Journal of Biomedical Optics

SPIEDigitalLibrary.org/jbo

Graphics processing unit accelerated optical coherence tomography processing at megahertz axial scan rate and high resolution video rate volumetric rendering

Yifan Jian
Kevin Wong
Marinko V. Sarunic

SPIE

Graphics processing unit accelerated optical coherence tomography processing at megahertz axial scan rate and high resolution video rate volumetric rendering

Yifan Jian, Kevin Wong, and Marinko V. Sarunic
Simon Fraser University, School of Engineering Science, Burnaby, BC V5A 1S6, Canada

Abstract. In this report, we describe how to highly optimize a computer unified device architecture based platform to perform real-time processing of optical coherence tomography interferometric data and three-dimensional (3-D) volumetric rendering using a commercially available, cost-effective, graphics processing unit (GPU). The maximum complete attainable axial scan processing rate, including memory transfer and displaying B-scan frame, was 2.24 MHz for 16 bits pixel depth and 2048 fast Fourier transform size; the maximum 3-D volumetric rendering rate, including B-scan, *en face* view display, and 3-D rendering, was ~23 volumes/second (volume size: $1024 \times 256 \times 200$). To the best of our knowledge, this is the fastest processing rate reported to date with a single-chip GPU and the first implementation of real-time video-rate volumetric optical coherence tomography (OCT) processing and rendering that is capable of matching the acquisition rates of ultrahigh-speed OCT. [DOI: 10.1117/1.JBO.18.2.026002]

Keywords: image processing; imaging coherence; medical imaging; parallel processing.

Paper 12663 received Oct. 11, 2012; revised manuscript received Dec. 7, 2012; accepted for publication Jan. 7, 2013; published online Feb. 1, 2013.

1 Introduction

Optical coherence tomography (OCT) is an essential diagnostic tool in ophthalmic clinics and is expanding its range of applications rapidly due to its ability to acquire high resolution cross-sectional images noninvasively.[1] The acquisition speed of OCT has increased tremendously since it was first demonstrated in 1991, going from 400 Hz to 20 MHz line rates for tissue imaging.[1] Commercially available swept sources are able to provide a 400 kHz axial scan rate with relatively minor modifications, such as dual channels and double buffers.[2] Multimegahertz Fourier domain mode locking -based swept source (SS) OCT systems capable of acquiring high-resolution volumes at video-rate have been demonstrated,[3] and 1.6 MHz systems have been presented that are used for clinical retinal imaging.[4] Spectral domain (SD) OCT systems operating in the 800 nm wavelength range have been presented operating with an axial scan rate of 500 kHz with a dual camera configuration.[5]

As the ultrahigh-speed OCT acquisition has continuously been extended, there followed an increasing demand for real-time volumetric visualization of OCT data to explore the full potential of the technology, such as intraoperative OCT[6] and functional OCT.[7,8] However, due to the complexity of OCT data processing and extremely high data throughput, processing interferometric fringe data into images requires significant computational resources, and to date has been far slower than the acquisition rate. Although ultrahigh-speed OCT is capable of acquiring volumetric data in real-time, nearly all of the OCT systems render the three-dimensional (3-D) images in post-processing which greatly limits the range of applications. Several attempts have been made to accelerate the OCT data processing and volume rendering utilizing graphics processing units (GPUs) and field-programmable gate array, including Refs. 8–15, just to name a few. A GPU-based approach for volume rendering was presented at a reduced volume size at 5 frames per second (fps),[13] but high resolution, video rate, real-time volumetric rendering has not yet been realized.

In this report, we discuss strategies to hide the latency of memory transfer, and describe a custom computer unified device architecture (CUDA) program for real time OCT data processing and volume rendering. We present data processing and display of high resolution volumes at video rate.

2 Methods

To exploit the massive parallel computational power of the GPU, we used NVIDIA's (Santa Clara, California) parallel programming platform, CUDA version 4.2 which offers easy integration and implementation of general purpose computation with GPUs, and OpenGL[16] as our display library. CUDA Visual Profiler was used in our project to record the timing and calculate the processing speed. Microsoft Visual Studio 2008 was used to build and compile the project. We tested our software on three generations of NVIDIA's low cost consumer grade GPUs (GTX 460 1 GB RAM, GTX 560 1 GB RAM, and GTX 680 2 GB RAM) to investigate their performance and scalability. Each GPU was hosted in a desktop computer with Intel Core i7 CPU running Windows 7 operating system. The only upgrade to the computer was to use a workstation level motherboard that provides sufficient PCI Express (PCIe) bandwidth for throughput of the data between the acquisition boards and the GPU.

The OCT images presented in this report were acquired by two custom OCT systems. The SS-OCT system used an AlazarTech (Pointe-Claire, QC, Canada) digitizer and a 100 kHz

Address all correspondence to: Marinko V. Sarunic, Simon Fraser University, School of Engineering Science, Burnaby, BC V5A 1S6, Canada. Tel: (778) 782 7654; Fax: (778) 782 4951; E-mail: msarunic@sfu.ca

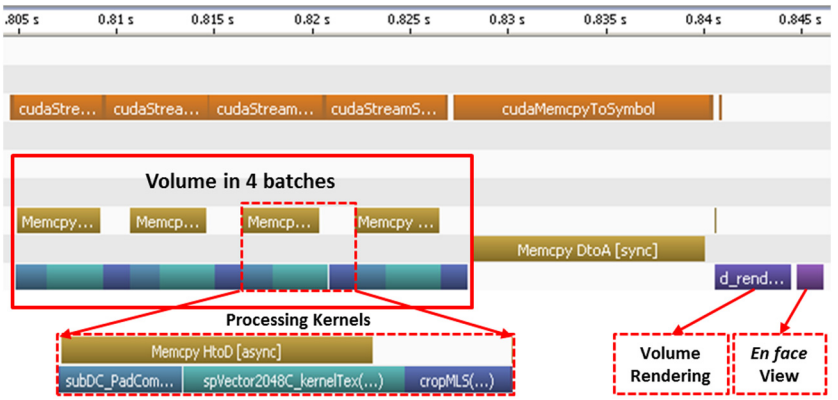

Fig. 1 CUDA Profiler Timeline for SS-OCT processing and volume rendering pipeline. Memcpy HtoD [async] is transferring the raw data, subDC_PadCom is a kernel with the function of type conversion, DC subtraction, and zero padding. spVector2048C_kernelTex is a kernel called by the cuFFT library which performs the fast Fourier transform (FFT). CropMLS performs the modulus and Log. Memcpy DtoA [sync] copies the processed OCT data into a CUDA 3D array for volume rendering.

Axsun (Billerica, MA) wavelength swept laser. This system was used for real-time acquisition and display of human retinal data. The SD-OCT system utilized a high-speed CMOS line scan camera detector (Sprint spL4096-140k, Basler AG, Germany) and superluminescent diode source centered at 860 nm with a 135 nm FWHM bandwidth (Superlum Inc., Moscow). The maximum line rate attainable to support this bandwidth was 125 kHz. Since we do not have a megahertz line rate OCT system, to evaluate the processing performance we loaded interferometric data from a file into an intermediate buffer in the host RAM, and then transferred the whole volume to another buffer also in the host RAM to represent the acquisition of real-time data from a frame grabber or analog-to-digital converter (ADC). We verified experimentally that the memory transfer speed from host to host in our computer is about 5.1 GB/s, thereafter we were able to simulate a specific OCT acquisition speed by adding delays during to the memory transfer. For example, by adding a 5 ms delay to a volume ($1024 \times 256 \times 200$ pixel) transfer, we were able to simulate a 2 MHz OCT acquisition.

The limiting factor in the previously reported GPU implementation of OCT data processing was the overhead in memory transfer between the CPU host RAM and GPU device off-chip global memory. Facing the constraint of the limited PCIe bus bandwidth, our strategy was to parallelize the data transfer from host to device with processing kernels to hide memory transfer latency, and eliminate the need of transferring the data back to the CPU side (host) by rendering the processed volume and B-scans directly on the GPU. This method itself gives a 3× speed boost compared to our previous work[9] since the processing kernel calls take almost the same time as the memory transfer on the PCIe bus. The concurrent memory transfer from host to device and data processing on GPU was designed using two CUDA streams that are executed concurrently and synchronized after each processing iteration: one to transfer the data from host to device, and one to process the raw OCT data on the GPU memory. The unprocessed raw data was transferred to the GPU as a batch of frames instead of as a single frame to fully utilize the PCIe bandwidth and the copy engine of the GPU. While one batch of raw data was being transferred to the GPU by the transfer stream, the previous batch of frames that was already in the GPU memory was simultaneously being processed by the kernel stream. Note that in Fig. 1, the memory copy function that was used to transfer data from host to device is asynchronous; it will return immediately without waiting for the completion of the memory transfer and will not block the execution of the processing kernels. In order to fulfill the restriction of concurrent memory transfer and kernel execution, the raw data was stored in the page-locked memory in the host. In addition, transferring data from the page-locked host memory to the device allows the utilization of the full PCIe bandwidth (6.2 GB/s in GTX 680, and 5.7 GB/s in GTX 460 and 560, compared to ~3 GB/s in nonpage-locked host memory).[10] To further reduce the memory transfer overhead, we transfer the interferometric fringe data to the GPU as 16-bit integers, and then cast the data to floating point format (float) on the GPU in a parallel manner.

Figure 2 shows a flow chart of the OCT data processing pipeline used in our method. The full OCT processing pipeline was optimized for SD-OCT, but some processing blocks were dropped from the SS-OCT pipeline in exchange for an increase in processing speed. For example, since in SS-OCT the interferometric data can be sampled linearly in wavenumber, the lambda to wavenumber resampling part of the pipeline can be dropped. Similarly, for 1310-nm and 1060-nm imaging systems, the dispersion can usually be compensated physically, whereas the processing blocks for numerical dispersion compensation are typically needed for SD-OCT systems in the 800 nm range.

3 Results and Discussion

A screen capture of the CUDA profiler timeline for SS-OCT processing at the maximum line rate is shown in Fig. 1 to demonstrate the detailed timing of our program. The timing for one complete SS-OCT volume processing and rendering iteration is

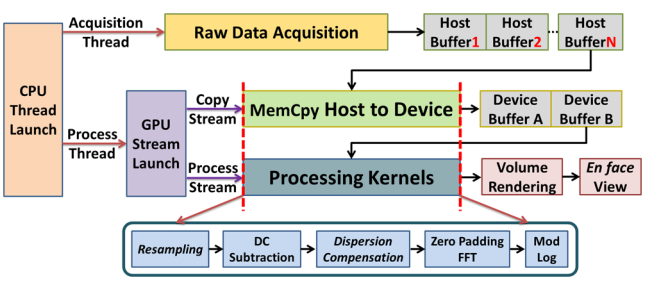

Fig. 2 Flow chart for OCT acquisition and GPU processing. The italicized procedure names in the Processing Kernels represent the blocks that were dropped for the SS OCT implementation.

shown in Fig. 1. Every acquired volume (256×200 A-scans) was divided into four batches for transfer and processing in the GPU. The profiler output in Fig. 1 indicates that the memory transfer and OCT processing kernel were overlapped. Once the whole OCT volume was processed, the processed data was transferred to another device buffer and assembled into a 3-D CUDA array in preparation for the volume rendering. As a final step, the volume rendering and *en face* view were executed. The complete pipeline required ~43 ms, corresponding to a volume processing and rendering rate of ~23 volumes/s.

As shown in Fig. 1, an *en face* view and a volume rendering were also performed on the GPU following the volume processing. To render the processed OCT B-scan, *en face* view, and 3-D volume directly from the GPU global memory as 32-bit floating point texture (which avoids type conversion and transferring data back to the host), the CUDA resource allocated for holding the processed data was registered to OpenGL using CUDA/OpenGL interoperability functions. The *en face* projection of the volume was generated by summing up all the pixels

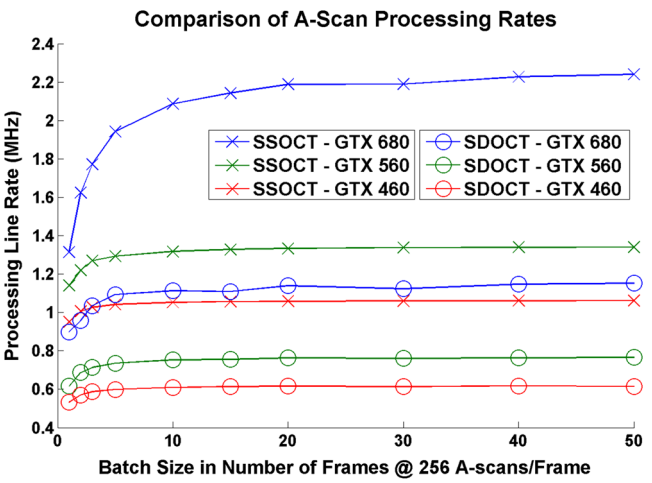

Fig. 3 OCT A-Scan batch processing rate. For SS-OCT, the processing pipeline includes DC subtraction, FFT, modulus and Log. For SD-OCT, the processing pipeline includes linear interpolation, DC subtraction, dispersion compensation, FFT, modulus and Log.

in each A-scan using an optimized parallel reduction algorithm.[17] A ray casting method was used to render the processed OCT volume.[18] Compared to other implementations reported in the literature, our method requires only one GPU to process and render the OCT volume which would lower the cost and more importantly reduce the data transfer across the PCIe bus.

In order to further optimize the processing and memory transfer speed of the GPU, the batch processing size was investigated. Figure 3 shows the plot of batch size versus axial scan line rate in our implementation, including time for memory transfer and displaying the B-scan frame. The processing rate plateaued around ~3840 A-scans per batch. Our GPU accelerated processing achieved an A-scan (16-bit, 2048 pixels) rate of >2.24 MHz, and volume ($1024 \times 256 \times 200$) rendering at >23 volumes per second. At the current PCIe data transfer rates from host to device (using PCIe 2.0×16), this upper limit will be at a line rate of ~3.1 MHz (16-bit, 1024 pixels/line). Figure 4(a) and Video 1 present screen captures of images processed and rendered using our GPU accelerated with SS-OCT for human retina. For this video, the raw data was loaded from a file with delays added to simulate an acquisition line rate of 1.2 MHz, as described above. To demonstrate the real-time acquisition and display capability, we performed an *in vivo* real-time SS-OCT imaging for human retina in the Eye Care Center at Vancouver General Hospital with a line rate of 100 kHz that was limited by our source, as seen in Fig. 4(b) and Video 2. The displayed B-scans were averaged (4 adjacent frames) and a bilateral filter was applied, with both steps implemented on the GPU.

For SD-OCT, the complete processing pipeline with wavenumber resampling and numerical dispersion compensation was implemented. Instead of developing a custom kernel to calculate the interpolation, we utilized GPU texture as a hard-wired linear interpolation method,[19] which also gives an extra benefit of implicitly casting integer data into floating point. Although linear interpolation offered the fastest processing speed, it produced lower quality images. We implemented a fast cubic spline interpolation that provided a balance between the image quality and processing speed.[20] With the additional computational load required for SD-OCT data, we demonstrated volume processing at an A-scan rate of 1.1 MHz using linear interpolation and

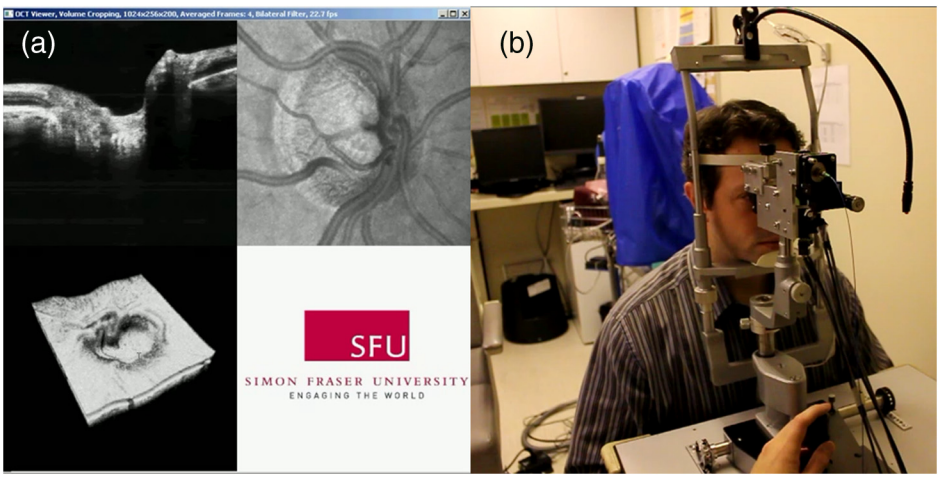

Fig. 4 (a) Screen captures of OCT images (Video 1). Upper left: OCT B-scan. Upper right: *En face* view. Lower left: 3D volumetric rendering. (Video 1, QuickTime, 9.69 MB) [URL: http://dx.doi.org/10.1117/1.JBO.18.2.026002.1]; (b) Real time acquisition and processing of human retinal images (Video 2, QuickTime, 3.33 MB) [URL: http://dx.doi.org/10.1117/1.JBO.18.2.026002.2].

1 MHz with fast cubic interpolation. This is well in excess of the fastest SD-OCT acquisition system reported,[5] leaving GPU resources available to implement more advanced image processing.

In conclusion, we have demonstrated sustained A-scan processing rates of 2.24 MHz for SS-OCT, and 1 MHz for SD-OCT using a commercial-grade GPU and desktop computer. Our program is able to process and render the volumetric OCT data at ~23 volumes/s (volume size $1024 \times 256 \times 200$). Real-time, video rate, volumetric visualization of OCT data has exciting applications in diagnostic and surgical applications. The GPU implementation is low cost, and can be easily integrated with existing acquisition systems. The source code for transferring interferometric data from the host to the GPU, and for processing to the point of display, is available.[21]

Acknowledgments

We acknowledge funding for this research from Michael Smith Foundation for Health Research (MSFHR), and Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. W. Drexler and J. G. Fujimoto, "State-of-the-art retinal optical coherence tomography," *Progr. Retinal Eye Res.* **27**(1), 45–88 (2008).
2. B. Potsaid et al., "Ultrahigh speed 1050 nm swept source/Fourier domain OCT retinal and anterior segment imaging at 100,000 to 400,000 axial scans per second," *Opt. Express* **18**(19), 20029–20048 (2010).
3. W. Wieser et al., "Multi-Megahertz OCT: high quality 3D imaging at 20 million A-scans and 45 GVoxels per second," *Opt. Express* **18**(14), 14685–14704 (2010).
4. C. Blatter et al., "Ultrahigh-speed non-invasive widefield angiography," *J. Biomed. Opt.* **17**(7), 070505 (2012).
5. L. An et al., "High speed spectral domain optical coherence tomography for retinal imaging at 500,000 A-lines per second," *Biomed. Opt. Express* **2**(10), 2770–2783 (2011).
6. Y. K. Tao et al., "Intraoperative spectral domain optical coherence tomography for vitreoretinal surgery," *Opt. Lett.* **35**(20), 3315–3317 (2010).
7. M. Sylwestrzak et al., "Four-dimensional structural and Doppler optical coherence tomography imaging on graphics processing units," *J. Biomed. Opt.* **17**(10), 100502 (2012).
8. K. K. C. Lee et al., "Real-time speckle variance swept-source optical coherence tomography using a graphics processing unit," *Biomed. Opt. Express* **3**(7), 1557–1564 (2012).
9. J. Li and P. Bloch et al., "Performance and scalability of Fourier domain optical coherence tomography acceleration using graphics processing units," *Appl. Opt.* **50**(13), 1832–1838 (2011).
10. J. Rasakanthan, K. Sugden, and P. H. Tomlins, "Processing and rendering of Fourier domain optical coherence tomography images at a line rate over 524 kHz using a graphics processing unit," *J. Biomed. Opt.* **16**(2), 020505 (2011).
11. M. Sylwestrzak et al., "Real-time massively parallel processing of spectral optical coherence tomography data on graphics processing units," *Proc. SPIE* **8091**, 80910V (2011).
12. Y. Watanabe and T. Itagaki, "Real-time display on Fourier domain optical coherence tomography system using a graphics processing unit," *J. Biomed. Opt.* **14**(6), 060506 (2009).
13. K. Zhang and J. U. Kang, "Graphics processing unit-based ultrahigh speed real-time Fourier domain optical coherence tomography," *IEEE J. Sel. Topics Quantum Electron.* **18**(4), 1270–1279 (2012).
14. A. E. Desjardins et al., "Real-time FPGA processing for high-speed optical frequency domain imaging," *IEEE Trans. Med. Imag.* **28**(9), 1468–1472 (2009).
15. T. E. Ustun et al., "Real-time processing for Fourier domain optical coherence tomography using a field programmable gate array," *Rev. Sci. Instrum.* **79**(11), 114301 (2008).
16. OpenGL et al., *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL(R)*, Version 2, 5th edn., Addison-Wesley Professional, Boston (2005).
17. M. Harris, S. Sengupta, and J. Owens, "Parallel Prefix Sum (Scan) with CUDA," in *GPU Gems 3*, H. Nguyen, Ed., pp. 851–876, Addison-Wesley, Boston (2007).
18. J. Probst et al., "Optical coherence tomography with online visualization of more than seven rendered volumes per second," *J. Biomed. Opt.* **15**(2), 026014 (2010).
19. S. Van Der Jeught, A. Bradu, and A. G. Podoleanu, "Real-time resampling in Fourier domain optical coherence tomography using a graphics processing unit," *J. Biomed. Opt.* **15**(3), 030511 (2010).
20. D. Ruijters and P. Thevenaz, "GPU prefilter for accurate cubic B-spline interpolation," *Comput. J.* **55**(1), 15–20 (2012).
21. Y. Jian, K. Wong, and M. Sarunic, "GPU Open Source Code," http://borg.ensc.sfu.ca/research/fdoct-gpu-code.html (20 January 2013).