Function to create static projections

By: Karthik Chellamuthu and Michael Mow

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

% Adapted from 'projectSimple.m' from Alejandro Sisniega

% Inputs:
    % image_matrix            : image file to be projected
    % frame_rate       : frame rate of the scanner
    % acquisition_time : total time of the scan

% Outputs:
    % y                : output matrix containing projections
    % pm               : projection matrices for all projections
    % G                : struct containing geometrical parameters of the
                         % system
function [y,pm,G] = static_projector(image_matrix,frame_rate,acquisition_time)

% Set the properties of the volumes
voxSize = [0.5 0.5 0.5];
volTransforms = [0 0 0 0 0 0];

% angular separation between projections
angleStep = 1;

% number of projections
nAngle = acquisition_time*frame_rate;

% GPU ID
nGPU = 1;

% geometry structure storing geometrical parameters of the system and
% detector sizez, pixel size, etc.
G.UVWdim = [512 512];
G.PixSize = [1 1];
G.angle = 0:angleStep:((nAngle-1)*angleStep);
% set offsets to perfect detector
G.u0 = G.UVWdim(1)/2;
G.v0 = G.UVWdim(2)/2;
% set source to axis and source to detector distances
G.SAD = 600;
G.SDD = 1200;

% create cudatools object
ct = CudaTools.Reconstruction(nGPU);

% add projections to the geometry
nb = length(G.angle);
pm = zeros(3,4,nb);
for b=1:nb,
    u0_l = (G.UVWdim(1)/2 - G.u0)*G.PixSize(1);
    v0_l = (G.v0 - G.UVWdim(2)/2)*G.PixSize(2);
    srcX = 0;
    srcZ = 0;
    outOfPlaneAngle = 0;
    inOfPlaneAngle = 0;
    ct.SetGeometryCircular(G.angle(b)*(pi/180),[0,0,1],G.SAD,G.SDD,...
        u0_l,v0_l,outOfPlaneAngle,inOfPlaneAngle,srcX,srcZ);
    pm(:,:,b) = ct.GetGeometry();
end;

% set the final geometry with all the projections matrices
ct.SetGeometry(pm);

% set the volume and the transforms with the included motion
vCT = ct.SetImage('vol',image_matrix,voxSize,volTransforms);

% create the 3D matrix to store the projections
y = zeros(G.UVWdim(1),G.UVWdim(2),nb);

% create the memory to store projections in the GPU
pCT = ct.SetImage('proj',y,[G.PixSize(1),G.PixSize(2),1]);

% project the data
tic;
disp(['Starting forward projection']);
ct.ProjectorSFForward(vCT, pCT); % Separable footprint
y = single(ct.GetImage(pCT).values);
toc;
```

```
% remove the ref to cudatools object
clear ct;

end
```

Published with MATLAB® R2015a