# Real-time massively parallel processing of Spectral Optical Coherence Tomography data on Graphics Processing Units

Marcin Sylwestrzak*, Daniel Szlag, Maciej Szkulmowski, and Piotr Targowski
Institute of Physics, Nicolaus Copernicus University, ul. Grudziadzka 5, 87-100 Toruń, Poland

## ABSTRACT

In this contribution we describe a specialised data processing system for Spectral Optical Coherence Tomography (SOCT) biomedical imaging which utilises massively parallel data processing on a low-cost, Graphics Processing Unit (GPU). One of the most significant limitations of SOCT is the data processing time on the main processor of the computer (CPU), which is generally longer than the data acquisition. Therefore, real-time imaging with acceptable quality is limited to a small number of tomogram lines (A-scans). Recent progress in graphics cards technology gives a promising solution of this problem. The newest graphics processing units allow not only for a very high speed three dimensional (3D) rendering, but also for a general purpose parallel numerical calculations with efficiency higher than provided by the CPU. The presented system utilizes CUDA™ graphic card and allows for a very effective real time SOCT imaging. The total imaging speed for 2D data consisting of 1200 A-scans is higher than refresh rate of a 120 Hz monitor. 3D rendering of the volume data build of 10 000 A-scans is performed with frame rate of about 9 frames per second. These frame rates include data transfer from a frame grabber to GPU, data processing and 3D rendering to the screen. The software description includes data flow, parallel processing and organization of threads. For illustration we show real time high resolution SOCT imaging of human skin and eye.

**Keywords**: Optical Coherence Tomography, data processing, massively parallel processing, CUDA, GPU

## 1. INTRODUCTION AND MOTIVATION

Spectral Optical Coherence Tomography is an imaging technique which allows for non-contact and non-invasive examination of internal structure of the objects weakly absorbing and scattering light. It has been successfully used for *in vivo* examination of human eye[1, 2] for about a decade. Comparing to its ascender, the time domain OCT, Fourier domain techniques (including SOCT) have 2 to 3 orders of magnitude higher imaging speeds. This is one of the reasons of their domination on the ophthalmology market nowadays. Nevertheless, this advantage concerns mostly the data acquisition. Data processing on a main processor of the computer (CPU – Central Processing Unit) is generally slower than collecting of the data. Therefore, in case of processing data on CPU, real-time imaging is limited to a small number of tomogram lines (A-scans), and consequently to a few OCT cross-sections (B-scan) of limited quality. However, recent progress in graphics cards technology gives a promising solution to the problem of OCT data processing: the newest graphics processing units (GPU) allow not only for high speed three dimensional (3D) rendering, but also for general purpose parallel numerical calculations with efficiency higher than provided by the CPU.

One of the most popular technologies used for general purpose computing on GPU is CUDA™ (Compute Unified Device Architecture) introduced by NVIDIA® (Santa Clara, California). Initially used in computer games industry it permits to create more natural effects of fire, fluid, light, *et cetera*. It has been soon recognised as a powerful tool for general purpose parallel computations and applied for molecular dynamics calculations,[3] chemistry,[4] quantum physics,[5] and others. Since the processing of the SOCT data can be performed in parallel, GPUs have been also used in OCT technology.[6-8] Special modifications in the OCT devices were applied for simplification of the data processing. These adjustments reduced the necessary amount of calculations. However, the results were not as satisfactory as obtained with full numerical processing which include $\lambda - k$ remapping, numerical dispersion compensation, and spectral shaping. Software with full numerical processing on GPU has been presented[9] but only as proof of concept of utilising GPU for this purpose and for simplification of the visualisation a GLUT (*Open**GL** Utility **T**oolkit*) library was chosen. Its easy to use interface makes this toolkit very useful, but its utilisation is limited to a simple "one-window" applications. In this

[*] mars@fizyka.umk.pl

Optical Coherence Tomography and Coherence Techniques V, edited by Rainer A. Leitgeb,
Brett E. Bouma, Proc. of SPIE-OSA Biomedical Optics, SPIE Vol. 8091, 80910V
© 2011 SPIE-OSA · CCC code: 1605-7422/11/$18 · doi: 10.1117/12.889805

contribution, we present a new software solution, which takes advantage of two technologies: low-level WinAPI (Microsoft's application programming interface) for visualisation window management and native OpenGL for faster and more flexible 3D rendering. This software is fully reconfigurable and can work very efficiently with several different OCT scanning protocols. Our software performs all steps of data processing required for high resolution (HR) imaging and is compatible with a standard SOCT instrument. Complete set of the data processing procedures is executed in the GPU. The results are displayed on the screen directly from the GPU rather than returned to the computer RAM memory which is crucial for fast 3D rendering.

## 2. HARDWARE AND INSTRUMENTATION

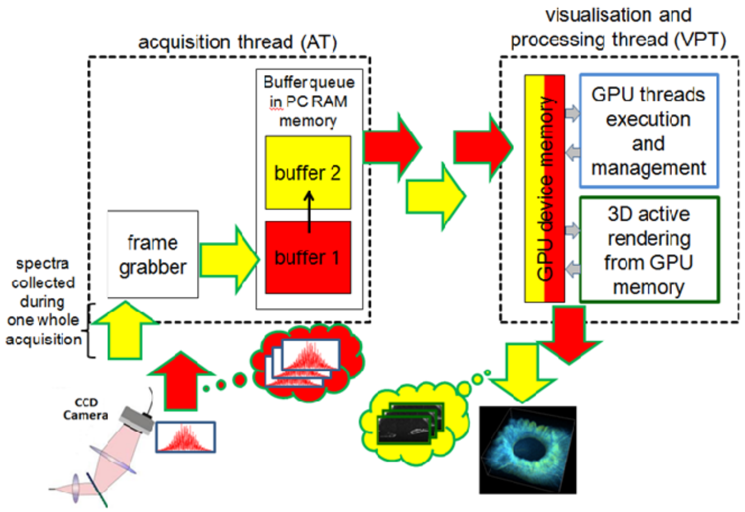
Results presented in this report were obtained with a workstation equipped with Intel® Core™ i7 920 (2.67 GHz) CPU with 6 GB RAM memory and low cost (less than ~$300) game-designed graphic card: NVIDIA® GeForce® GTX 285 with 2 GB device memory. To ensure fast acquisition of SOCT spectra, National Instruments PCIe-1429 frame grabber was employed, which allows transfer of 680 MB/s over 2 Camera Link cables. Our software was tested with two similar SOCT set-ups. The first one[10] was a laboratory-made Spectral OCT apparatus designed for biomedical applications. A superluminescent diode (Superlum, Ireland) was used as a light source. The measured imaging resolution was 4 μm axially and 13 μm laterally. The object was scanned with a pair of galvanometer scanners (Cambridge Technology, USA) and data was collected with a spectrometer equipped with a CMOS camera (Basler AG, Germany) Sprint SPL4096-140km working with the 128 kHz line rate in the 2048 pixel mode. The second system,[11] designed for material examination used a Superlum D-series Broadlighter as a light source providing 4 μm axial resolution in air. The spectrometer was based on linear CCD Atmel AViiVA® SM2 camera (e2v, UK) with 28 kHz line rate.

## 3. SOFTWARE ARCHITECTURE

The software was developed under Microsoft® Windows™ 7 Professional x64 operating system and was written in C++ programming language. The Microsoft® Visual Studio and Intel® C++ Compiler Professional Edition 11.1 with Intel® Integrated Performance Primitives library were used to ensure better efficiency of the data buffering. All procedures for GPU were prepared with NVIDIA® CUDA™ compiler version 3.2.16.[12] The OpenGL® 3.0 Library was used for visualization. Data acquisition was performed with IMAQ™ application programming interface (National Instruments, USA).

The presented software utilises two main threads running on the host computer. One of them is responsible for the data acquisition and the other for the data processing (Fig.1). The first (acquisition) thread is running synchronously with the OCT instrument. Its main task is to collect the spectra from the camera and transfer to the memory buffers. Each buffer contains data (spectra) of whole image – cube of points in case of 3D imaging. These buffers are organised in a two buffer queue. Initially, both buffers are empty. In typical mode of work (single spectrum collection time is longer than 10 μs) data processing is faster than data acquisition. Therefore, buffers are removed from the queue faster than added and the same set of data may be processed several times, to ensure fluent modifications of imaging parameters (contrast, brightness, image rotations or translations, *et cetera*) in real time. For the shortest exposition time of the CMOS camera (when each spectrum is exposed for 6.6 μs) data acquisition is faster than its processing. In this case, new data can be added to the two buffers queue if only the older data is removed (and processed). Therefore some buffers can be omitted during this mode of real time imaging.

In the acquisition thread (AT), a few simple pre-processing procedures are implemented after a new spectrum is added to the queue. First, the data is converted from two byte integer representation (native for the spectrometer cameras) to 32 bit floating point values to match the internal format of GPU. Then, a few simple pre-processing procedures are implemented: removing of edge data due to fly-back time of galvanometer scanners and the background spectra averaging to improve signal to noise ratio in the resultant images. During these operations all data is converted from two byte integer data representation (native for the spectrometer’s cameras) to 32 bit floating point values to match the internal format of GPU. This quick and simple pre-processing, which consist of operations such as additions, multiplications and data shift in linearly aligned memory is implemented using Intel® Integrated Performance Primitives library.


visualisation and processing thread (VPT)
acquisition thread (AT)
Bufferqueue in PC RAM memory
buffer 2
buffer 1
frame grabber
spectra collected during one whole acquisition
CCD Camera
GPU device memory
GPU threads execution and management
3D active rendering from GPU memory

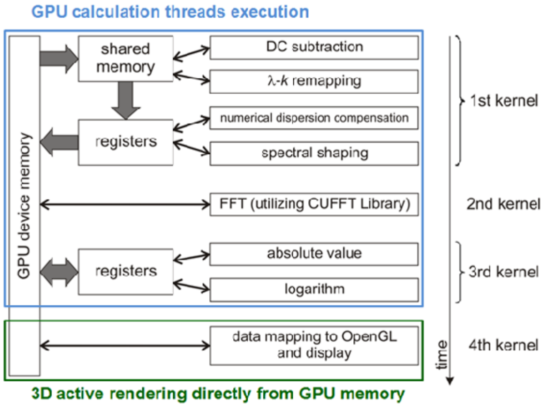

GPU calculation threads execution
GPU device memory
shared memory
registers
registers
DC subtraction
λ-k remapping
numerical dispersion compensation
spectral shaping
FFT (utilizing CUFFT Library)
absolute value
logarithm
data mapping to OpenGL and display
1st kernel
2nd kernel
3rd kernel
4th kernel
time
3D active rendering directly from GPU memory

Figure 1. Two main threads working on the host.

Figure 2. GPU threads execution.

The second main thread (visualization and processing thread – VPT in Fig. 1) is responsible for data processing and visualisation: removing data form buffer queue, transfer to the GPU, execution and management of GPU threads, preparing 3D virtual scene and data rendering on the screen. This thread runs independently of the acquisition thread and either downloads new data from the buffer queue if it is not empty, or processes the same data again for fluent data visualization.

The Graphics Processing Unit utilized in this study consists of 30 multiprocessors comprising 8 cores each. On the contrary to the main processor, the GPU cores comprised in every multiprocessor must execute simultaneously the same commands. The GPUs supporting CUDA™ technology have several types of memory with different parameters like size, latency, type of access. In order to fully exploit the GPU computational power the key problem to be solved is effective utilisation of various memory types present on board. For instance, the GPU device memory (Fig. 2) is the biggest one (several GB), but access to data may take several hundred of GPU processor cycles, therefore it is well suited for keeping input and output data. During calculation, the fastest data access is through registers, but they can allocate only 16 kB of data per multiprocessor. The completely unique is shared memory: it is fast, small, and it is the only type of memory which allows for exchange data between threads. It is utilised in the second step of our processing ($\lambda - k$ remapping of spectral data, Fig. 2) because the threads need simultaneous access to the whole spectrum. Then data is transferred to separate for each thread fast read-write registers with one clock cycle latency for numerical dispersion compensation and spectral shaping. Then data is returned to the device memory and the second kernel performs the Fourier transformation utilising CUFFT library (from NVIDIA®). Obtained A-scan is processed in the third kernel: absolute value and a logarithm (for better visibility of image details) are calculated. Data ready for displaying is returned to the GPU device memory. The forth kernel maps this memory to the textures presented on the screen.[13]

## 4. SOFTWARE PERFORMANCE

Data processing on GPU is as effective as many CUDA threads are executed in parallel. For architecture of the graphic card utilised in this study the maximum number of simultaneously running CUDA threads able to exchange data through shared memory is 512. Therefore it was necessary to associate each tread with four consecutive numbers of 2048 building a whole spectrum. Since the overall number of available CUDA threads is much larger and it is no data exchange between spectra, the significant increase of data processing speed is achieved when large number of A-scans is parallel processed. For our system this speed stabilizes at about 300,000 A-scans/s for data bigger than 24,000 A-scans (Fig. 3).

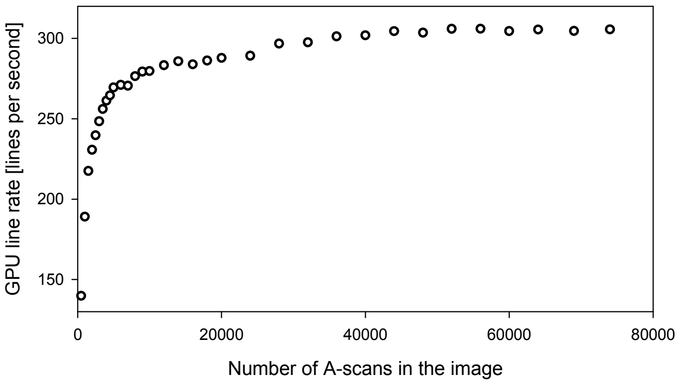


Figure 3. GPU line rate (step 4 in Fig. 4) as a function of number of A-scans.

The timing of data transfer, processing, and volume imaging are presented in Figure 4 using the 3D acquisition protocol comprising 100 B-scans build of 100 A-scans each as an example. Firstly data from the frame grabber is transferred to the buffer queue (step 2, Fig. 4) and then five A-scans are removed from each B-scan to discard artefacts caused by fly-back time of galvanometer scanners. The visualisation and processing thread (VPT) starts with checking for new data (step 3). If there is a new data set, it is taken from queue and transferred to the GPU for processing and rendering. If there is no data waiting for transfer, the GPU repeats processing with the old set.

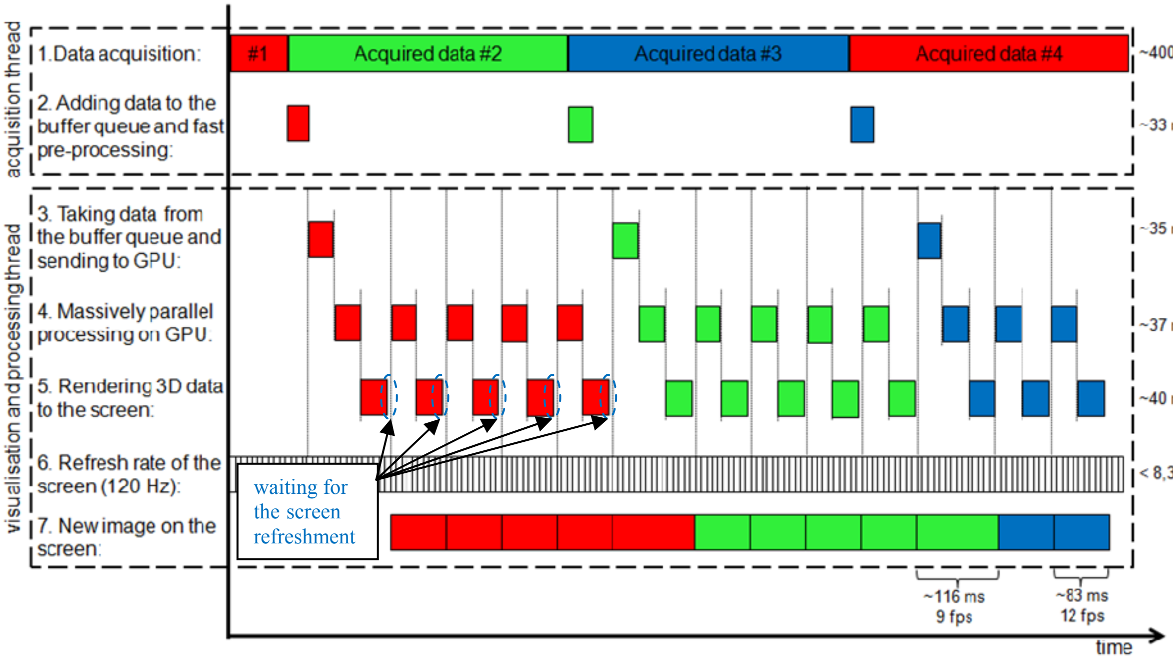

acquisition thread
1. Data acquisition:
#1
Acquired data #2
Acquired data #3
Acquired data #4
~400 ms
2. Adding data to the buffer queue and fast pre-processing:
~33 ms
visualisation and processing thread
3. Taking data from the buffer queue and sending to GPU:
~35 ms
4. Massively parallel processing on GPU:
~37 ms
5. Rendering 3D data to the screen:
~40 ms
6. Refresh rate of the screen (120 Hz):
< 8,3 ms
7. New image on the screen:
waiting for the screen refreshment
~116 ms
9 fps
~83 ms
12 fps
time

Figure 4. Timing of data transfer, processing, and volume imaging within two main threads: acquisition thread and visualisation and processing thread. The width of each color box is proportional to the required time for relevant step. The values are measured during imaging in 3D protocol with 100 B-scans build of 100 A-scans, the resultant image is limited to 800 pixels in depth .

In developed software double buffering for visualisation in OpenGL is utilised: one image is prepared while another one is presented on the screen. In this case OpenGL buffers containing screen frames are swapped according to the refresh rate of the screen. This solution force the VPT to be synchronised with refresh rate of the screen (in our case a LCD panel was able to work with refresh rate of 120 Hz). Therefore, as shown in Figure 4, after each data rendering the VPT has to wait with swapping OpenGL buffers for the next screen refreshment strobe – no longer than 1/120 Hz = 8.3 ms in our case. Thus the processing time by VPT is always a multiple of the reciprocal of the screen refresh rate (8.3 ms in our case). It is longer if a new data is processed (116.2 ms), and shorter if the same data is processed again (83 ms). These numbers lead to the frame rate of OCT imaging of 3D data of about 8.6 and 12 frames per second (fps) respectively. For the biggest acceptable by our system data set build of 140 B-scans comprising 140 A-scans each these rates are still acceptable and equal 7.1 and 4.6 fps. Generally, in case of 3D imaging it is rather expected that system will be used with the highest acceptable number of A-scans to ensure best available quality of images.

In the case of 2D imaging, when a few of B-scans (very often just one) are presented on the screen, it is important to manage properly the number of A-scans. It is well known that a significant increase of the image quality is achieved when a few adjoining A-scans are averaged for displaying on the screen. As a rule of thumb it may be assumed that the number of A-scans in a B-scan should be two to threefold higher than the number of available pixels on screen. Therefore B-scans composed of 2 000 to 5 000 A-scans are usually collected. Since the VPT processing time is roughly proportional to a number of A-scans and simultaneously synchronisation with refreshing of the screen is enforced, some numbers of A-scans will be preferred for efficient imaging. Taking in account that the data acquisition time always dominates over the data processing, this feature is especially important in the case re-processing of the same data (steps 4 and 5 in Fig. 4). In Fig. 5a an example of useful 2D imaging is given and in Fig. 5b a real processing time (a combination of steps 4 and 5) is compared with an imaging time (defined as a time necessary for re-processed data to be presented on the screen) as a function of a total number of A-scans to be imaged. As one can see from the figure, for 120 Hz monitor optimal number of A-scans are 1200, 3000, and 4600, whereas for 60 Hz monitor only in case of 3000 A-scans the processing power is used effectively.

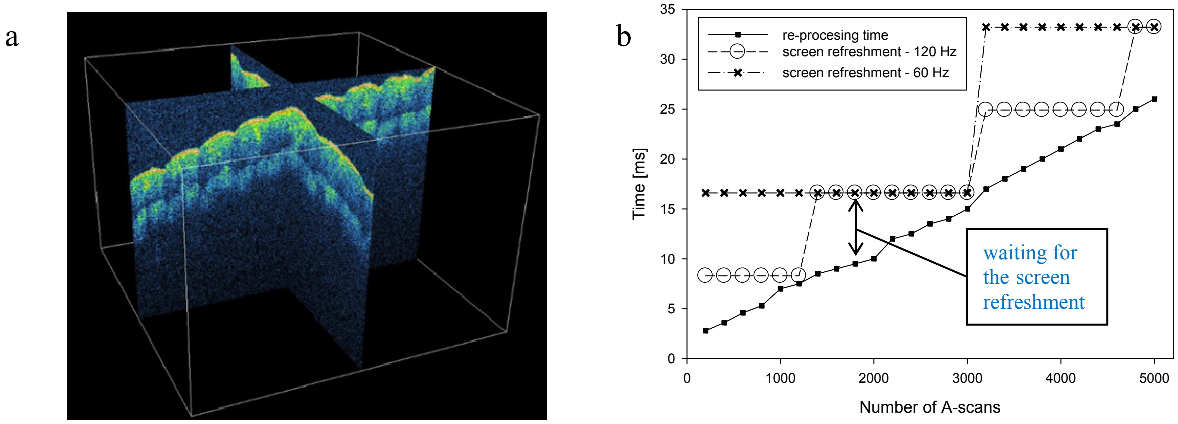


Figure 5a. An example of two perpendicular cross sections comprising 500 A-scans each. Human finger skin is shown. Cage dimensions (W,D,H): 3x3x1,8 mm, b: Comparison of VPT processing times (for 120 Hz and 60 Hz monitors) with the re-processing time (steps 4 and 5 in Fig. 4) as a function of number of A-scans

## 5. CONCLUSIONS

In the SOCT systems utilising the main processor (CPU) for data processing, the main limitation of imaging speed origins from the time of data processing. In our (GPU-based) set-up, despite of the mode of imaging (3D/2D), the efficiency of data processing is so high that the overall speed is constrained by the data acquisition process. In the case of 3D imaging it is due to massively parallel processing advantage, whereas in case of 2D imaging it is caused by the extremely fast rendering to screen. Therefore, at present stage of GPU technology further increase of real time imaging rate is increase the data acquisition speed. It is possible to some extend by lowering of exposition time, but then must be compensated by increased power of the probing light. This drawback seems fundamental, especially for opthalmological imaging.

It is worthwhile to note that the above efficiency analysis does not take into account storage of the obtained images to disk. This procedure is time- and disk space-consuming. However, by dint of HR real time imaging it is usually possible to reduce an amount of stored data by more precise choice of the place of examination.

Presented software might be useful for dynamic measurements of pupil reactivity (Fig. 6) and to examine variation of human lens curvature during the process of accommodation. Additionally, it might be helpful to clarify whether ocular surface expansion and changes in ocular volume could cause some fine variation of corneal radius.[14]

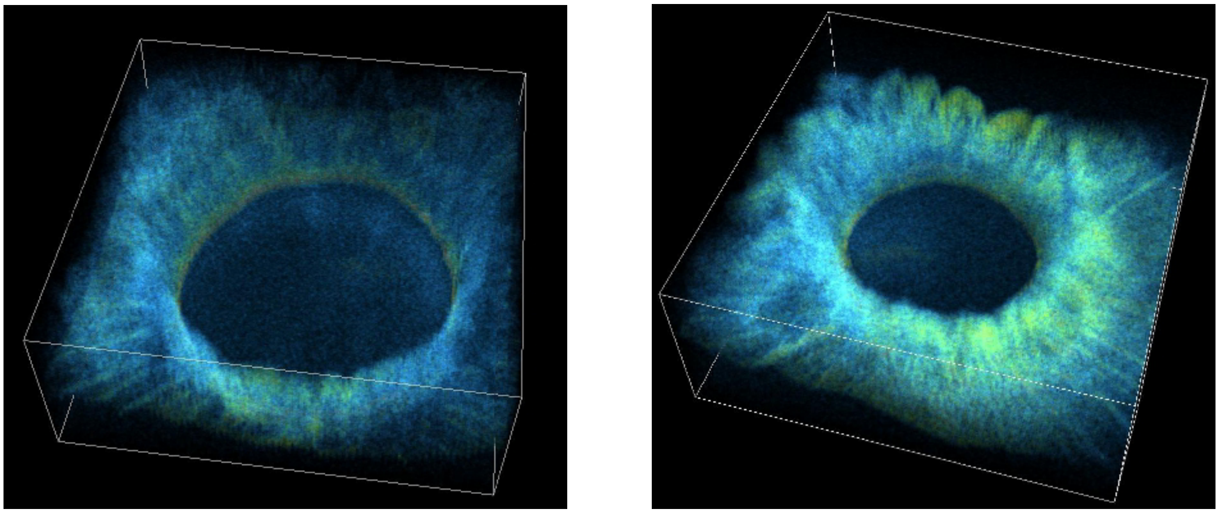


Figure 6. An example of 3D, real time images comprising 140 × 140 A-scans. Human eye's pupil is presented before and after light stimulus. Cage dimensions (W,D,H): 8x8x1,4 mm.

## ACKNOWLEDGEMENT

This project is funded by Polish Government Research Grant through years 2008–2011. M.Syl. and D.Sz. acknowledge support from the "Step into the Future III" program co-financed by the European Social Fund and Polish Government. M. Sz. is supported by the Foundation for Polish Science (scholarship START 2010).

## REFERENCES

[1] Wojtkowski, M., Bajraszewski, T., Gorczynska, I., Targowski, P., Kowalczyk, A., Wasilewski, W. and Radzewicz, C., "Ophthalmic imaging by spectral optical coherence tomography", American Journal of Ophthalmology 138(3), 412-9 (2004).

[2] Wojtkowski, M., Srinivasan, V., Fujimoto, J. G., Ko, T., Schuman, J. S., Kowalczyk, A. and Duker, J. S., "Three-dimensional retinal imaging with high-speed ultrahigh-resolution optical coherence tomography", Ophthalmology 112(10), 1734-46 (2005).

[3] Stone, J. E., Phillips, J. C., Freddolino, P. L., Hardy, D. J., Trabuco, L. G. and Schulten, K., "Accelerating molecular modeling applications with graphics processors", Journal of Computational Chemistry 28(16), 2618-2640 (2007).

[4] Ufimtsev, I. S. and Martínez, T. J., "Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-Electron Integral Evaluation", Journal of Chemical Theory and Computation 4(2), 222-231 (2008).

[5] Gutierrez, E., Romero, S., Trenas, M. A. and Zapata, E. L., "Parallel Quantum Computer Simulation on the CUDA Architecture ", Lecture Notes in Computer Science 5101, 700-709 (2008).

[6] Van der Jeught, S., Bradu, A. and Podoleanu, A. G., "Real-time resampling in Fourier domain optical coherence tomography using a graphics processing unit", J Biomed Opt 15(3), 030511 (2010).

[7] Probst, J., Koch, P. and Hüttmann, G., "Real Time 3D Rendering of Optical Coherence Tomography Volumetric Data", Proc SPIE 7372, 7372_0Q (2009).

[8] Zhang, K. and Kang, J. U., "Real-time 4D signal processing and visualization using graphics processing unit on a regular nonlinear-k Fourier-domain OCT system", Optics Express 18(11), 11772-11784 (2010).

[9] Sylwestrzak, M., Szkulmowski, M., Szlag, D. and Targowski, P., "Real-time imaging for Spectral Optical Coherence Tomography with massively parallel data processing", Photonics Letters of Poland 2(3), 137-139 (2010).

[10] Grulkowski, I., Gorczynska, I., Szkulmowski, M., Szlag, D., Szkulmowska, A., Leitgeb, R. A., Kowalczyk, A. and Wojtkowski, M., "Scanning protocols dedicated to smart velocity ranging in Spectral OCT", Optics Express 17(26), 23736-23754 (2009).

[11] Targowski, P., Ostrowski, R., Marczak, J., Sylwestrzak, M. and Kwiatkowska, E. A., "Picosecond laser ablation system with process control by Optical Coherence Tomography", Proc. SPIE 7391, 739115 (2009)

[12] NVIDIA, [NVIDIA CUDA C Programming Guide Version 3.2], (2010).

[13] Sylwestrzak, M., Kwiatkowska, E. A., Karaszkiewicz, P., Iwanicka, M. and Targowski, P., "Application of graphically oriented programming to imaging of structure deterioration of historic glass by Optical Coherence Tomography ", Proc. of SPIE 7391, 739109-1 (2009).

[14] Kowalska, M., Kasprzak, H., Iskander, D. R., Danielewska, M. and Mas, D., "Ultrasonic in-vivo measurement of ocular surface expansion", IEEE Transactions on Biomedical Engineering 58(3), 674-680 (2011).