g1-Optical Coherence Tomography Angiography (g1OCTA)
Data Processing Guide

Jianbo Tang

Aug 7, 2019

Introduction:

This guide is for post data processing of g1OCTA which outputs 3D vascular structure with flow direction and minimized tail artifacts.

Please cite the following references[1]:

1. Tang, J., Erdener, S. E., Sunil, S. & Boas, D. A. Normalized field autocorrelation function-based optical coherence tomography three-dimensional angiography. *J. Biomed. Opt.* **24**, 036005 (2019).

I. Data acquisition

OCT-based M-mode data acquisition, i.e. repeat Ascan at each X-Y scanning location for a certain period. The data should be saved sequentially as a 1D array (ASCII int16) and named as: RAW-nk-nxRpt-nx-nyRpt-ny-iC, e.g. RAW-1024-100-00400-001-400-1.dat.

II. Input

```
% 1D array spectrum, nK*nXrpt*nX*nY, data format: ASCII int16
        % nK: spectrum pixel (camera elements); nXrpt: Ascan repeat;
        % nX: number of Ascans per Bscan; nY: number of Bscans for the whole volum
        % NOTE: the raw data for the whole volume is usually very large, it's
recommended to process chunk by chunk
    % PRSinfo: processing information
    % PRSinfo.FWHM: Full width at Half Maxim, Amplitude, [transverse, axial], m
    % PRSinfo.fAline: DAQ Aline rate, Hz
    % PRSinfo.Lam: [light source center, wavelength bandwidth], m
    % PRSinfo.Dim: [nz,nx,nyPchk,nTau]
    % PRSinfo.g1_Start: start time for g1 calculation
    % PRSinfo.g1_nt: number of time points for g1 calculation
    % PRSinfo.g1_ntau: number of g1 time lag
    % PRSinfo.intDk: OCT lambda to k space interpolation factor (calibration is
required)
```

Example data:

https://drive.google.com/open?id=168HD4lKt0K97g09zus6H9h7lAyO0jOBZ
https://drive.google.com/open?id=1QvTO_41cPN3_wM9wxCh9NECv_hypVZPC

III. Output

```
 % g1AG, [nz,nx,ny,2]
```

I. CPU calculation-based sub-functions

```
% subFunctions:
    % function [Dim, fNameBase, fIndex]=GetNameInfoRaw(filename0)
    % function DAT= ReadDat_int16(filePath, Dim, iseg, ARpt_extract,RptBscan)
    % function RR = DAT2RR(Dat, intpDk)
    % function GG = RR2g1(RR, PRSinfo)
    % function g1AG=GG2g1AG(GG)
```

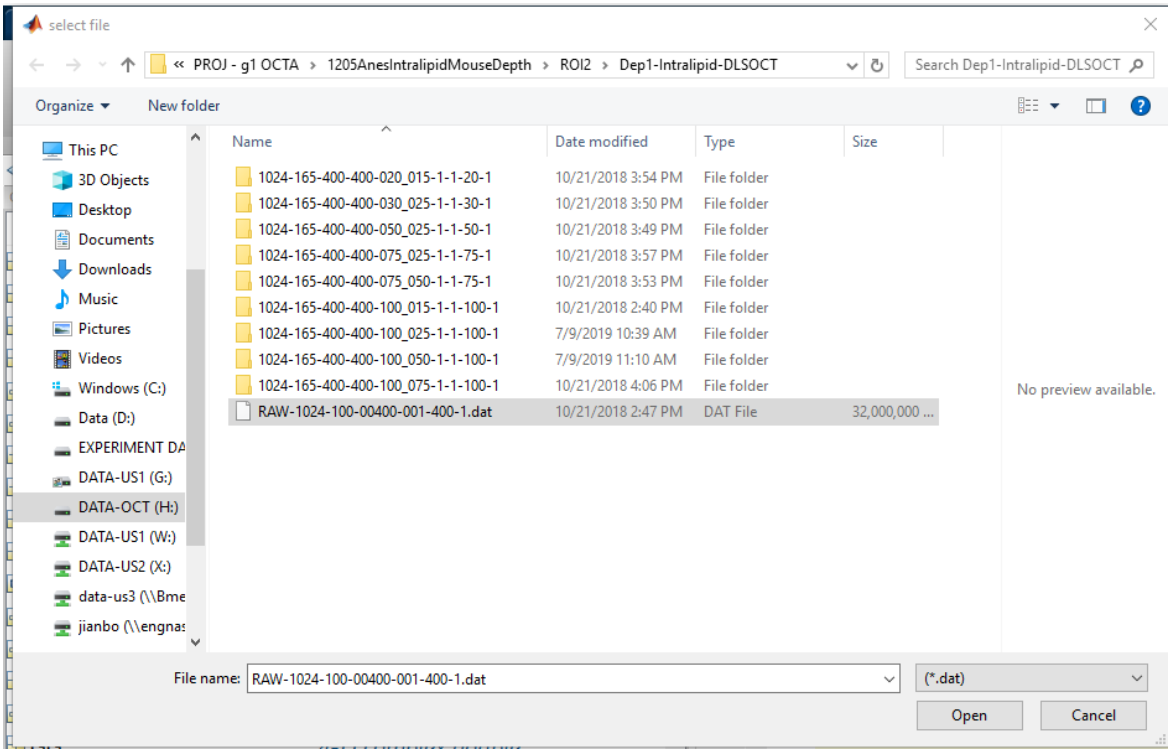
II. GPU calculation-based sub-functions

Note: the minimal GPU memory requirement is 16 GB.

```
% subFunctions:
    % function [Dim, fNameBase, fIndex]=GetNameInfoRaw(filename0)
    % function DAT= ReadDat_int16(filePath, Dim, iseg, ARpt_extract,RptBscan)
    % function RR = DAT2RR_GPU(Dat, intpDk)
    % function GG = RR2g1_GPU(RR, PRSinfo)
    % function g1AG=GG2g1AG_GPU(GG)
```

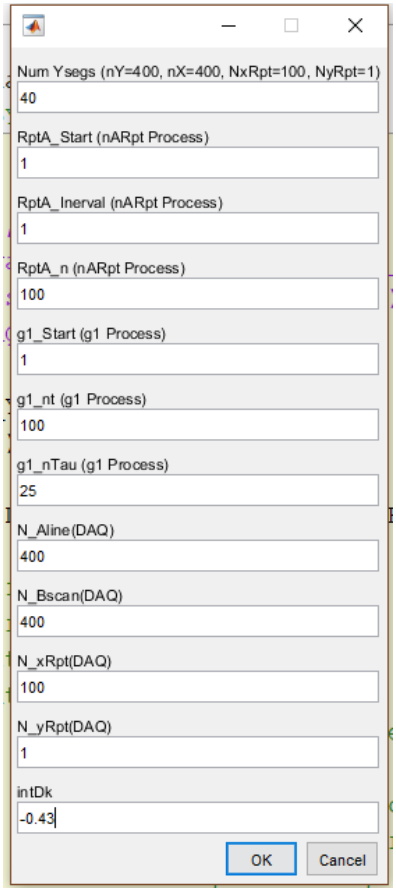
III. Main_g1OCTA data processing

III.1 select file

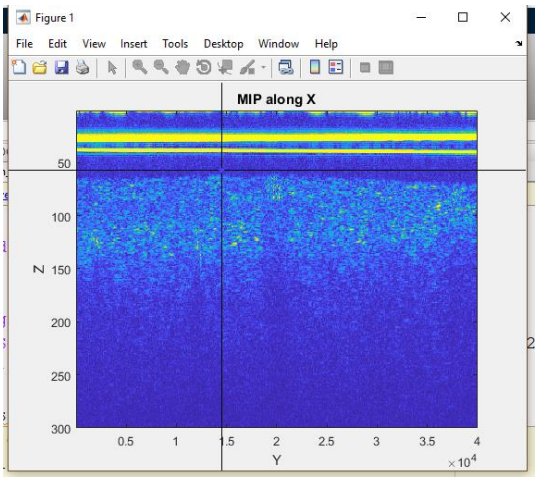


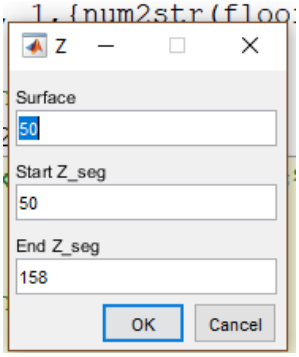
III.2 data processing parameter

Specify the number of chunks for data processing (split large data size into small chunks), and the g1 calculation parameters (nTau and nt). Double check the intDk.

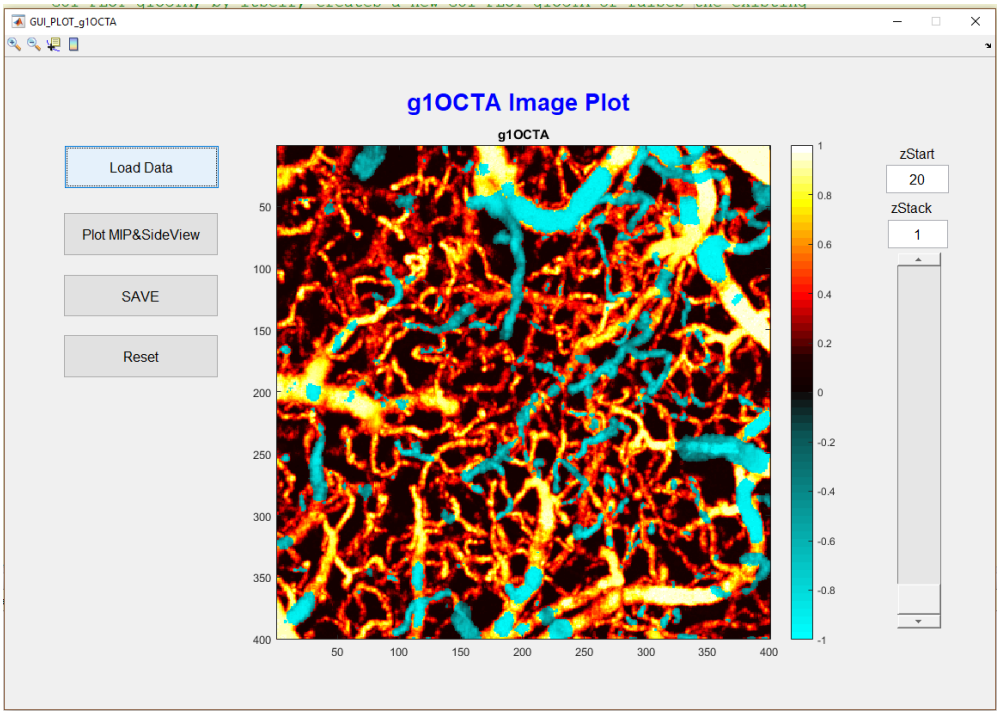


III.3 select the axial data processing range [surface, zStart, zEnd]

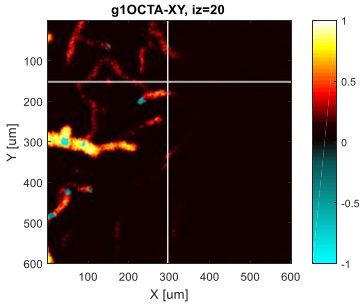


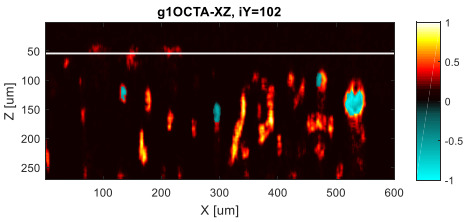


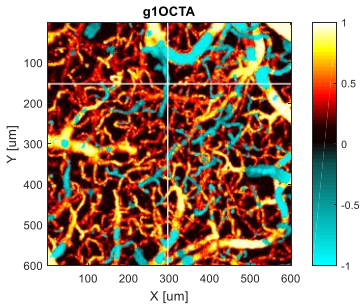
III.4 g1OCTA result plot (GUI_PLOT_DLSOCT)

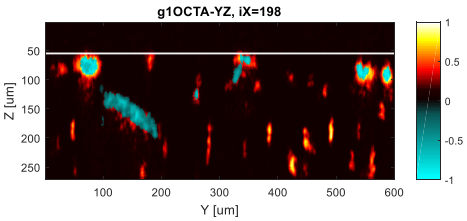


1. Load the saved g1OCTA data
2. Use the slider or zStart+zStack to check single or MIP (maxim intensity projection) en face plane. Cyan color means descending flow
3. Use ‘Plot MIP&SideView’ to plot a MIP for certain depth range (set SideView(N:0; Y:1) to 0). Or plot XY, YZ, XZ, and MIP figures by set SideView(N:0; Y:1) to 1.


g1OCTA-XY, iz=20
Y [um]
X [um]
100
200
300
400
500
600
1
0.5
0
-0.5
-1


g1OCTA-XZ, iY=102
Z [um]
X [um]
50
100
150
200
250
100
200
300
400
500
600
1
0.5
0
-0.5
-1


g1OCTA
Y [um]
X [um]
100
200
300
400
500
600
1
0.5
0
-0.5
-1


g1OCTA-YZ, iX=198
Z [um]
Y [um]
50
100
150
200
250
100
200
300
400
500
600
1
0.5
0
-0.5
-1