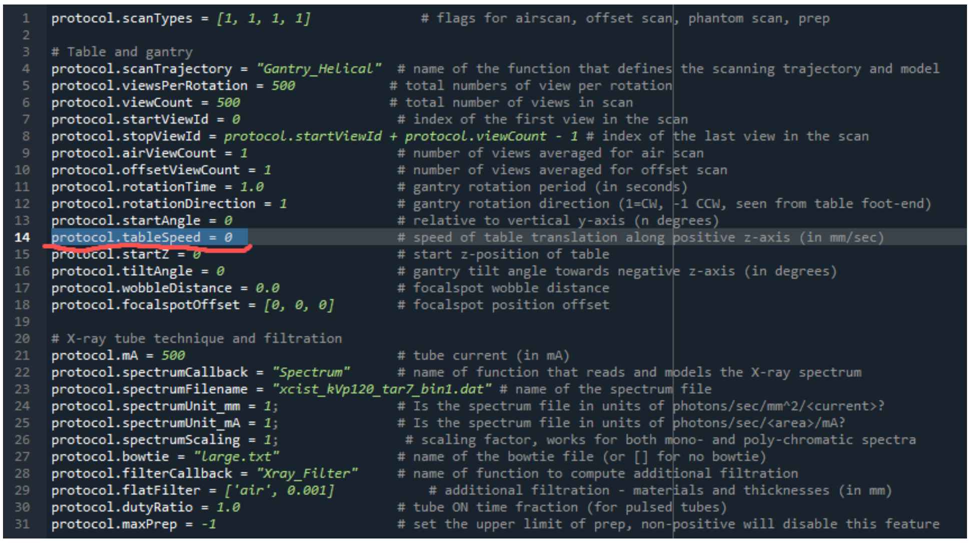
Short Scan Using XCIST

Botao Zhao

October 10, 2025

1 Introduction

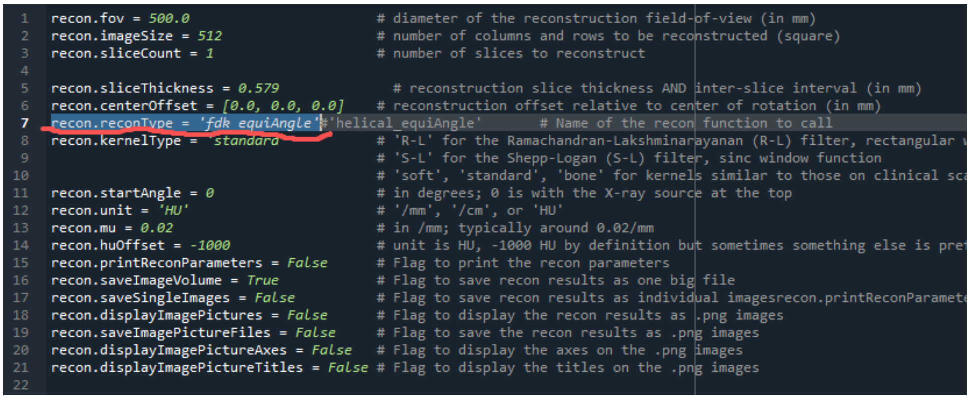
This document explains how to perform short-scan reconstruction using XCIST. XCIST supports two reconstruction modes: `fdk_equiAngle` and `helical_equiAngle`. `fdk_equiAngle` is used for axial scans, where the patient table remains stationary (table speed $= 0$). `helical_equiAngle` is used for helical scans, where the patient table moves continuously (table speed $\neq 0$). The table speed can be configured in the `protocol.cfg` file, while the reconstruction type is specified in the `recon.cfg` file.



```
protocol.scanTypes = [1, 1, 1, 1]             # flags for airscan, offset scan, phantom scan, prep

# Table and gantry
protocol.scanTrajectory = "Gantry_Helical"  # name of the function that defines the scanning trajectory and model
protocol.viewsPerRotation = 500            # total numbers of view per rotation
protocol.viewCount = 500                   # total number of views in scan
protocol.startViewId = 0                    # index of the first view in the scan
protocol.stopViewId = protocol.startViewId + protocol.viewCount - 1 # index of the last view in the scan
protocol.airViewCount = 1                   # number of views averaged for air scan
protocol.offsetViewCount = 1                # number of views averaged for offset scan
protocol.rotationTime = 1.0                 # gantry rotation period (in seconds)
protocol.rotationDirection = 1              # gantry rotation direction (1=CW, -1 CCW, seen from table foot-end)
protocol.startAngle = 0                     # relative to vertical y-axis (n degrees)
protocol.tableSpeed = 0                     # speed of table translation along positive z-axis (in mm/sec)
protocol.startZ = 0                         # start z-position of table
protocol.tiltAngle = 0                      # gantry tilt angle towards negative z-axis (in degrees)
protocol.wobbleDistance = 0.0               # focalspot wobble distance
protocol.focalspotOffset = [0, 0, 0]        # focalspot position offset

# X-ray tube technique and filtration
protocol.mA = 500                           # tube current (in mA)
protocol.spectrumCallback = "Spectrum"      # name of function that reads and models the X-ray spectrum
protocol.spectrumFilename = "xcist_kVp120_tar7_bin1.dat" # name of the spectrum file
protocol.spectrumUnit_mm = 1;               # Is the spectrum file in units of photons/sec/mm^2/<current>?
protocol.spectrumUnit_mA = 1;               # Is the spectrum file in units of photons/sec/<area>/mA?
protocol.spectrumScaling = 1;                # scaling factor, works for both mono- and poly-chromatic spectra
protocol.bowtie = "large.txt"               # name of the bowtie file (or [] for no bowtie)
protocol.filterCallback = "Xray_Filter"     # name of function to compute additional filtration
protocol.flatFilter = ['air', 0.001]            # additional filtration - materials and thicknesses (in mm)
protocol.dutyRatio = 1.0                    # tube ON time fraction (for pulsed tubes)
protocol.maxPrep = -1                       # set the upper limit of prep, non-positive will disable this feature
```

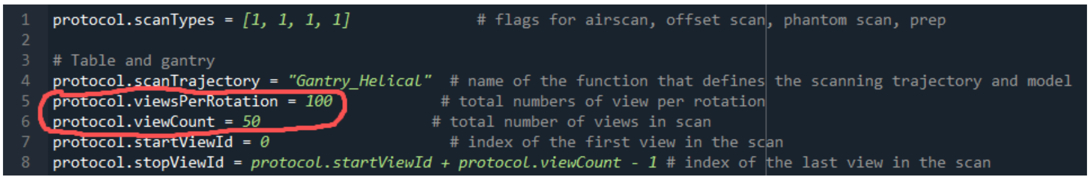


```
recon.fov = 500.0                        # diameter of the reconstruction field-of-view (in mm)
recon.imageSize = 512                    # number of columns and rows to be reconstructed (square)
recon.sliceCount = 1                     # number of slices to reconstruct

recon.sliceThickness = 0.579               # reconstruction slice thickness AND inter-slice interval (in mm)
recon.centerOffset = [0.0, 0.0, 0.0]     # reconstruction offset relative to center of rotation (in mm)
recon.reconType = 'fdk_equiAngle'#'helical_equiAngle'       # Name of the recon function to call
recon.kernelType = 'standard'            # 'R-L' for the Ramachandran-Lakshminarayanan (R-L) filter, rectangular
                                         # 'S-L' for the Shepp-Logan (S-L) filter, sinc window function
                                         # 'soft', 'standard', 'bone' for kernels similar to those on clinical sca
recon.startAngle = 0                     # in degrees; 0 is with the X-ray source at the top
recon.unit = 'HU'                        # '/mm', '/cm', or 'HU'
recon.mu = 0.02                          # in /mm; typically around 0.02/mm
recon.huOffset = -1000                   # unit is HU, -1000 HU by definition but sometimes something else is pre
recon.printReconParameters = False       # Flag to print the recon parameters
recon.saveImageVolume = True             # Flag to save recon results as one big file
recon.saveSingleImages = False           # Flag to save recon results as individual imagesrecon.printReconParamete
recon.displayImagePictures = False       # Flag to display the recon results as .png images
recon.saveImagePictureFiles = False      # Flag to save the recon results as .png images
recon.displayImagePictureAxes = False    # Flag to display the axes on the .png images
recon.displayImagePictureTitles = False # Flag to display the titles on the .png images
```

2 Running Short Scan

The easiest way to do short scan is setting the value of `protocol.viewCount` to be less than `protocol.viewsPerRotation`, in `protocol.cfg` file. For example, if you set `protocol.viewsPerRotation = 100` and `protocol.viewCount = 50`, that's equivalent to "half scan".



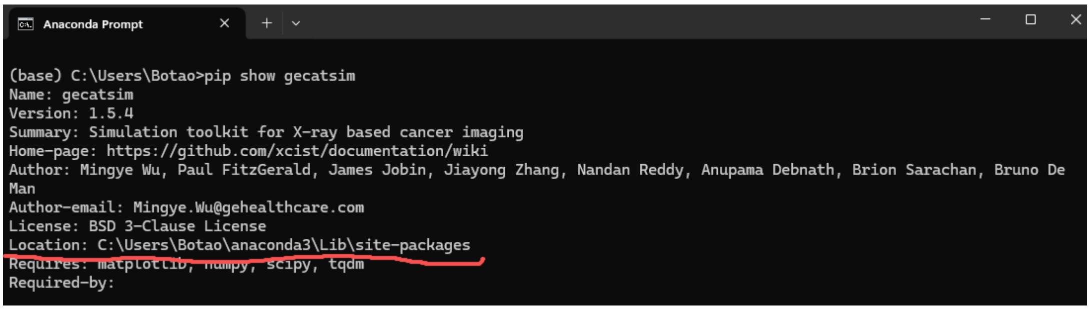
```
protocol.scanTypes = [1, 1, 1, 1]              # flags for airscan, offset scan, phantom scan, prep

# Table and gantry
protocol.scanTrajectory = "Gantry_Helical"  # name of the function that defines the scanning trajectory and model
protocol.viewsPerRotation = 100            # total numbers of view per rotation
protocol.viewCount = 50                   # total number of views in scan
protocol.startViewId = 0                    # index of the first view in the scan
protocol.stopViewId = protocol.startViewId + protocol.viewCount - 1 # index of the last view in the scan
```

2.1 Running Short Scan - Axial Scanning

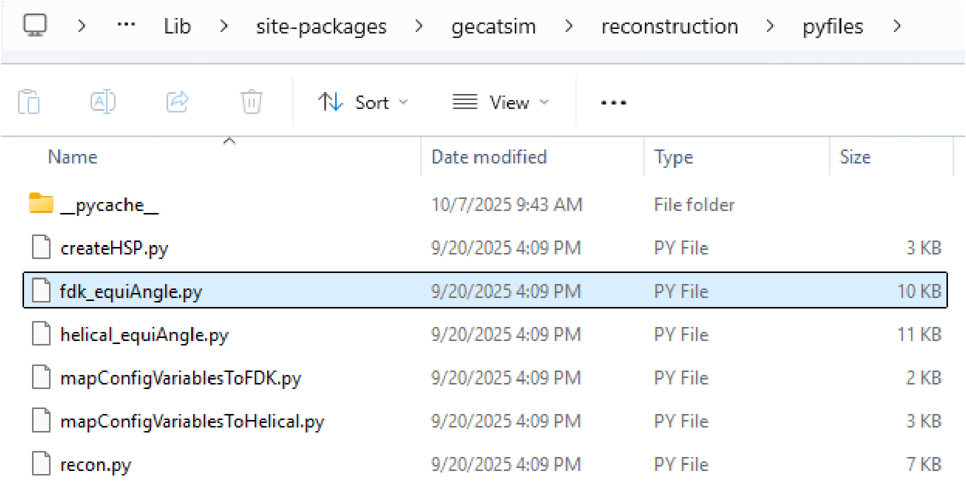
To run short scan with axial scanning mode, you need to set `recon.reconType = 'fdk_equiAngle'` in `recon.cfg`. However, the default code doesn't allow you to do short scan in axial scan mode (It will gives you index error). So we have to modify the source code `fdk_equiAngle.py` in the XCIST package.

Run `pip show gecatsim` in Anaconda Prompt:



```
(base) C:\Users\Botao>pip show gecatsim
Name: gecatsim
Version: 1.5.4
Summary: Simulation toolkit for X-ray based cancer imaging
Home-page: https://github.com/xcist/documentation/wiki
Author: Mingye Wu, Paul FitzGerald, James Jobin, Jiayong Zhang, Nandan Reddy, Anupama Debnath, Brion Sarachan, Bruno De
Man
Author-email: Mingye.Wu@gehealthcare.com
License: BSD 3-Clause License
Location: C:\Users\Botao\anaconda3\Lib\site-packages
Requires: matplotlib, numpy, scipy, tqdm
Required-by:
```

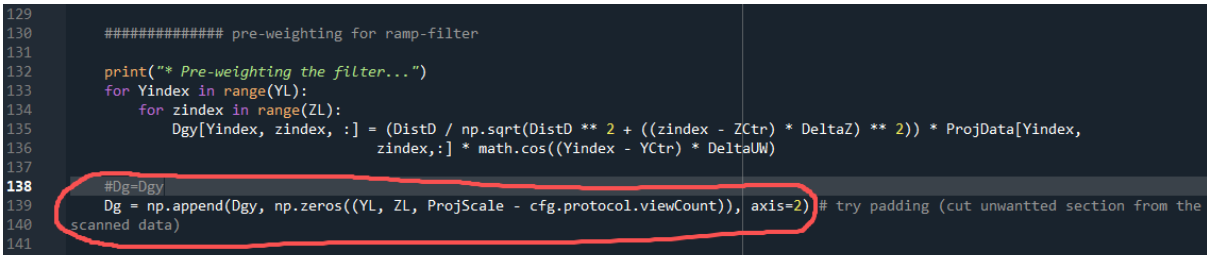
This should give you the location of where the package is installed at. Then go to `gecatsim\reconstruction\pyfiles`. Open `fdk_equiAngle.py`.



··· › Lib › site-packages › gecatsim › reconstruction › pyfiles ›
Sort
View
···

Name	Date modified	Type	Size
__pycache__	10/7/2025 9:43 AM	File folder	
createHSP.py	9/20/2025 4:09 PM	PY File	3 KB
fdk_equiAngle.py	9/20/2025 4:09 PM	PY File	10 KB
helical_equiAngle.py	9/20/2025 4:09 PM	PY File	11 KB
mapConfigVariablesToFDK.py	9/20/2025 4:09 PM	PY File	2 KB
mapConfigVariablesToHelical.py	9/20/2025 4:09 PM	PY File	3 KB
recon.py	9/20/2025 4:09 PM	PY File	7 KB

At line 138, replace `Dg = Dgy` with `Dg = np.append(Dgy, np.zeros((YL, ZL, ProjScale - cfg.protocol.viewCount)), axis=2)`:



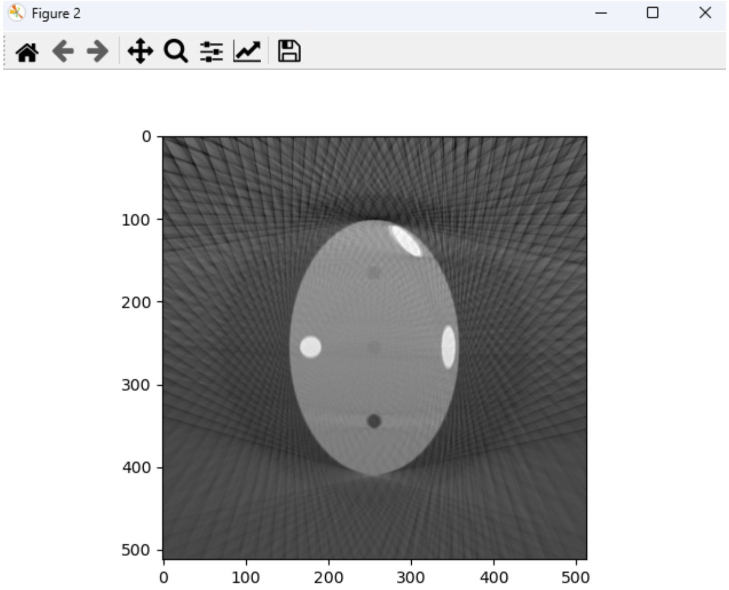
```
    ############## pre-weighting for ramp-filter

    print("* Pre-weighting the filter...")
    for Yindex in range(YL):
        for zindex in range(ZL):
            Dgy[Yindex, zindex, :] = (DistD / np.sqrt(DistD ** 2 + ((zindex - ZCtr) * DeltaZ) ** 2)) * ProjData[Yindex,
                                    zindex,:] * math.cos((Yindex - YCtr) * DeltaUW)

    #Dg=Dgy
    Dg = np.append(Dgy, np.zeros((YL, ZL, ProjScale - cfg.protocol.viewCount)), axis=2)# try padding (cut unwantted section from the scanned data)
```

What this does is it pads the missing section from a short scan (pretending it's a full scan but fill the missing section with 0), and then reconstructs using pre-defined view counts.

Running the main script with `protocol.viewsPerRotation = 100` and `protocol.viewCount = 50`. You should get the result shown below:



2.2 Running Short Scan - Helical Scanning

Running Short Scan with Helical Scanning mode is much easier. Simply set `recon.reconType = 'helical_equiAngle'` in `recon.cfg`. You don't need to modify anything, but the table speed must be non-zero otherwise the recon result will be pure dark image.