

## Function to generate head phantom and ROI projection data from cookie cutter method

By: Karthik Chellamuthu and Michael Mow

```
% Last updated: 2/27/2016

% Inputs:
% frame_rate      : frame rate of the scanner
% acquisition_time : total time of the scan
% radii           : vector of radii for each ROI
% centers         : centers of each ROI
% filename        : filename to store data to into inData folder

% Outputs:
% HEAD           : struct containing the projections of the head
                  % phantom, the frame rate, and acquisition time
% ROI            : struct containing the projections of each ROI

function [HEAD,ROI]=projections_generator(frame_rate,acquisition_time,radii,centers,filename)

% directory to load files from and save files to
vnDir = 'inData';
% input image mat file name
vnFN = 'head';

% load head phantom image from inData folder
vn = load([vnDir, '/', vnFN, '.mat']);
im = vn.u;
clear vn;

%ROI struct: include ROI_projs, pm, G
% loop through each ROI in struct
for i=1:length(radii)

    % place radius and center into struct
    ROI(i).radius = radii(i);
    ROI(i).center = centers(i,:);

    % generate a 0 1 ROI with specified center and radius
    ROI(i).s = ROI_generator(size(im),ROI(i).center,ROI(i).radius);

    %use if want to use only spherical ROI and no head phantom
    %im = ROI(i).s *0.02;

    % create spherical ROI
    ROI(i).cutout = 1 - transramp(xyzfunc(['sqrt((Y-', num2str(ROI(i).center(2)), ...
        ').^2+(X-', num2str(ROI(i).center(1)), ').^2+(Z-', num2str(ROI(i).center(3)), ').^2)']...
        , size(im), [1 size(im,1)], [1 size(im,2)], [1 size(im,3)]),ROI(i).radius,ROI(i).radius+1);

    % smooth ROI
    ROI(i).cutout = smooth3(ROI(i).cutout);

    % find locations of ROI in image
    ROI(i).locations = find(ROI(i).s == 1);

    % calculate background mean attenuation in ROI
    ROI(i).bg_att = mean(im(ROI(i).locations));

    % convert to HU
    ROI(i).bg_HU = (ROI(i).bg_att-0.02)/0.02*1000;

    % scale ROI cutout by background
    ROI(i).cutout = ROI(i).cutout*ROI(i).bg_att;

    % create projections for ROI
    [ROI(i).projs,ROI(i).pm, ROI(i).G] = static_projector(ROI(i).cutout,frame_rate,acquisition_time);

    % cut out ROI from head phantom image
    im(ROI(i).locations) = 0;

    clear vn;
end

% create projections for head phantom
[HEAD.projs,HEAD.pm,HEAD.G] = static_projector(im,frame_rate,acquisition_time);

% store frame rate and acquisition time in head struct
HEAD.frame_rate = frame_rate;
HEAD.acquisition_time = acquisition_time;

% save files to inData folder
save([vnDir, '/head_projections_', filename, '.mat'], 'HEAD', '-v7.3');
```

```
save([vnDir, '/ROI_projections_', filename, '.mat'], 'ROI', '-v7.3');  
end
```