

Journal of Biomedical Optics

SPIEDigitalLibrary.org/jbo

Four-dimensional structural and Doppler optical coherence tomography imaging on graphics processing units

Marcin Sylwestrzak
Daniel Szlag
Maciej Szkulmowski
Iwona Gorczynska
Danuta Bukowska
Maciej Wojtkowski
Piotr Targowski

Four-dimensional structural and Doppler optical coherence tomography imaging on graphics processing units

Marcin Sylwestrzak, Daniel Szlag, Maciej Szkulmowski, Iwona Gorczynska, Danuta Bukowska, Maciej Wojtkowski, and Piotr Targowski

Nicolaus Copernicus University, Institute of Physics, Grudziadzka 5, PL-87-100 Torun, Poland

Abstract. The authors present the application of graphics processing unit (GPU) programming for real-time three-dimensional (3-D) Fourier domain optical coherence tomography (FDOCT) imaging with implementation of flow visualization algorithms. One of the limitations of FDOCT is data processing time, which is generally longer than data acquisition time. Utilizing additional algorithms, such as Doppler analysis, further increases computation time. The general purpose computing on GPU (GPGPU) has been used successfully for structural OCT imaging, but real-time 3-D imaging of flows has so far not been presented. We have developed software for structural and Doppler OCT processing capable of visualization of two-dimensional (2-D) data (2000 A-scans, 2048 pixels per spectrum) with an image refresh rate higher than 120 Hz. The 3-D imaging of 100×100 A-scans data is performed at a rate of about 9 volumes per second. We describe the software architecture, organization of threads, and optimization. Screen shots recorded during real-time imaging of a flow phantom and the human eye are presented. © 2012

Society of Photo-Optical Instrumentation Engineers (SPIE). [DOI: [10.1117/1.JBO.17.10.100502](https://doi.org/10.1117/1.JBO.17.10.100502)]

Keywords: spectral optical coherence tomography; real-time imaging; Doppler optical coherence tomography; graphics processing unit programming; general purpose computing on graphics processing unit; compute unified device architecture.

Paper 12467L received Jul. 20, 2012; revised manuscript received Sep. 12, 2012; accepted for publication Sep. 13, 2012; published online Oct. 5, 2012.

Optical coherence tomography (OCT) is a method of noncontact and noninvasive imaging of the internal structure of objects that are semi-transparent in infrared light. In 2002, Fourier domain optical coherence tomography (FDOCT) was successfully used for *in vivo* examination of the human eye¹ for the first time. Specialized modalities of the FDOCT technique have been developed to enable functional imaging of biological tissue. One of them is Doppler OCT, which enables blood-flow imaging.²

Generally, the quality of *in vivo* imaging depends strongly on the ergonomics of the OCT system, especially a fast, real-time preview that helps to adjust the instrument and get the best possible results. In the case of Doppler measurements, the problem is a little bit more complicated because the flow velocity range depends on the scanning protocol's parameters.³ Therefore, we believe that the introduction of the real-time preview of the Doppler signal is necessary to obtain a reliable quantitative signature of the retinal blood flow.

In the case of biomedical imaging, parallel processing on GPUs has been used in computed tomography,⁴ ultrasonography,⁵ and optical coherence tomography for structural imaging.⁶ Utilizing GPU for Doppler OCT imaging of the flow has already been presented but only for a single B-scan in a flow phantom.⁷ In this report the application of GPU for real-time OCT data processing is revisited and its utilization for blood flow examination in the human eye *in vivo* is demonstrated. The dynamic changes of the object are visualized as a 3-D "point cloud" (a finite set of points in a geometric space), which evolves in time.

The results presented were obtained with a computer workstation equipped with Intel® Core™ i7 920 (2.67 GHz) CPU, 6 GB RAM memory and a low-cost graphic card: NVIDIA® GeForce® GTX 580 with 3 GB device memory. It was tested with two laboratory-made, high-resolution spectral domain OCT systems. The first, designed for material examination, was equipped with a Superlum D-series Broadlighter D-855 (comprising two coupled superluminescent diodes) with central wavelength $\lambda_c = 845$ nm and full spectral width at half maximum $\Delta\lambda = 107$ nm, providing measured axial resolution $\delta z = 4$ μ m (in air). Spectra were collected with a line-scan CCD camera (e2v, AViiVA® SM2). The second OCT system was designed to examine the human retina. A femtosecond laser (Femtolasers, Fusion) was used as a light source ($\lambda_c = 810$ nm, $\Delta\lambda = 140$ nm, $\delta z = 3$ μ m). Spectra were collected with a CMOS camera (Basler, Sprint SPL4096-140 km). The data from both cameras were buffered in frame grabbers (National Instruments PCIe-1429).

The software was written in C++ programming language in Microsoft® Visual Studio 2010. All procedures for parallel processing on GPU were compiled with NVIDIA® CUDA™ compiler version 4.0. The OpenGL® 3.0 Library was used for visualization of results.

The developed software performs full numerical processing of the OCT data. This way, the tomograms are equivalent to those produced by the standard procedures implemented on the CPU. Specifically, the processing includes background subtraction, spectra remapping to wave-number space, numerical dispersion compensation, spectral shaping, and fast Fourier transformation. Finally, a logarithm of modulus of the resulting complex signal is calculated to obtain an A-scan Eq. (1):

$$g(z) = 20 \log |f(z)| = 20 \log \{ [FT^{-1}(G(k))] \}, \quad (1)$$

where k is the wave number, $G(k)$ is the remapped and preprocessed power spectral density of the light registered on the camera, $f(z)$ is the Fourier transform of the power spectral density, g is an A-scan, and z is in-depth coordinate.

In Doppler OCT analysis, phase differences for each corresponding frequency component between each two subsequent A-scans are calculated. The phase difference between the m 'th and a consecutive A-scan is given by Eq. (2):²

Address all correspondence to: Marcin Sylwestrzak, Institute of Physics, Nicolaus Copernicus University, Grudziadzka 5, PL-87-100 Torun, Poland. E-mail: mars@fizyka.umk.pl

$$\Delta\phi_m(z) = \arctg\left\{\frac{\text{Im}[f_m(z) \cdot f_{m+1}^*(z)]}{\text{Re}[f_m(z) \cdot f_{m+1}^*(z)]}\right\}$$

$$= \arctg\left\{\frac{\text{Re}[f_m(z)\text{Im}[f_{m+1}(z)] - \text{Re}[f_{m+1}(z)\text{Im}[f_m(z)]]}{\text{Re}[f_{m+1}(z)\text{Re}[f_m(z)] + \text{Im}[f_{m+1}(z)\text{Im}[f_m(z)]]}\right\}, \quad (2)$$

where a complex number $f_m(z)$ is defined for m 'th A-scan in Eq. (1). The change of the phase between $f_m(z)$ and $f_{m+1}(z)$ is caused by a very small (less than the axial resolution) movement of reflecting particles. In the case of oversampling and for known time Δt between consecutive acquisitions, the axial component (parallel to the light beam) of the flow velocity can be calculated² as follows:

$$v = \frac{\Delta\phi}{2nk\Delta t} = \frac{\lambda}{4n\Delta t} \frac{\Delta\Phi}{\pi}. \quad (3)$$

The unambiguous velocity measurement is limited to $v_{\pm \max} = \pm\lambda/4\Delta t$ due to phase wrapping.

The data acquisition, processing and visualization software utilizes two main CPU threads running on the host computer (Fig. 1). These two threads are in detail described below.

The acquisition thread is responsible for collecting the spectra, and therefore works synchronously with the spectrometer camera. Its main task is to pass data from the frame grabber to two alternative buffers in a PC RAM memory. The size of these buffers depends on the size of the acquired data defined in the scanning protocols. Our software supports three protocols: a "cross," "four slices," and a "3-D preview." The "cross" is used mostly for a fast preview and consists of two cross-sectional images (B-scans) collected in perpendicular directions. The "four slices" protocol, consists of four B-scans: three are collected in parallel and the fourth is perpendicular to them. The "3-D preview," is a raster scan generating a volume data and allows for real-time visualization of the object in 3-D as a "point cloud."

The main task of the visualization and processing thread is to invoke procedures for parallel data processing on the GPU. All custom-designed procedures are implemented in kernels, which are specific functions running in parallel in many GPU threads. The first kernel performs background subtraction, λ - k spectral remapping, numerical dispersion compensation and spectral shaping. Then the fast Fourier transformation is performed with the aid of the CUFFT library delivered by NVIDIA. Finally, the second kernel is put into operation to finalize calculation (all procedures after FFT) and to generate the textures presented on the screen. Several versions of this kernel have been prepared for different scanning protocols and modes of imaging [structural Eq. (1) or Doppler Eq. (2)].

All GPU threads are organized in vectors (blocks). Threads working inside a certain block can exchange data with other threads (in this block) through a fast shared memory. Each block of threads works on a single A-Scan because sharing data is necessary for remapping spectra to wave-number space. For each A-scan, 512 GPU threads are started, so each thread works on four data points of each spectrum. These blocks are organized in a grid (a two-dimensional matrix), with dimension sizes equal to the number of A-scans by the number of B-scans.

The proper design of the parallelism structure of the program is necessary for effective use of GPU. To reduce the demand of each thread for registers the same variables have been utilized for several different purposes. Additionally calling a single

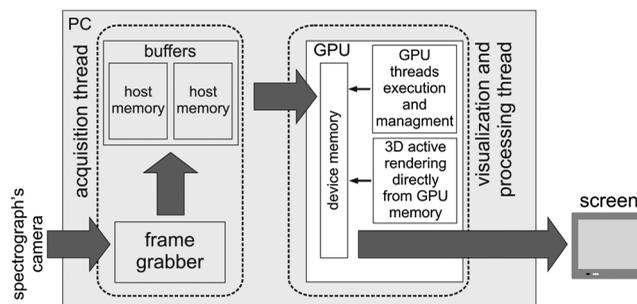


Fig. 1 Two main application threads working on the host computer.

precision functions such as $\text{logf}(x)$ instead of double precision $\text{log}(x)$, and replacing the time-consuming modulo function with bitwise "and" operation for power of 2 numbers significantly reduces the calculation time. Another important optimization is proper organization of the output matrices (textures for displaying the results) to ensure coalesced memory access what reduces the processing time by 37%.

It must be admitted that the data processing on GPU is necessary but insufficient for real-time 3-D imaging of OCT data. Fast visualization of the results is required. Our method⁸ utilized 2-D textures presented in three orthogonal directions with enabled transparency. However, without parallel processing, this method was noticeably slow. We managed to reduce the visualization time significantly by application of a pixel buffer object (PBO) for mapping textures to kernels within CUDA architecture.

The overall imaging rate depends on the execution times of three major processes: signal acquisition; data transfer time from PC host memory to the GPU (Table 1, col. 3); and processing and visualization time (Table 1, col. 4). In the most time-effective situation the data acquisition rate should match the rate of the processing and visualization. In such a case, frame rate would depend on the transfer time (Table 1, col. 3) and

Table 1 The performance of the software for structural and Doppler OCT imaging.

Protocol	Number of A-scans	Data transfer to the GPU	Reprocessing and visualization only	Total frame rate	
1	2	3	4	5	6
Cross:					
2 × 1600	3,200	12 ms	8,3 ms	120 fps ^a	49 fps
2 × 2000	4,000	13 ms	9,6 ms	103 fps	44 fps
Four slices:					
4 × 2000	8,000	29 ms	18 ms	55 fps	21 fps
4 × 5000	20,000	70 ms	43 ms	23 fps	9 fps
3D preview:					
100 × 100	10,000	33 ms	75 ms	13 fps	9 fps
140 × 140	19,600	70 ms	117 ms	9 fps	5 fps

^aScreen refreshment limit.

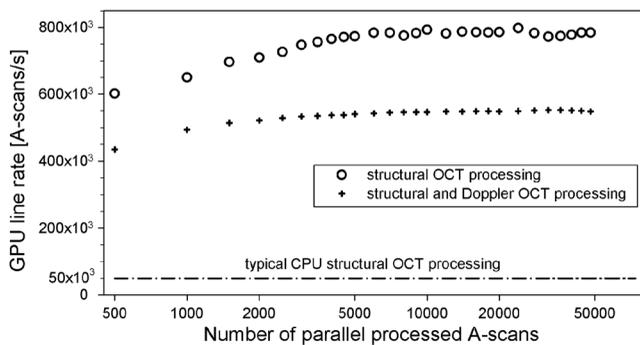


Fig. 2 GPU line rate as a function of the number of A-scans for structural and Doppler OCT.

processing time (Table 1, col. 4), and the overall frame rate would be given by the numbers in Table 1, col. 6. However, at the present level of the development of the technology, data acquisition is slower than data processing, and new data are rarely available for the processing software. Therefore, the same data set can be processed and displayed several times with frame rates given in Table 1, col. 5. This feature is used in our software for online adjustments of data processing (e.g., dispersion compensation) and display parameters (e.g., rotation angle, zoom). When the numerical analysis is taken into account (without transfers) the processing rate on GPU is about 800,000 spectra (2048 points each) per second for structural examination and 550,000 spectra per second in the case of Doppler OCT analysis (Fig. 2).

To demonstrate the capability of our software to perform real-time structural and Doppler OCT imaging we measured a flow phantom [Fig. 3(a), Video 1] and a retina of a healthy volunteer [Fig. 3(b), Video 2]. In both videos, the grayscale corresponds to the OCT structural information. The color scale codes axial component of the flow. Doppler signal was calibrated using a syringe pump with predetermined flow rate. Video 1 demonstrates real-time 3-D imaging of the bi-directional flow in the phantom (a 200 μm diameter glass capillary with 0.5% water solution of Intralipid®). Data were processed and displayed as a “point cloud” with the frame rate of 6 fps. The Intralipid® flow was controlled manually using a syringe. Video 2 demonstrates an example of the Doppler measurement of the blood flow in the human eye. In order to demonstrate the possibility of real-time targeting for Doppler measurements of the human retina, the “four slices” protocol was adopted. It combines high frame rate with high sampling density necessary for sensitive flow visualization over a 2 \times 2 mm area.

In conclusion, parallel GPU computations are very well-suited for OCT data processing. Optimization of the code allows to visualize structural and real-time Doppler information. Moreover, the ascendancy of the processing speed over data acquisition speed allows for processing of the same data set several times in order to optimize processing and/or visualization conditions in real-time. All this seems to open a gate for new applications of the OCT technique, especially if the development of GPU technology continues to be as rapid as at present.

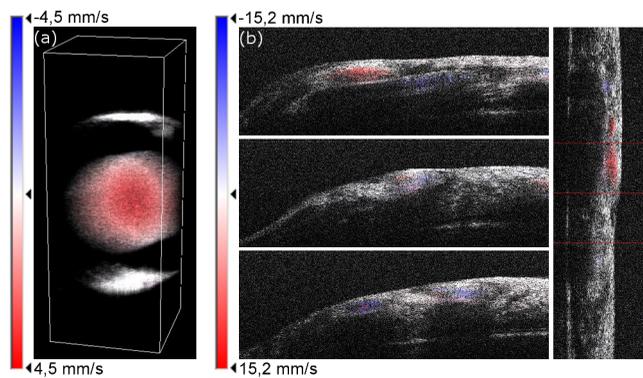


Fig. 3 (a) Real-time 3D imaging of the flow of homogeneous water solution of Intralipid® in glass capillary (200 μm diameter) in protocol “3D preview” (200 \times 100 A-scans) with frame rate 6 fps (Video 1). The flow rate varies between 0...0.4 $\mu\text{L/s}$...0...-0.4 $\mu\text{L/s}$. Cage dimensions (W, D, H) are: 0.3 \times 0.3 \times 0.8 mm. And (b) blood flow imaging near the optic head nerve (Video 2), four B-scans (2.2 \times 1.1 mm) build of 4000 A-scans each, recorded with frame rate of about 11 fps (Video 1, MPG, 1.79 MB) [URL: <http://dx.doi.org/10.1117/1.JBO.17.10.100502.1>]; (Video 2, MPG, 7.81 MB) [URL: <http://dx.doi.org/10.1117/1.JBO.17.10.100502.2>.]

Acknowledgments

Marcin Sylwestrzak and Danuta Bukowska acknowledges support from the “Step into the Future IV” program co-financed by the European Social Fund and Polish Government. Iwona Gorczynska and Maciej Szkulmowski acknowledges support by the Polish Ministry of Science and Higher Education (years 2009 to 2014). Maciej Wojtkowski acknowledges EURYI grant/award funded by the European Heads of Research Councils (EuroHORCs) together with the European Science Foundation (ESF—EURYI 01/2007PL) operated by the Foundation for Polish Science.

References

1. M. Wojtkowski et al., “In vivo human retinal imaging by Fourier domain optical coherence tomography,” *J. Biomed. Opt.* **7**(3), 457–463 (2002).
2. A. Szkulmowska et al., “Phase-resolved Doppler optical coherence tomography limitations and improvements,” *Opt. Lett.* **33**(13), 1425–1427 (2008).
3. I. Grulkowski et al., “Scanning protocols dedicated to smart velocity ranging in Spectral OCT,” *Opt. Express* **17**(26), 23736–23754 (2009).
4. X. Zhao, J. -J. Hu, and P. Zhang, “GPU-based 3-D cone-beam CT image reconstruction for large data volume,” *Int. J. Biomedical Imag.* **2009**, 149079 (2009).
5. W. Li, S. Dan, and D. C. Liu, “Optimized GPU framework for pulsed wave Doppler ultrasound,” in *4th Int. Conf. on Bioinformatics and Biomedical Eng. (iCBBE)*, pp. 1–4, IEEE, Chengdu (2010).
6. K. Zhang and J. U. Kang, “Real-time 4D signal processing and visualization using graphics processing unit on a regular nonlinear-*k* Fourier-domain OCT system,” *Opt. Express* **18**(11), 11772–11784 (2010).
7. H. Jeong et al., “Ultra-fast displaying spectral domain optical Doppler tomography system using a graphics processing unit,” *Sensors* **12**(6), 6920–6929 (2012).
8. M. Sylwestrzak et al., “Application of graphically oriented programming to imaging of structure deterioration of historic glass by optical coherence tomography,” *Proc. SPIE* **7391**, 73910A (2009)