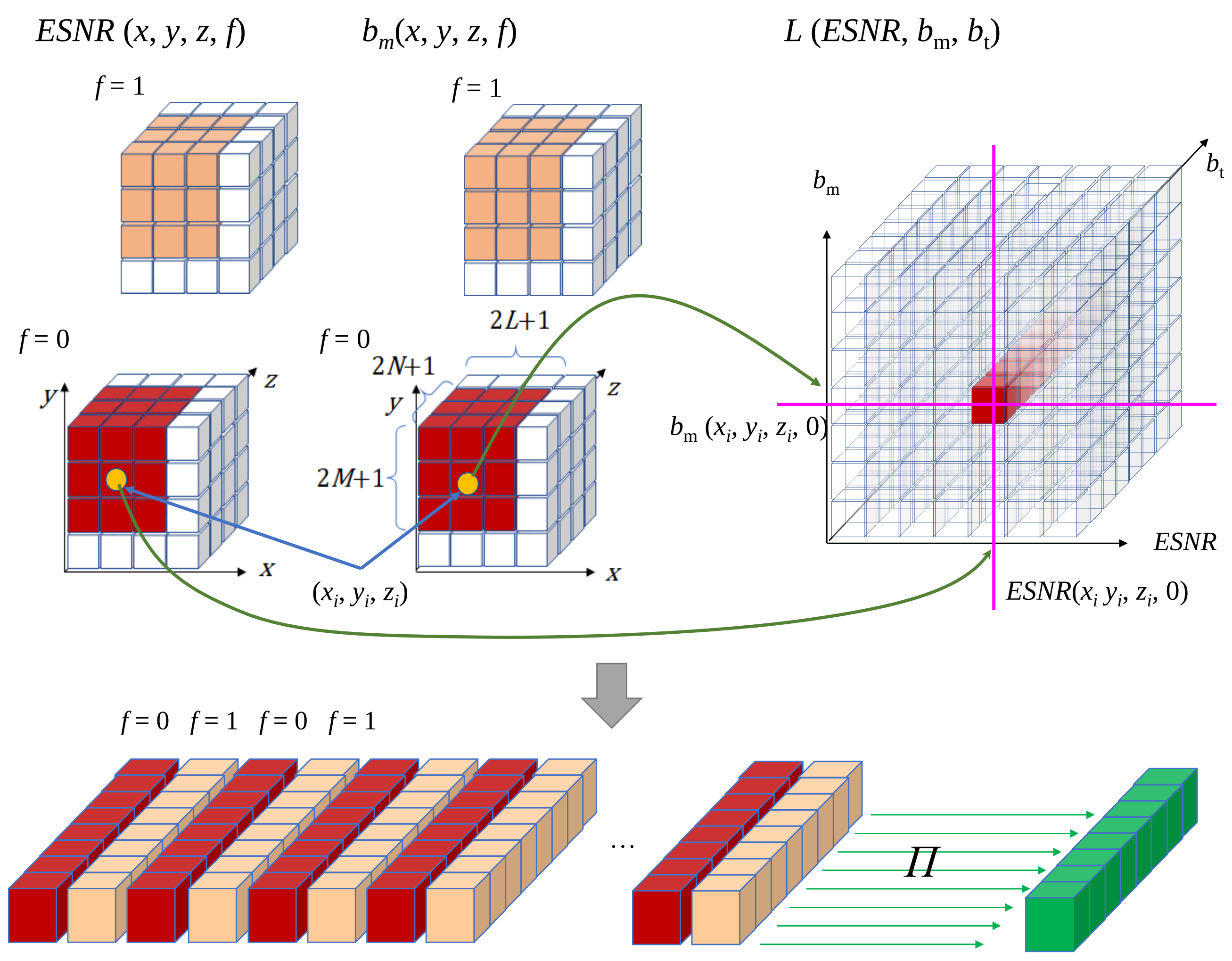
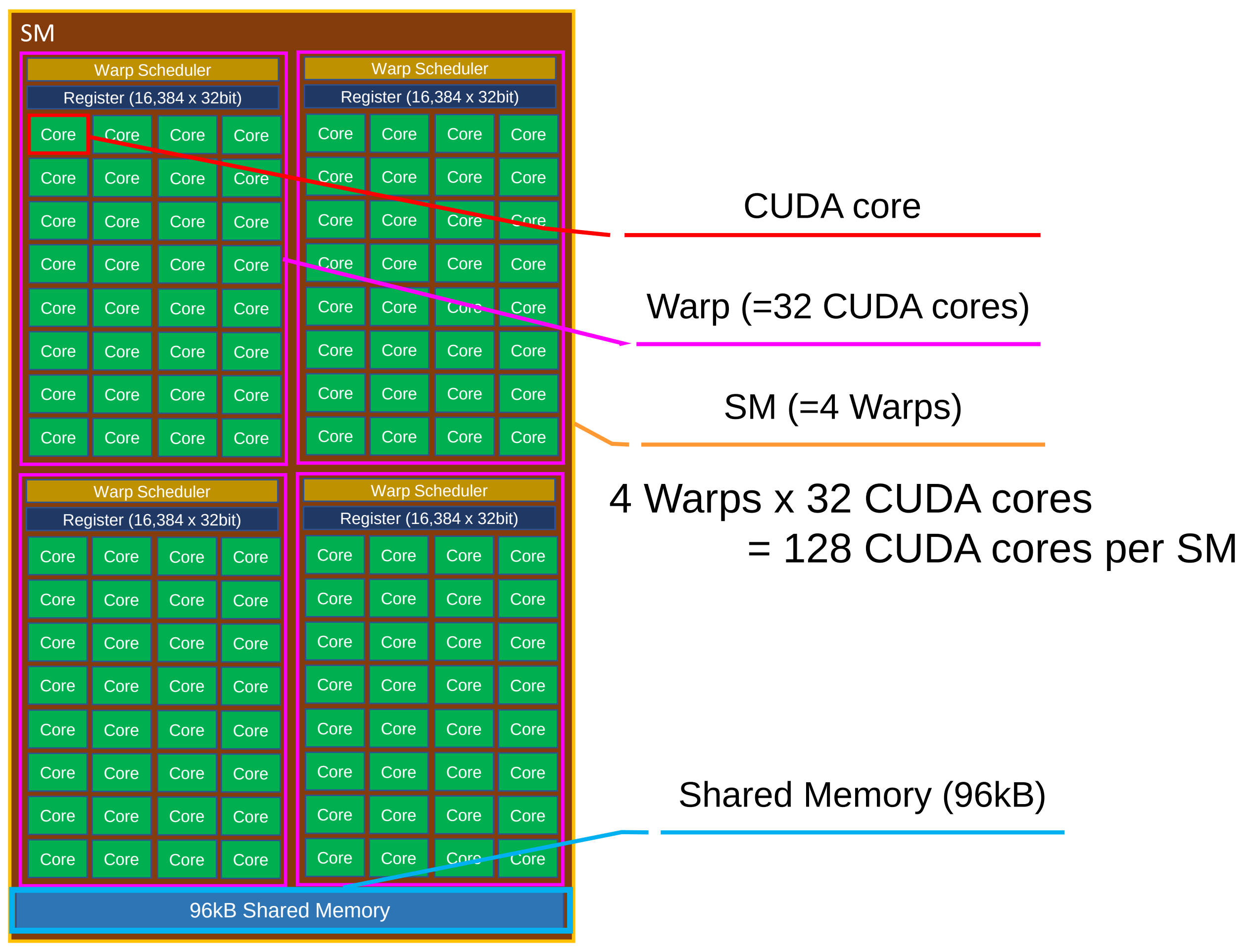


Fig. 1 Schematic diagram of data and their relationship in MAP birefringence estimate.

GPU architecture

In this implementation, we used GPU of NVIDIA pascal architecture. In the Pascal architecture, 4 Warp units per Streaming Multiprocessor (SM), 32 CUDA cores per Warp, and a total of 128 CUDA cores are available. For example, GTX1080Ti (Nvidia, CA, US), one of the Pascal architecture GPUs, has 28 SMs and 3,584 CUDA cores can be used. All CUDA cores in one SM share 96 kB of Shared Memory.

In order to speed up the GPU operation, it is important to efficiently use the Shared Memory and to simultaneously execute more threads (= CUDA core). It is because the Shared Memory has a high access speed but a small capacity.



4 Warps x 32 CUDA cores = 128 CUDA cores per SM

Fig. 2 Streaming Multiprocessor(SM) configuration of NVIDIA pascal architecture.

GPU implementation

In the process of estimating the MAP of 1 pixel (1 point) in a JM-OCT image, it is necessary to store a one-dimensional likelihood array data of float type 1,024 elements in the same memory and to read and write multiple times. To store this array, 1,024 × 4 bytes = 4 kB of memory is required.

In this implementation, this 1-pixel estimation process is performed by 1-thread (CUDA core), and multiple estimation processes for multiple pixels are parallelly performed.

The number of parallelly performed estimation is limited by the shared memory size. This GPU has 96 kB shared memory per SM. And hence, a SM parallelly performs 96 kB / 4 kB = 24 estimations for 24 pixels in parallel [Fig 3]. This parallel number is less than the number of CUDA cores (128) per SM, so parallelism is not limited by the number of CUDA cores, but by the size of the shared memory.

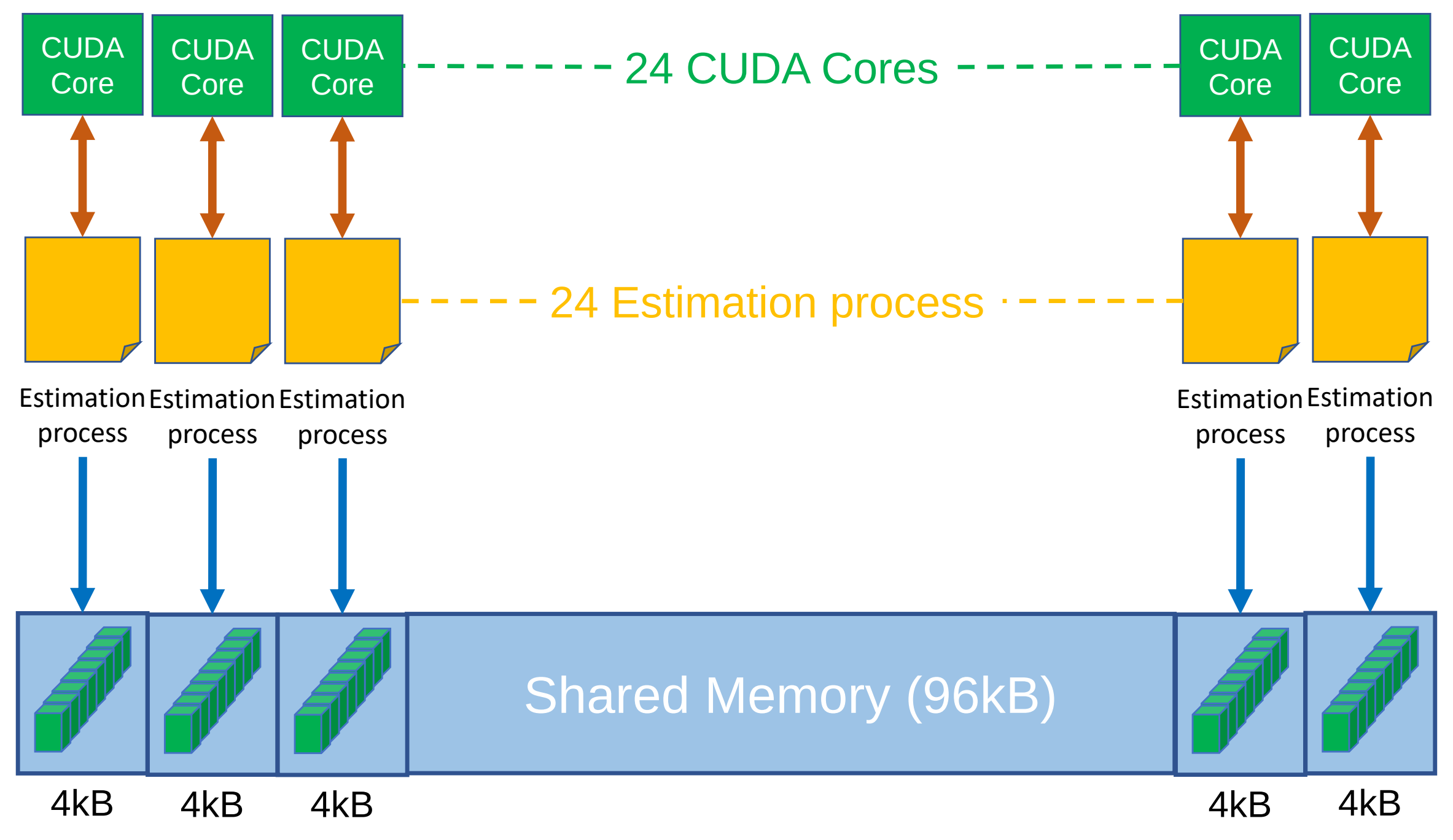


Fig. 3 MAP estimation process per SM of NVIDIA pascal architecture.

Result

MAP estimation of birefringence was performed using three-dimensional JM-OCT data (512 × 640 × 256 pixels) in which each point was repeatedly measured four times. Table 1 compares the estimation speeds of multiple GPU models and conventional CPU implementations. As a result, it was found that the use of the GPU can provide 8-times faster estimation speed than the CPU.

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.

The number of threads that can be processed in parallel differs among the GPU models. Fig. 4 shows the relationship between the number of threads that can be processed in parallel and the estimated speed of the OCT slice (512 × 640 pixels). It is found that the estimation speed increases in proportion to the number of threads that can be processed in parallel.

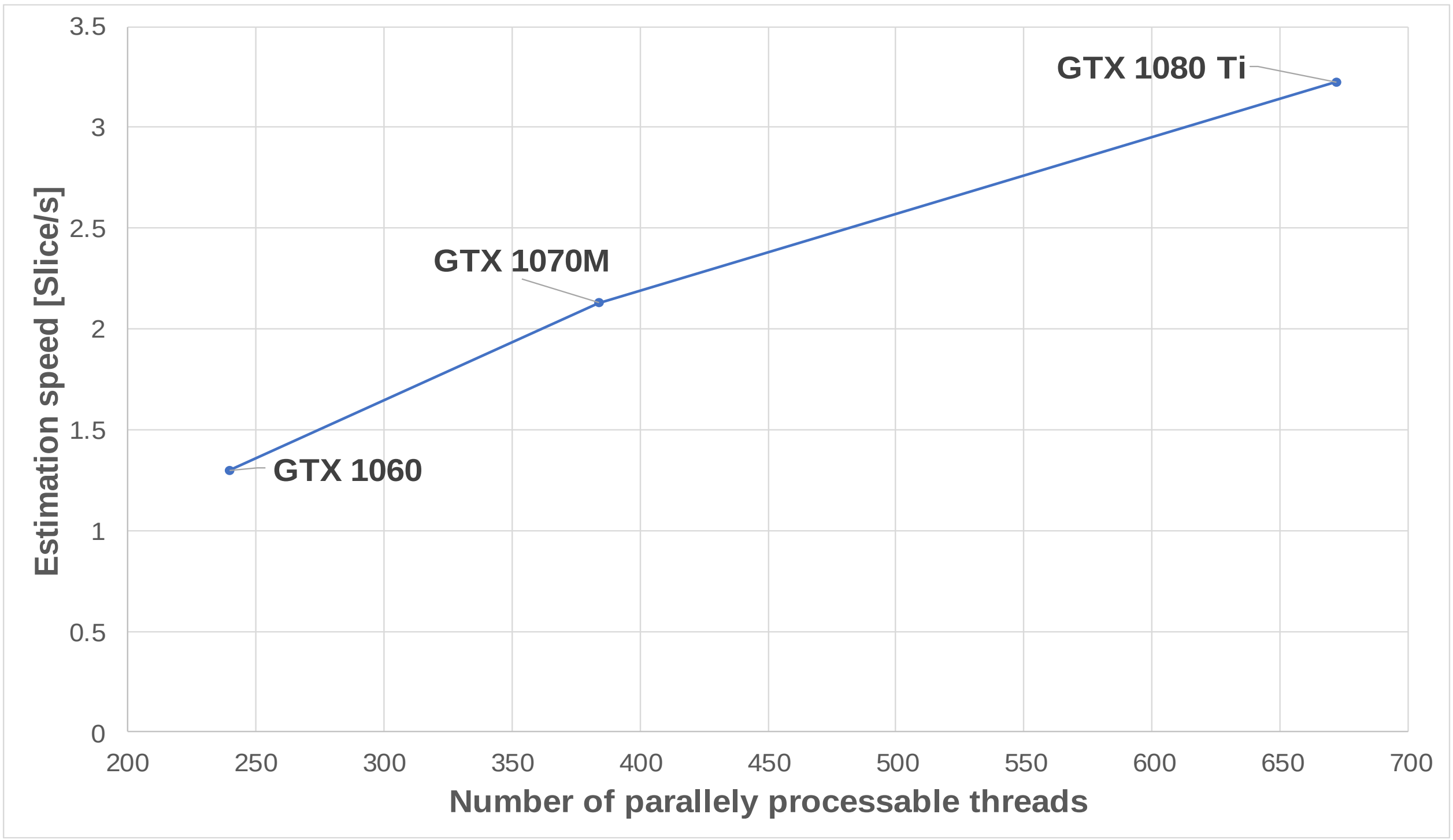


Fig. 4: The MAP processing speed related to the number of simultaneously processable threads.

Conclusion

The MAP birefringence estimation processing using GPU was implemented. Significantly faster processing speed was achieved than a CPU implementation. The comparison among GPU models showed that the processing speed increased linearly with the number of threads that could be executed in parallel.

It was found that the processing time of the MAP birefringence estimation processing is limited by the shared memory size of the GPU SM. Therefore, the following two methods are expected to further improve the speed.

(1) Shared memory size per SM increased due to improved GPU architecture.

(2) Increase in the number of threads per pixel processing by improving the operation algorithm.

Owing to this GPU-based MAP birefringence estimation, the birefringence computation time became a non-significant portion of full JM-OCT image reconstruction, which includes OCT reconstruction, phase stabilization, OCT angiography computation, and degree-of-polarization uniformity computation. In the future, faster computation of JM-OCT is expected by GPU implementation of other processing than MAP birefringence estimation.