IMAGE QUALITY IMPROVEMENT IN MIDDLE EAR OPTICAL COHERENCE TOMOGRAPHY

DALHOUSIE UNIVERSITY
FACULTY OF ENGINEERING

Chen Su; Jun Wan; Matthew Townsend

Department of Electrical and Computer Engineering

Supervisors:
Dr. Rob Adamson (External); Dr. Jeremy Brown (Internal)

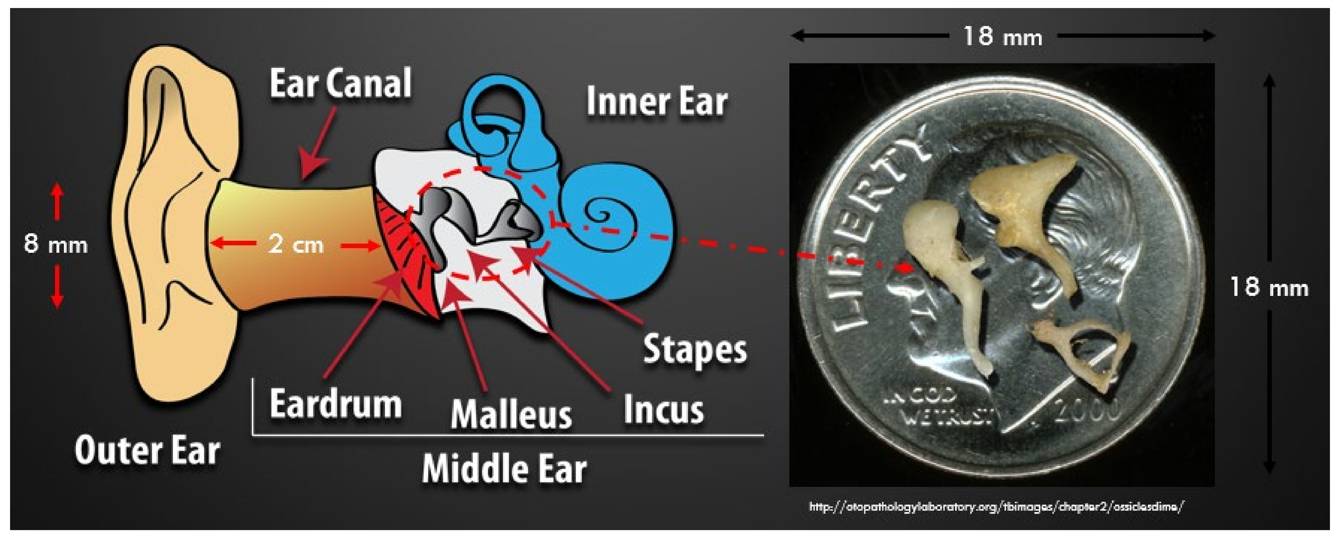
Special Thanks:
Joshua Farrell; Dan MacDougall

Overview

Objective

Develop a set of image processing modules for the Ossiview middle ear optical coherence tomography imaging system under development in the Dr. Adamson lab.

Middle Ear Anatomy & Scale



Design Flow

The respective algorithms are designed and tested in Python 2.7 and then reimplemented in Visual Studio 2017 using CUDA 10.0.

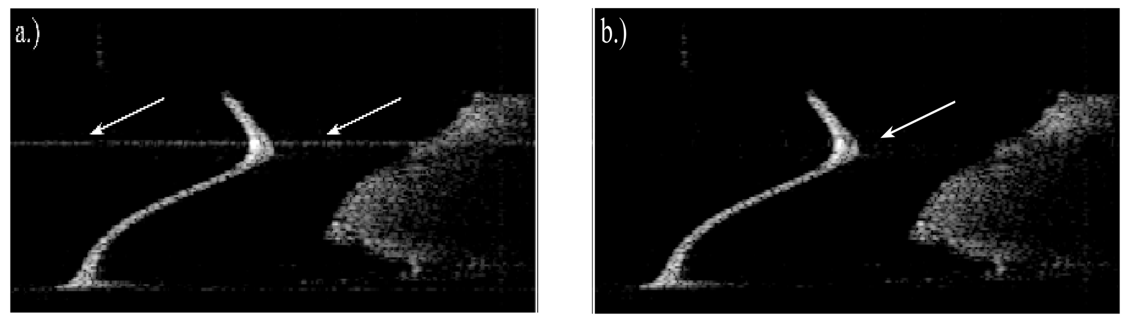
Develop Algorithm in Python → Test algorithm → Document algorithm → Reimplement algorithm in Cuda C++ → Validate image processing module → Merge into mainline codebase

Wavelet-based Artefact Removal

This project is to design a 3D tool to remove noise from OCT images.

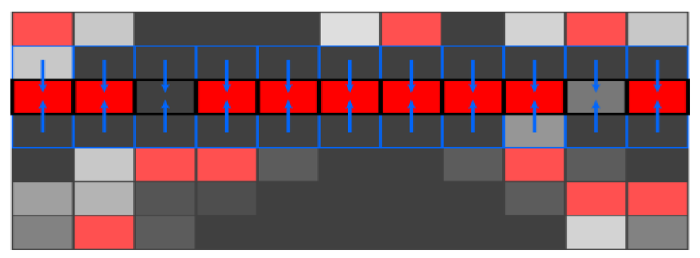
What is the noise in OCT image?

If the sample dynamic range exceeds the PSF dynamic range, the PSF noise floor will appear in the image for lines containing the bright reflectors. This effect manifests itself as artefactual lines that extend over the full depth of the image for image lines that contain a bright reflector as shown in Figure below, Figure a. is the image with artefact line, Figure b. is the image without noise:



The algorithm of artefact line removal:

To Remove the noise from the image without distorting the original image in a more efficient way, a algorithm of 3D artefact line removal is designed. In this project, one OCT image contains 330*256 pixels of which the brightness ranging from 60 to 90dB. The algorithm is a 3*3 kernel which calculates the intensity of middle line and compares it with the left and right lines to detect the artefact. Once the kernel finds it, then it will average the intensity of middle line (Farrell, 2018).



Special characteristic: The 3D tool loads all the 256 image slices and process them on GPU at the same time. The processing time is improved from about 110 seconds to 5 seconds. To process 256 slices at the same time, a three-dimensional GPU memory is used. The total number of pixels of a 3D image is 330*256*256. Therefore, 512 threads and 42240 blocks are allocated on GPU.

Calculation:

- Convert binary data from the OCT image into intensity in dB:

$$intensity = 20 * \log_{10}(original\ binary\ data)$$

- The method of allocate memory on GPU:
  - Block dimension in threads: 2*4*64
  - Grid dimension in blocks: 165*64*4

OCT Angiography

Background

Optical Coherence Tomography Angiography (OCTA) is a non-invasive approach that can identify blood vessels by differences in the OCT signal versus time between that arising from moving scatters in blood and that due to the surrounding largely static tissue.

OCTA algorithms can be generally categorized into the following groups: phase-, amplitude- and complex-based algorithms (Zhang, 2015).

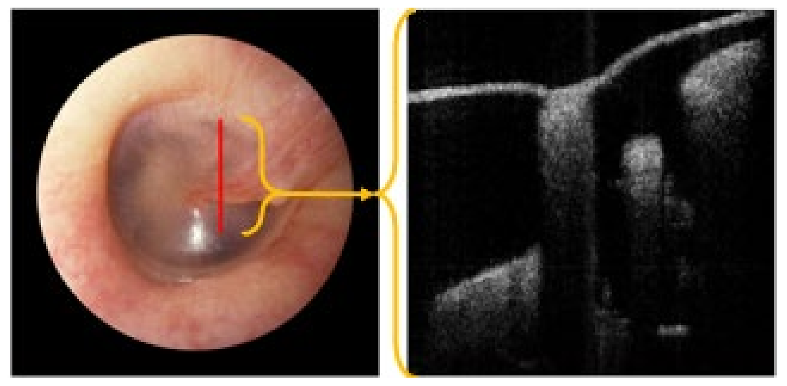


Fig.1 Eardrum *Fig.2 OCT Image*

OCT-microangiography (OMAG)

$$I_{OMAG}(x,z) = I_{OMAG}(x,z) = \frac{1}{N-1}\sum_{i=1}^{N-1}|C_{i+1}(x,z) - C_i(x,z)|$$

where

- φ is the phase information of OCT signal in the i'th repeated measurement
- $C_i(x,z)$ is the corresponding complex information expressed by $C_i = A_i * e^{j\Phi}$.

Result

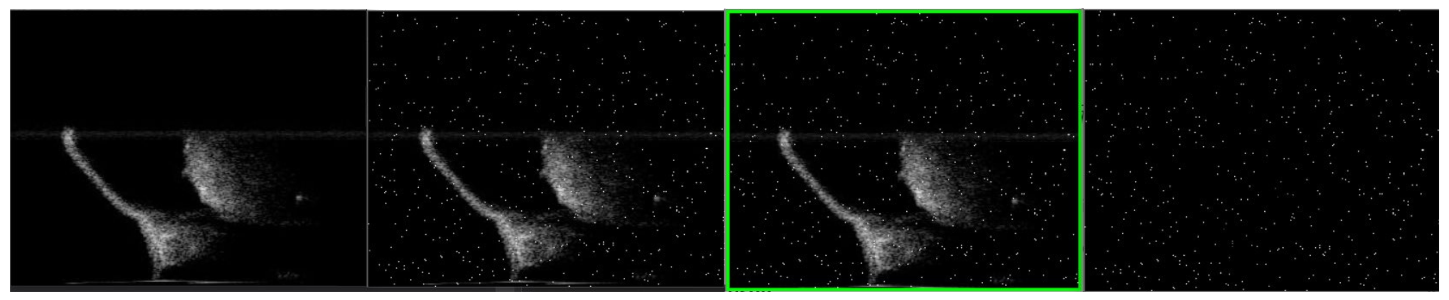


Fig.1 Source Image *Fig.2 Add Noise* *Fig.3 Modified Image* *Fig.4 Result Image*

Geometric Correction

Background

In engineering image processing, B-mode images are captured and stored as a two-dimensional array of data elements arranged in Cartesian space. An algorithm is created that can convert OCT B-mode Cartesian image into a geometrically correct sector scan for display.

Steps

1. A spatial mapping of the coordinates of a source Cartesian image (X, Y) to define a new Polar image (R, ϑ) .

$$Distance = \sqrt{x^2 + y^2},\ \vartheta = arctan\frac{Y}{X}$$

2. Interpolation of non-integer coordinates to integer values

(1) Nearest Neighbour Interpolation (Rukundo, 2012)

- Computationally simple
- Select the value of the nearest pixel by rounding the coordinates of the desired interpolation point.

$$dataOut[m,n] = datain[round\left(\frac{m}{2}\right), round\left(\frac{n}{2}\right)]$$

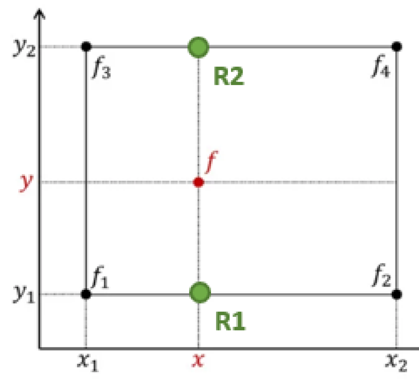
10	4	22
2	18	7
9	14	25

10	10	4	4	22	22
10	10	4	4	22	22
2	2	18	18	7	7
2	2	18	18	7	7
9	9	14	14	25	25
9	9	14	14	25	25

angeljohnsy.blogspot.com

(2) Bilinear Interpolation (Gao, 2011)

A resampling method that uses the nearest four mapped source pixels to estimate a new pixel value.



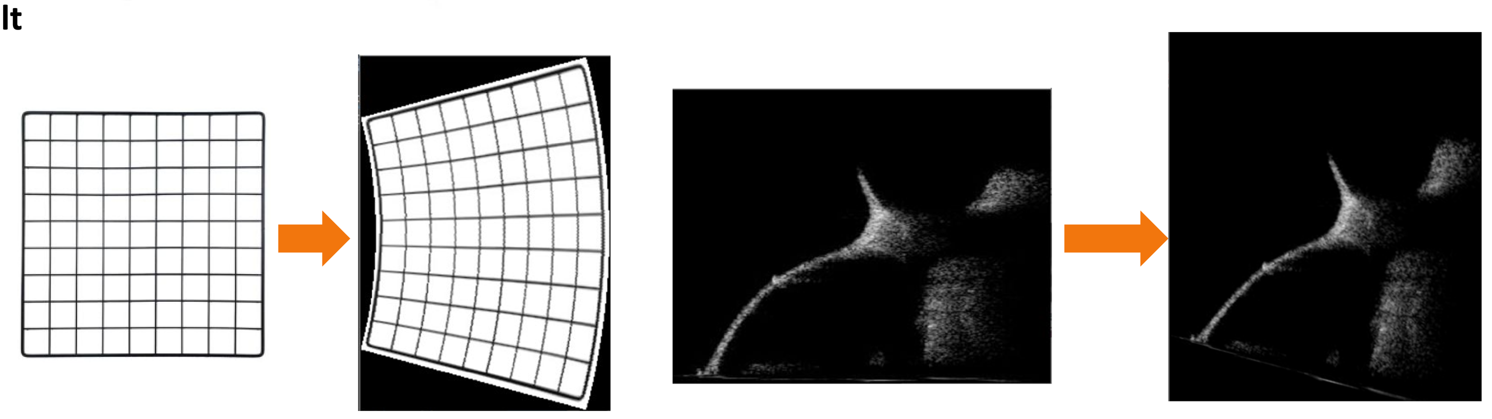
$$func(x,y1) \approx \frac{x2-x}{x2-x1}func(f1) + \frac{x-x1}{x2-x1}func(f2)$$

$$func(x,y2) \approx \frac{x2-x}{x2-x1}func(f3) + \frac{x-x1}{x2-x1}func(f4)$$

$$func(x,y) \approx \frac{y2-y}{y2-y1}func(x,y1) + \frac{y-y1}{y2-y1}func(x,y2)$$

Here, $func(x,y1)$ and $func(x,y2)$ are representing points R1 and R2, respectively.

Result



Automatic Segmentation

Background

Currently, all images obtained from the Ossiview imaging software must be interpreted manually to obtain a dynamic range. Therefore an algorithm was desired that was able to compute accurate dynamic ranges of obtained images automatically.

Algorithm

The algorithm depends on the color values of each pixel on a given line of an image. The algorithm expects a solid structure to be a continuous stream of white pixels, so it is desired to ensure light-gray pixels are properly recorded as white, and dark-gray pixels are properly recorded as black. To do this, a smoothing filter is applied to the image. Fig 1. shows the original image and Fig 2. shows the filtered image.

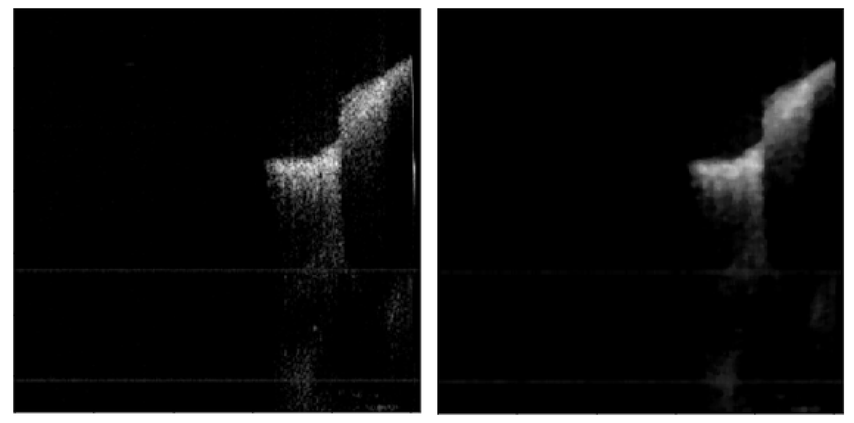


Fig.1 Original Image *Fig.2 Filtered Image*

Once the filter has been applied, the algorithm will search a single line for the first white pixel. Each pixel has a value that determines what the color of the pixel is. When a white pixel is found, the algorithm will continue to count the white pixels until several black pixels are detected. Fig 3. shows an example of the white pixels to be recorded on a single line.

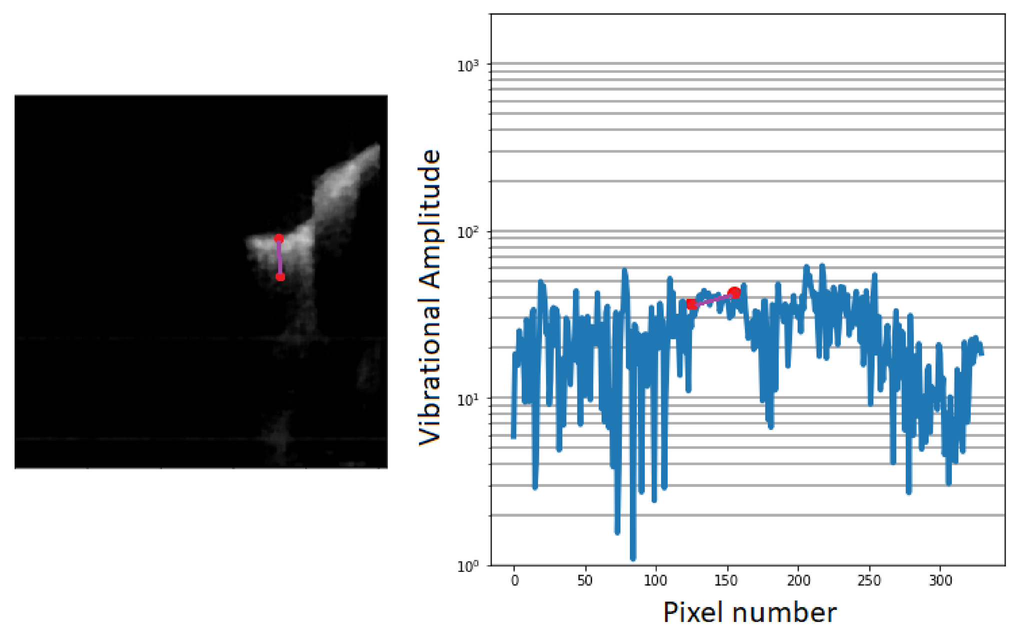


Fig.3 Dynamic Range of a single line

Finally the vibration level across the white pixels is recorded and averaged to determine the Dynamic Range. This value is then made available for the user.

Reference

1. Gao, S. (2011). Bilinear and bicubic interpolation methods for division of focal plane polarimeters. Optics Express, Vol. 19, Issue 27, pp. 26161-26173 .
2. Rukundo, O. (2012). Nearest Neighbor Value Interpolation. International Journal of Advanced Computer Science and Applications, Vol. 3, No. 4.
3. Xu, J. (2017). Complex-based OCT angiography algorithm recovers microvascular information better than amplitude- or phase-based algorithms in phase-stable systems. Physics in Medicine & Biology.
4. Zhang, A. (2015). Methods and algorithms for optical coherence tomography-based angiography: a review and comparison. Journal of Biomedical Optics.
5. J. Farrell, D. MacDougall, R. B A Adamson, "Limited dynamic range artefact removal from Fourier domain optical coherence tomography images", 2018.