

(1) Overview

Title

ABC-OCT - A cross-platform implementation of real-time Fourier-Domain Optical Coherence Tomography

Paper Authors

1. Nandakumar, Hari;
2. Srivastava, Shailesh.

Paper Author Roles and Affiliations

1. Department of Physics, Sri Sathya Sai Institute of Higher Learning, Prasanthi Nilayam, AP - 515134, India.
2. Department of Physics, Sri Sathya Sai Institute of Higher Learning, Prasanthi Nilayam, AP - 515134, India.

Abstract

ABC-OCT, Affordable B-scan Camera-based Optical Coherence Tomography, implements Fourier-Domain Optical Coherence Tomography with real-time display using cross-platform C++ and the OpenCV framework. The software can be compiled using current versions of GCC for *nix/Mac and Microsoft® Visual Studio for Windows. Full functionality of ABC-OCT needs the camera SDK from QHYCCD and a QHYCCD camera to be connected; but the code can be easily modified to support other camera drivers, as is shown by an included demo version which can use any installed webcam. The code is made available under the MIT license. The software is available from GitHub (<https://github.com/hn-88/FDOCT>).

Keywords

Optical Coherence Tomography; Imaging; C-plus-plus; QHYCCD; webcam; Spectral Domain OCT; SD-OCT

Introduction

Optical Coherence Tomography (OCT) is a non-invasive imaging methodology which attains fine resolutions [1] of several microns similar to microscopy, with a common limitation of penetration depths of a few millimetres. Since its introduction in Ophthalmology in the early 1990s, OCT has cycled through several generations [2] - Time-Domain or TD-OCT and Fourier-Domain or FD-OCT, with its two variants Spectral-Domain or SD-OCT, and Swept-Source or SS-OCT. FD-OCT has an SNR advantage [3] over the earlier TD-OCT. Most commercial OCT systems are shipped with bundled software which does the data processing necessary for FD-OCT. A large number of open-sourced packages exist, many of them implemented in MATLAB™, which take in the data from the commercial OCT systems and do post-processing such as segmentation [4]. Some groups have also open-sourced GPU-based [5] OCT pipelines. Our research effort is to produce low-cost OCT [6] implementations, along with the software required for data acquisition and processing.

Our focus on lowering cost precludes SS-OCT due to relatively high costs of swept-sources. From our initial [6] TD-OCT implementation, we moved on to FD-OCT to shorten acquisition times as is needed for in-vivo biological samples. We found parallel SD-OCT [7] to be an architecture which could support a low-cost implementation. The software reported here implements data acquisition and processing without the need for expensive components like GPU-accelerated high-throughput data-acquisition cards. Simplicity and ease of use were our main considerations.

The cost of OCT devices represents a significant barrier to entry into the field, preventing the adoption of OCT for a wider range of applications. Commonly available commercial OCT devices cost more than five figures in US dollars till recently [2]. This software would be useful for optics researchers who wish to assemble lab-built camera-based parallel SD-OCT devices for one-tenth the cost of commercial devices. The detailed descriptions of our device designs, published separately [8], along with this software, help in the democratization of OCT. Such low-cost OCT devices are not just useful for clinical use; they also encourage the use of OCT devices in University laboratories, enabling inter-disciplinary work with Material Science, Chemistry or Biology departments. Compared to electron microscopy, OCT tomograms are less expensive and have additional advantages of ease of sample preparation and of being non-destructive. Tomographic imaging of crystal imperfections, biofilms, measurements of refractive index and dispersion of samples, volume renderings of micro-machined or laser etched parts and many more applications can result from inter-departmental collaborations using OCT. Many more innovative investigations could result from lowering the cost barrier to entry for OCT imaging.

Researchers in OCT commonly use the terms A-scan and B-scan. A-scan refers to the one-dimensional reflectivity plot at one point on the sample, along its depth. B-scan refers to a 2-D plot of sample reflectivity, with the lateral extent along the sample shown on the horizontal axis, and the depth axis going into the sample being depicted as the downward vertical axis. This is illustrated in Figure 1. We refer to B-scans repeatedly in our discussion.

Implementation and architecture

ABC-OCT, Affordable B-scan Camera-based Optical Coherence Tomography, is implemented in C++ using the OpenCV [9] framework. It is compatible with the platforms supported by OpenCV, and has been tested on Windows 8.1, Windows 10 and Linux (Mint 18.1, kernel 4.4.0). We use a version including the camera SDK from QHYCCD [10] of our camera (QHY5L-II M) compiled from BscanFFT.cpp for our research. In case a camera from another manufacturer is being used, if an SDK is supplied, our code can easily be modified to use that SDK. To demonstrate this, we have included a version of our software which will work with any available webcam, which can be compiled from BscanFFTwebcam.cpp.

The block diagram presented in Figure 2 shows the processing steps being carried out. A screenshot of our software in action is presented in Figure 3, with the various windows labelled.

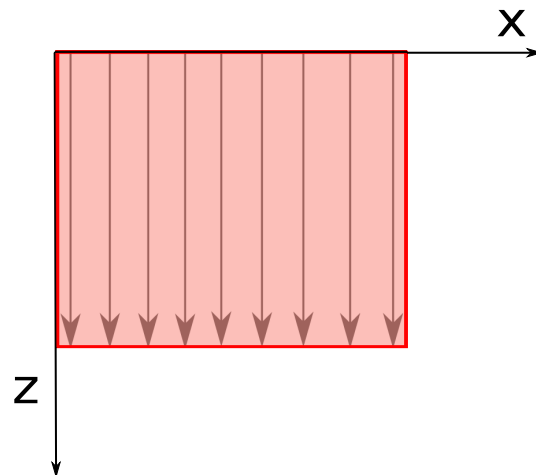


Figure 1: *Illustration of A-scans and B-scans in OCT. The intensity plot of the sample reflectance from various depths along the arrows are A-scans, while the collection of A-scans in the shaded area is a B-scan. The downward z-axis is into the sample. The x-axis shows the lateral extent of the sample.*

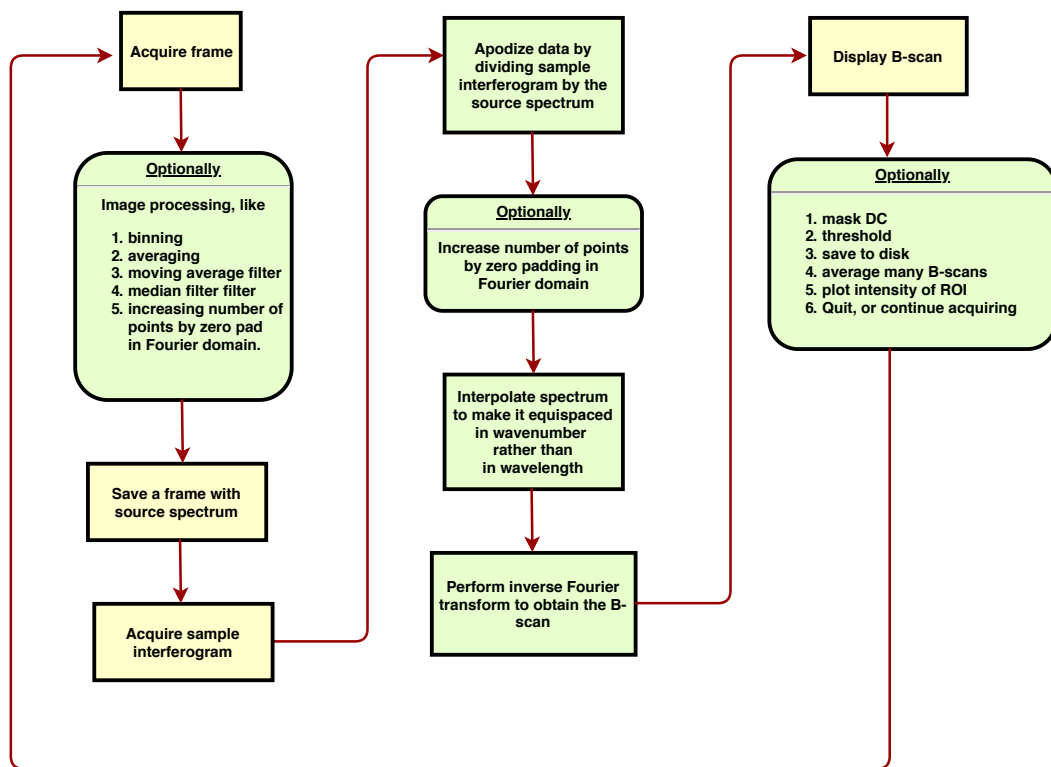


Figure 2: *Flowchart of operations performed by our ABC-OCT software.*

Quality control

The software has been tested for both algorithmic correctness as well as for cross-platform operation by running on multiple platforms, as listed in the availability section. The test images included in the source repository were used to verify correctness of the code, by comparing the results of the included Matlab-compatible

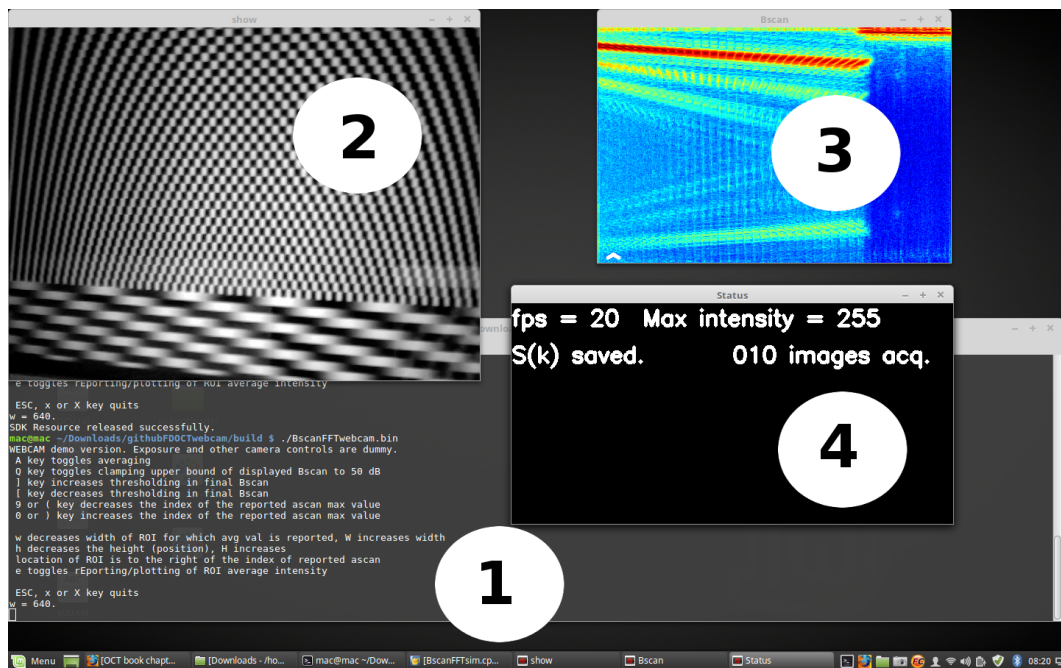


Figure 3: Screenshot of ABC-OCT in action. The labelled windows are - 1. Terminal window, where some status messages and error messages appear; 2. Display window, showing the camera view; 3. The computed B-scan in false colour; 4. Status window with information about capture parameters.

GNU Octave code and our C++ code. The BscanFFTsim.cpp file, when compiled, creates a test version of our software which can directly read test image files. To quickly test the webcam version of our software, the user just needs to run it and display the included test images in front of the webcam, which can be done using a mobile phone, for example.

A possible test procedure for the webcam version is:

1. Copy the images included in our code repository to a smartphone - `imgi.png` and `backg.png` which are located in the Matlab files directory of our source repository.
2. Display `backg.png` full-screen on the phone using the Gallery app or equivalent.
3. Start the webcam version of our software - of which binaries are included in the Release.
4. Hold up the phone to the webcam until the displayed screen shows that the mobile phone screen covers the entire field of view of the webcam, and it is in focus.
5. Press `b` to capture the background. (The software needs to have focus for this, not the commandline window.) This would make the B-scan screen change colour from being uniformly blue to a varying display, indicating that the background has been captured. The mobile phone can now be removed from its position in front of the webcam.
6. Display `imgi.png` on the mobile phone and hold it up to the webcam, just as in step 4.
7. You would now see a display of the B-scan corresponding to the interferogram in `imgi.png`. This can be saved by pressing `s`.
8. The file is saved in a sub-directory of the present working directory, named with

a date-timestamp followed by a descriptive string as set in the ini file.

(2) Availability

Operating system

The software has been tested by compiling and running on Microsoft®Windows 8.1, Microsoft®Windows 10 and Linux Mint 18.1. Software using the same QHYCCD SDK has been successfully run on Windows XP. Hence, it is likely that compiling with suitable architecture flags (32-bit or 64-bit) would enable this software to run on the wide range of OS families supported by the OpenCV framework and the QHYCCD SDK. The included Windows binaries have been tested and found to be running on Windows 8.1 and Windows 10. The included AppImage Linux binaries have been tested and found to be running on Linux Mint 18.1, Ubuntu 16.04 LTS and Fedora 24. Newer distros are likely to have problems with the AppImage due to libc incompatibilities, and hence compiling from source is recommended.

Programming language

The software is written in C++, and has been tested by compiling with GCC version 5.4.0 and Microsoft®Visual Studio Community version 15.9. It is likely that the software will compile and run on the wide range of architectures supported by the OpenCV framework and the QHYCCD SDK.

Additional system requirements

Installed size of the ABC-OCT software is only a few MB, but a modern PC with at least 4GB of RAM is recommended. The default settings make the software use slightly more than 1 GB of RAM, in order to optimize performance. The raw data dumps saved also use up disk space; typically 1 GB for 50 frames of capture data with saving of intermediate frames enabled. The full functionality of the software, with exposure control and arbitrary capture frame size needs a QHYCCD (brand name) camera and its SDK to be installed. The webcam version needs a functioning webcam installed on the system. Keyboard inputs are used to control the software.

Dependencies

The ABC-OCT software requires the OpenCV framework. opencv-3.3.1 and opencv-3.4.0 have been tested, higher versions should also work. Realtime display requires a camera SDK from QHYCCD, or any installed webcam.

List of contributors

Hari Nandakumar did the software development and is its current maintainer. Shailesh Srivastava supervised the development process.

Software location:

Archive

Name: Zenodo

Persistent identifier: <https://doi.org/10.5281/zenodo.3403735>

Licence: MIT license.

Publisher: Hari Nandakumar.

*REFERENCES**REFERENCES*

Version published: 1.12b.

Date published: 25/03/19

Code repository GitHub

Name: ABC-OCT

Persistent identifier: <https://github.com/hn-88/FDOCT>

Licence: MIT license.

Date published: 25/03/19

Language

English (UK).

(3) Reuse potential

Our ABC-OCT software enables anyone with access to an undergraduate level optics lab to assemble and use a parallel Fourier-Domain Optical Coherence Tomography (FD- OCT) instrument. Even the webcam demo version enables qualitative tomographic imaging. The software has been made freely available to the community to further encourage the use of OCT imaging in student laboratories and elsewhere.

Support

The MIT License has been selected to ensure that the code is available with minimum restrictions. Support for modifying and using the software is available through GitHub issues page, or by contacting the corresponding author.

Acknowledgements

We thank Bhagawan Sri Sathya Sai Baba, the founder Chancellor of our University, who guided and encouraged us throughout this work.

Funding statement

This work has not been funded by any grants.

Competing interests

The authors declare that they have no competing interests.

References

- [1] Wolfgang Drexler and James G Fujimoto. *Optical Coherence Tomography: Technology and Applications, Second Edition*. Springer International Publishing Switzerland, 2015.
- [2] James Fujimoto and Eric Swanson. “The development, commercialization, and impact of optical coherence tomography”. In: *Investigative ophthalmology & visual science* 57.9 (2016), OCT1–OCT13. DOI: 10.1167/iovs.16-19963. URL: <https://iovs.arvojournals.org/article.aspx?articleid=2536019>.
- [3] Johannes F De Boer et al. “Improved signal-to-noise ratio in spectral-domain compared with time-domain optical coherence tomography”. In: *Optics letters* 28.21 (2003), pp. 2067–2069. DOI: 10.1364/OL.28.002067. URL: <http://ol.osa.org/abstract.cfm?URI=ol-28-21-2067>.

REFERENCES

REFERENCES

- [4] Markus A. Mayer et al. “Retinal Nerve Fiber Layer Segmentation on FD-OCT Scans of Normal Subjects and Glaucoma Patients”. In: *Biomed. Opt. Express* 1.5 (Dec. 2010), pp. 1358–1383. DOI: 10.1364/BOE.1.001358. URL: <http://www.osapublishing.org/boe/abstract.cfm?URI=boe-1-5-1358>.
- [5] Yifan Jian, Kevin Wong, and Marinko V Sarunic. “GPU accelerated OCT processing at megahertz axial scan rate and high resolution video rate volumetric rendering”. In: *Optical Coherence Tomography and Coherence Domain Optical Methods in Biomedicine XVII*. Vol. 8571. International Society for Optics and Photonics. 2013, 85710Z. DOI: 10.1117/12.2006670. URL: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/8571/85710Z/GPU-accelerated-OCT-processing-at-megahertz-axial-scan-rate-and/10.1117/12.2006670.short>.
- [6] Hari Nandakumar, Adithya Kokkodu Subramania, and Shailesh Srivastava. “Sub-4-micron full-field optical coherence tomography on a budget”. In: *Sādhanā* 43.6 (2018), p. 97. DOI: 10.1007/s12046-018-0881-3. URL: <https://www.ias.ac.in/article/fulltext/sadh/043/06/0097>.
- [7] Branislav Grajciar et al. “Parallel Fourier domain optical coherence tomography for in vivo measurement of the human eye”. In: *Optics Express* 13.4 (2005), pp. 1131–1137. DOI: 10.1364/OPEX.13.001131. URL: <http://www.opticsexpress.org/abstract.cfm?URI=oe-13-4-1131>.
- [8] Hari Nandakumar and Shailesh Srivastava. “Low cost open-source OCT using undergraduate lab components”. In: *Optical Coherence Tomography and Its Non-medical Applications*. Ed. by Michael Wang. IntechOpen, 2019.
- [9] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000). URL: <https://github.com/opencv/opencv>.
- [10] Jan Soldan. “QHYCCD Linux SDK”. This repository contains a new (2017+) QHYCCD’s Ubuntu Linux 64 and ARM V8 (Raspberry PI3, Odroid XU4) deliveries. Oct. 2018. URL: https://github.com/qhyccd-lzr/QHYCCD_Linux_New.

Copyright Notice

Authors who publish with this journal agree to the following terms:

Authors retain copyright and grant the journal right of first publication with the work simultaneously licensed under a Creative Commons Attribution License that allows others to share the work with an acknowledgement of the work’s authorship and initial publication in this journal.

Authors are able to enter into separate, additional contractual arrangements for the non-exclusive distribution of the journal’s published version of the work (e.g., post

REFERENCES

REFERENCES

it to an institutional repository or publish it in a book), with an acknowledgement of its initial publication in this journal.

By submitting this paper you agree to the terms of this Copyright Notice, which will apply to this submission if and when it is published by this journal.