

OPT_recon

Introduction

The OPT_recon software is designed and developed to allow the reconstruction of OPT images through the various stages of reconstruction either in one shot as one program or by using the different stages of OPT reconstruction code as separate entities for the reconstruction of images.

The following explains each different step in further detail,

Image formats

"minc2" is the gold standard of image formats at the MICE Imaging Center. Other image formats can be used as long as you are willing to figure out the right converter to minc2 format.

Currently the OPT scanner produces a series of .bin images (note, these are 12 bit images which are passed to 16 bit) plus a .log file which includes the scanner specification. Also, there are total of 10 dark images supplied with the bin files which must be averaged and subtracted from the bin file before creating the 3D minc file.

Creating OPT images in MINC(2) format

The first step prior to reconstruction of the OPT data is creating a 3D minc(2) format image which will be used by all the different stages of the opt reconstruction code. Currently the OPT scanner creates a series of 2D images in (.bin) format along with one (.log) file which includes some information about the data including the image size, step, zoom, etc. Also, there are 10 dark background files which must be averaged and subtracted from the bin files before they are used, however, since background subtraction is already accounted for in the next section of OPT_recon (ccd correction), this part is omitted. However, there is code to subtract these images from the raw images, to activate this code and subtract the average background image from each raw file before creating a minc file simply use "-s" option.

This section of the opt reconstruction code "optdataTOminc" can be called as a stand alone program without any options in which case the program assumes the following:

1. directory that contains all .bin files and the .log file (current "." in linux)
2. creates a 3D image called "projections.mnc" which will be saved in the current directory(1)

further, "optdataTOminc" also allows the user to provide the above (#1 and 2) using flags, **(-f)** --> **filename.mnc** and **(-p)** --> **/micehome/mydirectory**

and if the user wishes to subtract the dark images, **(-s) flag should be used to average all dark images and subtract the average image from each raw file.**

```
optdataTOminc -f filename.mnc -p /micehome/mydirectory
```

which results in creating a 3D image "/micehome/mydirectory/filename_projections.mnc" using all .bin files (which were subtracted from the average dark images) and .log file located in /micehome/mydirectory. NOTE: filename.mnc --> filename_projections.mnc

This 3D image created above records the local time and the name and location of all files used at time of creation. Check the "history" minc variable using

```
mincheader filename_projections.mnc
```

CCD Detection

The first stage of the reconstruction code attempts to calibrate the photon response of the CCD (variation of pixel sensitivity) in order to determine the extent of the imperfections. Calibration of the CCD using a second order polynomial prevents the streak artifact.

In order to create the coefficients for the second order polynomial, a set of Dark and Bright images at different exposure times of 50 ms to 1550 ms were acquired. Using the Linear Model from R(refer to R tutorial), three sets of coefficients, intercept, exposure_time and exposure_time2 were created.

To account for illumination which is not accounted for in OPT, the three coefficients were modified as follows,

assuming $B_0 = \text{intercept}$, $B_1 f_i = \text{exposure_time}$ and $B_2 f_i^2 = \text{exposure_time}^2$

$B_0_hat = \text{intercept}$,

$N_i = \text{norm}(\text{smooth}(B_1 f_i))$ (normalizing the values so the maximum is set to 1). The spline smoothing (spline_smooth with distance of 25 mm was used)

$B_1_hat = B_1 f_i / N_i$

$B_2_hat = B_2 f_i^2 / (N_i)^2$

Using the above three coefficients the polynomial equation can be solved as follows,

B2_hat = "coefa"

B1_hat = "coefb"

B0_hat = "coefc"

plugging the above to solve for x in the polynomial equation using

B2_hat = a

B1_hat = b

B0_hat - acquired Image = c

$$x = (-b + \sqrt{b^2 - 4ac}) / 2a$$

NOTE: The OPT_recon software currently uses the above three coefficients (located in share/OPT_recon) automatically, options are provided in the code should you require to use different ones. This is specially handy if you are using the opt code with a different CCD other than the one currently used at MICE (2012).

This section of the OPT_recon code is written in python hence using you must use python to run the software,

```
opt_applyccdcoeffs
```

prints a short message stating how the program must be used

```
opt_applyccdcoeffs -help
```

prints the help message along with all the parameters which are available for use with this software.

Finally to use the CCD calibration software with the **existing coefficients** any of the commands below can be executed, note: both commands below result in creating the output named outputfile.mnc in directory /micehome/mydirectory using the coefficients created using the above description.

```
opt_applyccdcoeffs inputfile.mnc outputfile.mnc -p /micehome/mydirectory
opt_applyccdcoeffs inputfile.mnc /micehome/mydirectory/outputfile.mnc
```

Fluctuations in Illumination and photo-bleaching

Power fluctuations of mercury lamp results in fluctuations in OPT views since fluorescent output is proportional to the intensity of incident light. Photo bleaching is the fluorescent signal intensity decay with respect to time due destruction of fluorphorse.

Correcting for both fluctuations and photo bleaching as follows,

1. fit the data into a spline --> $f(t) = \text{spline} \cdot x(t)$
2. fit the spline fitted data in an exponential --> $g(t) = A e^{-Bt}$
3. $f(t)/x(t) \cdot A/g(t)$

Processing the above creates one correction factor per projection of OPT image which are then multiplied by each image accordingly.

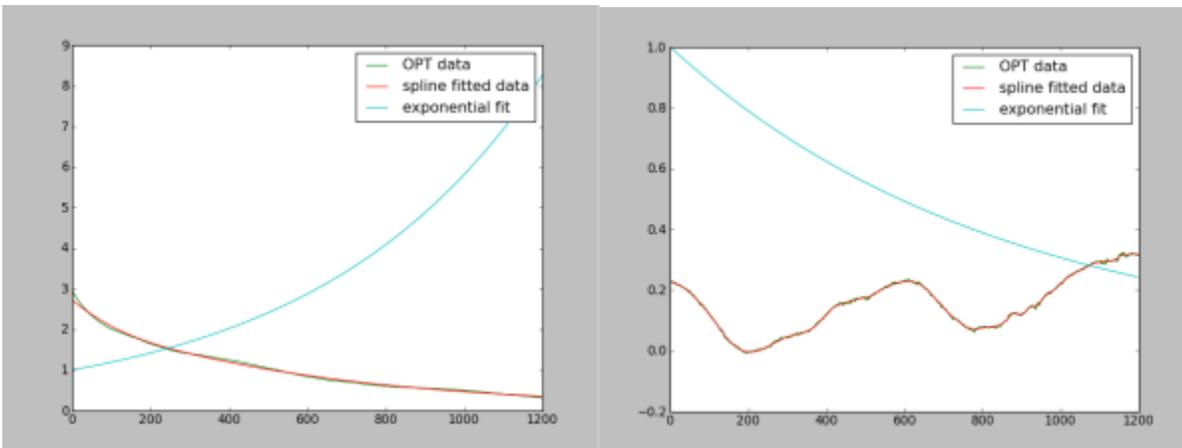
important note: The above corrects for signal decay on the assumption that the raw OPT data has a decaying shape (left figure). However, in the case of embryo imaging, the raw data sometimes has a more variant shape (right figure), correcting with exponential in this case will not be recommended and OPT_recon will choose the spline correction or normalization depending on the options specified at run time.

method 1: spline correction only

method 2: spline and exponential correction

method 3: normalized all projections to one mean

method 4: none of the above (no correction)



This section of the OPT_recon code is written in python hence using you must use python to run the software

```
opt_signaldecay
```

prints a short message stating how the program must be used.

```
opt_signaldecay -help
```

prints the help message along with all the parameters which are available for use with this software.

Finally to use the signal decay software with some of the available options the command below can be used. This will determine a smooth spline from the which will then be fitted to an exponential. The software can also show the plot of the data versus the fitted using the **(-s)** option and further save the plot using **(-f)** filename in "png" format.

```
opt_signaldecay inputfile.mnc outputfile.mnc -p /micehome/mydirectory -f plot_filename -s
```

Point Spread Function

OPT images suffer from blurring that gets worse as the distance from rotational axis increases. This is due to collection of images with varying degrees of defocus. The image of a point object positioned at any radius from the rotational axis is in focus for half the revolution and out of focus for the other half. The out of focus data is included in the 3D back projection reconstruction and contributes to lack of focus in 3D images. We proceed with "filtering the frequency space" information of 3D OPT using the frequency distance relationship (FDR) filter to remove the out of focus data and narrow the point spread function of the in focus data. Please note, to fully understand the following description you must be familiar with terms such as DFT (Discrete Fourier Transform), IFT (Inverse Fourier transform), Complex images (i.e. two channels one for real and one for imaginary part of each voxel).

The complete Frequency space filter or FDR filter is calculated as follows (this is based on the paper Resolution improvement in emission OPT published in 2007),

$$H^{-1}_{final} = H^{-1}_{lim} * W_r * W_w * W_b$$

H^{-1}_{lim} : max limit recovery filter

W_r : roll off filter

W_w : band limit filter

W_b : wiener filter

The following is the description of how each of the above is calculated in detail: (for detailed formulas regarding each filter refer to the paper cited above)

1. max limit recovery filter --> (to avoid the overemphasis of noise, the vectors in FDR recovery filter are scaled by a weighing factor paper formula 7)
 - a) calculate the psf in FDR space (i.e. psf is transformed to FDR space)
 - b) calculate deconvolution filter in FDR space
 - c) limit recovery from deconvolution filter

2. The roll off filter de-emphasizes the out of focus data along the lines of decreasing slope and is calculated based on equation 9 paper
3. The band limit filter to remove the highest frequencies in the FT of the lens which contain no information paper formula 11
4. The wiener filter to de-emphasize the frequencies which are mostly noise

As every other section of the OPT_recon software, the directory which contains all the above filter is called "PSF". The following is a brief description of how each section of the code in this directory is designed and implemented to produce the above filter.

**To calculate the FFT and Inverse FFT of all images the "fftw3" libraries are used. The 3D FFT of the data is calculated by calculating the 2D FFT of the data in one direction and the 1D FFT in another direction. The FFT of PSF is different than above because its meant to be calculated as 2D FFT only (refer to paper), so in this case a separate code is written to calculate the 2D FFT of every slice of the PSF image while shifting the four quadrant of each slice (to make sure the center of FFT is in the center).

The following is the piece by piece description of the software which produces the FDR filter and applies it to the OPT image prior to back projection reconstruction.

1. max limit recovery filter (complex value image), can be calculated using the following executables,
 - a) opt_psflnFdrFilter
 - b) opt_deconvlnFdrFilter
 - c) opt_limit_recoveryFilter

***Note the result of each one of the above will be used as the input to the next section.

To create a user friendly code, all of the above is packed into one executable

```
opt_frequencySpaceFilterRecon_fdr  projectionsfile.mnc psffile.mnc  max_limit_recovery.mnc  -n noise_reduction
-r fdrlimit -d distance_to_psf
```

2. roll off filter can be calculated using the executable,


```
opt_rollofffilter
```
3. band limit filter can be calculated using the executable,


```
opt_bandlimitfilter
```

Note that the wieners filter is implemented as part of the deconvolution filter so no separate code is provided.

```
opt_frequencySpaceFilterRecon_projections  projectionsfile.mnc projections_rolloff_bandlimit.mnc -w weight -m
maxslope -b bandlimit
```

The final task is to

- a) complex multiply the result of the above two section, which are the max limit recovery filter and the combination of roll off and band limit filter
- b) calculate the inverse FFT of result from section a
- c) calculate the shift of the inverse FFT from section b
- d) calculate the magnitude of the inverted shifted result from section c

```
opt_frequencySpaceFilterRecon_complete  projections_rolloff_bandlimit.mnc  max_limit_recovery.mnc
psf_corrected.mnc
```

The above three components are designed and implemented in this manner due to the complexity of the PSF correction and to make it simpler to execute. However, every single section of the PSF correction code is also implemented as a stand alone executable so the user can run them one by one observe the results and move on to the next section. There are total of 17 executables which also reside on the network along side the above three components.

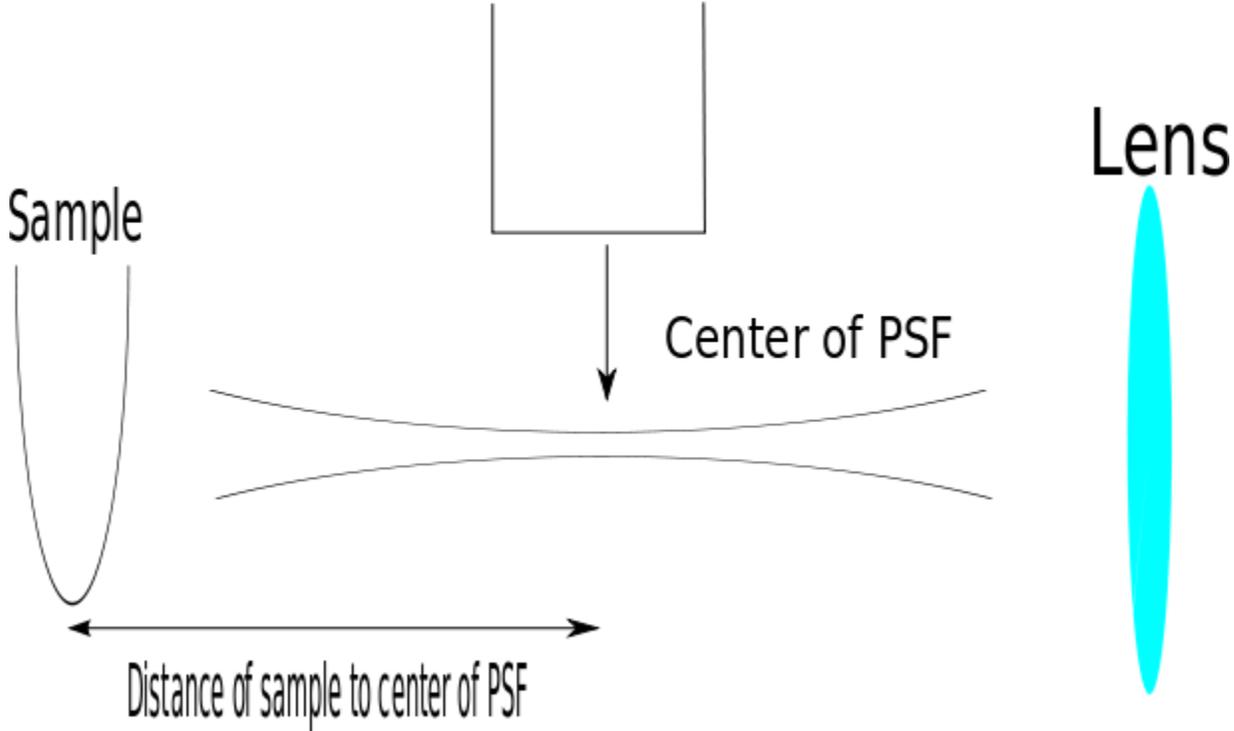
Finally, to place all the PSF correction code(and believe me there is alot of them) a simple python script places all the above in one call,

```
opt_frequencySpaceFilterRecon projectionsfile.mnc psffile.mnc final_psf_corrected_file.mnc
```

Note that all options available to each single executable program (PSF related), is also avaiable from this script i.e, distance of sample to center of PSF (-d), check below for more information.

Distance of sample to center of PSF

It is important to note that the center of the PSF of lens is very important for the calculation of the above and is already implemented in the code, however, the other measurement is the distance of the sample from the center of the PSF which must be entered by the user since it can only be determined at scan time, this is further explained in the following figure.



Filtered Back Projection

Computed tomography is a technique for estimating the interior of an object from measurements of radiation collected around the object. This radiation can be either projected through or emitted from the object. **CTSim** (<http://ctsim.org>, designed and developed by Kevin Rosenberg) simulates the process of projecting light through a phantom object which can then reconstruct the interior of the object from those projections is the software used for the filtered back projection section of OPT_recon. This software is written in C++ and requires knowledge of C++ and Make files prior to use.

The back projection software is modified to include the calculation of the offset of rotational axis from the center of detector array. First, we calculate a coarse offset of the rotational axis using 2D normalized cross-correlation of the provided slices of the image which result in correlation coefficients on a subset of the data, which can range in value from -1.0 to 1.0.

Cmake files are then implemented to only build the back projection section of the software as part of the OPT_recon software in on step. This modification is part of the MICE version of the software and is "not" included in the source code available for download from CTSim website.

To simplify the use of CTSim, a python wrapper using (python-boost libraries) is implemented. This would allow the easy access of data both to and from CTSim.

The back projection reconstruction requires a projection file to be created from each individual slice of OPT data, this projection is then reconstructed into data. If the offset correction calculation is specified by the user, the code will take a sample of the data slices and reconstruct them to calculate the offset correction based on above which will be applied to every slice after reconstruction. The user can also specify a reconstruction size other than the one taken from the image (x,y dimensions) using the "resize" option. However, the reconstruction size must not be smaller than the original image size or correct results will not be guaranteed.

The important note about the back projection of OPT data is that the axis presenting the sinograms must be submitted to back projection code in correct orientation. To make this clear, the image on the left panel (**original from scanner**) shows the sinogram on the X axis hence the ZY slices must be submitted to ctsim, however, the image on the right panel shows the sinogram on the Y axis hence ZX slices should be reconstructed. The back projection code assumes the orientation of the original image (i.e., ZY slices will be reconstructed), if you are unsure about the orientation of the image, check it out with "Display". Use the "flip" option if your image has the old orientation (hence ZX slices will be reconstructed).

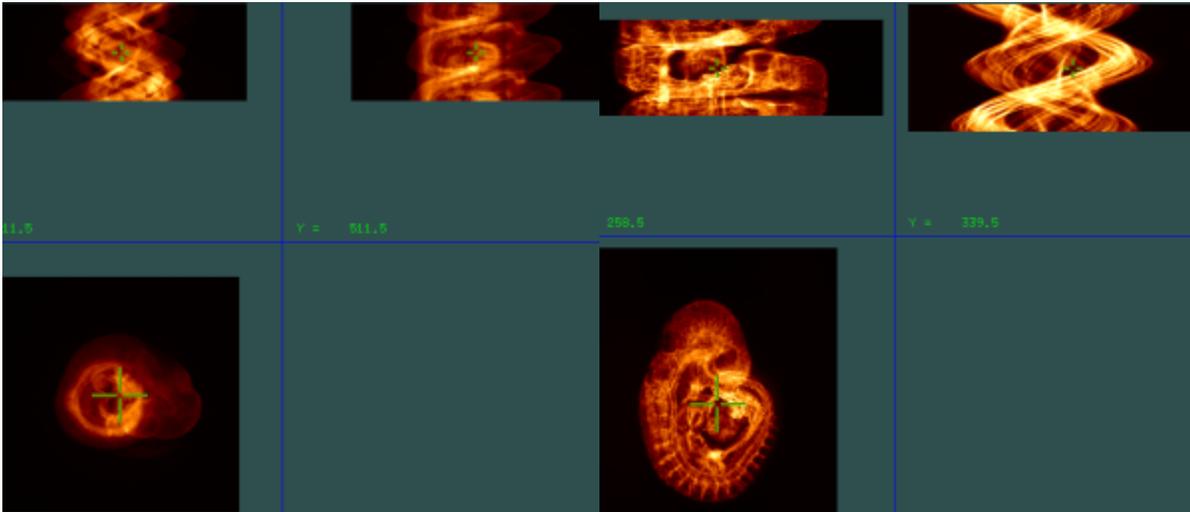


Image dimensions after the reconstruction is completed will be different than the original image. For instance, images from the current OPT scanner (ZY reconstructed),

original	reconstructed(default reconsize)	reconstructed(reconsize=P,Q)
Z=A	Z=C	Z=C
Y=B	Y=A	Y=Q
X=C	X=B	X=P

To get a printout of all options available to the program

```
opt_backproject -help
```

To reconstruct a dataset with offset correction, to use automatic offset correction with ONE initial guess use "auto". For a more accurate offset correction use "twostep" which uses two initial offset guesses instead of one and "MAY" provide a better correction value. Also you can specify the reconstruction size using the the reconsize flag, using sizes X and Y.

```
opt_backproject infile.mnc outfile.mnc --reconsize=X,Y --offset=auto
```

or

```
opt_backproject infile.mnc outfile.mnc --reconsize=X,Y --offset=twostep
```

***Important Note, all python modules written for OPT_recon can be executed as a python script (above command) or imported as a python module from within python (below)

```
python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:57:41)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import opt_backproject
>>> opt_backproject.process_options(infile='/micehome/mydirectory/filein.mnc',outfile='/micehome/mydirectory/outfile.mnc', offset='auto',reconsize='X,Y')
```

***Note: The back projection code does take sometime to go through all slices of image specially if your image is large. For instance, it takes roughly 2-3 hours to reconstruct an image of size 400x1024x1024, on one of the back room machines.

Automatic Cropping

To reduce the image size by at least half, a simple cropping algorithm can be applied to the reconstructed volume. This program uses the multiprocessing module of python to process 10 slices of the image at a time. Its designed to determine if the sum of the values is above a certain threshold in all directions, which means, it figures out minimum and maximum values for all three dimensions and then uses "mincreshape" to crop the data using these values. Currently the program uses threshold intensity value of 200, to use a different threshold value use the "-t" flag and provide the new value.

```
opt_crop_volume inputfile.mnc outputfile.mnc
```

Master OPT Reconstruction

To make the OPT_recon user friendly and to place all the above in a simple concept, a master opt recon is created. This is a simple python script that calls all the above " in the order which they appear" and create one final result image as well as all the in between resulting images. All that is needed is the location of all the .bin files and optionally a filename. If you choose to omit the filename "projections.mnc" will be created.

so here is brief description of what goes on underneath

1. Create a 3D mnc file (given name or "projections.mnc") from all .bin files, read the log file that is supplied with the bin files into "opt_vars".
2. Apply the CCD correction to the above image (#1) and save it as projections_linearity_corrected.mnc or filename_linearity_corrected.mnc
3. Apply the correction for fluctuations of illumination and photo-bleaching to the above image (#2) and save it as projections_linearity_decay_corrected.mnc or filename_linearity_decay_corrected.mnc
4. Create the FDR filter using the PSF image provided and apply the correction to the image and save it as projections_linearity_decay_fdr_corrected.mnc or filename_linearity_decay_fdr_corrected.mnc
5. Back projection reconstruction of the above image (#2) with default reconvolve and automatic offset correction and save it as volume.mnc or filename_volume.mnc
6. Crop the reconstructed volume to reduce its size by at least half

*** If you are unsure about what happened to any of the above images , just do "mincheader" and check the history of the file, all the resulting images, have a history which states what happened to them when.

```
masterOPT filename.mnc psffile.mnc -p /micehome/mydirectory
```

If you want to read all raw files from one directory and save all the resulting images in another directory, you can specify that with the filename as follows

```
masterOPT /micehome/directory/tosave/allimages/filename.mnc psffile.mnc -p /micehome/directory/toread/all  
/rawfiles
```

Finally, running the above requires large memory and time to complete so run it on the farms or on a machine that can handle your data.

******* keep in mind that if you submit a job to the farms which requires more memory than what you have requested, your job will get killed with NO WARNING and this has NOTHING to do with the OPT_recon software so when submitting a job with any of the OPT_recon software sections or masterOPT make sure you ask for ENOUGH MEMORY.**

Here is an example

```
masterOPT -p /projects/souris/baghdadi/For_Leila/ g.mnc  
-rw-r--r-- 1 baghdadi mice 838877352 2012-04-24 10:02 g_projections.mnc ---> under 1 minute to create 3D minc projections  
-rw-r--r-- 1 baghdadi mice 1677735344 2012-04-24 10:06 g_projections_linearity_corrected.mnc ---> 4 minutes to apply CCD correction  
-rw-r--r-- 1 baghdadi mice 1677735520 2012-04-24 10:09 g_projections_linearity_decay_corrected.mnc --> 3 minutes to correct for signal decay  
  
-rw-r--r-- 1 baghdadi mice 1677735520 2012-04-24 10:09 g_projections_linearity_decay_fdr_corrected.mnc --> 30 minutes to correct for signal decay  
-rw-r--r-- 1 baghdadi mice 4294980544 2012-04-24 11:47 g_volume.mnc ---> just under 2 hours to reconstruct the projections  
-rw-r--r-- 1 baghdadi mice 2684377720 2012-04-24 11:54 g_volume_cropped.mnc --> 7 minutes to figure out where to crop the data and crop it
```

*** NOTE, size change