ReconDBT: Reconstruction Package for Digital Breast Tomosynthesis (DBT)

Rongping Zeng

I. Summary

This package, named *ReconDBT*, contains open-source reconstruction functions for digital breast tomosynthese (DBT) written in Mathworks' MATLAB language. Filtered back projection (FBP) and two iterative reconstruction (IR) functions, the simultaneous algebraic reconstruction technique (SART) and maximum likelihood (ML), are included in this package [1]. Example codes are provided on how to use the three reconstruction functions for the common circular-arc trajectory DBT and non-circular arc trajectory DBT. *ReconDBT* is built on top of the publically available CT reconstruction code developed by Dr. Fessler [2]. The package runs on Linux and Mac operating systems. Note that it does not run on Windows operating systems because Fessler's CT reconstruction package does not support Windows system.

Contact for questions:
Rongping Zeng
rongping.zeng@fda.hhs.gov

II. Legal Disclaimer

This software and documentation (the "Software") were developed at the Food and Drug Administration (FDA) by employees of the Federal Government in the course of their official duties. Pursuant to Title 17, Section 105 of the United States Code, this work is not subject to copyright protection and is in the public domain. Permission is hereby granted, free of charge, to any person obtaining a copy of the Software, to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, or sell copies of the Software or derivatives, and to permit persons to whom the Software is furnished to do so. FDA assumes no responsibility whatsoever for use by other parties of the Software, its source code, documentation or compiled executables, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. Further, use of this code in no way implies endorsement by the FDA or confers any advantage in regulatory decisions. Although this software can be redistributed and/or modified freely, we ask that any derivative works bear some notice that they are derived from it, and any modified versions bear some notice that they have been modified.

For the part of Fessler's CT reconstruction package, refer to the open-source license specified at the website http://web.eecs.umich.edu/~fessler/code/index.html. You may cite Fessler's CT reconstruction package as follows: Jeffrey A Fessler, "Michigan Image Reconstruction Toolbox," available at [url], downloaded at [url]".

II. Instructions for ReconDBT setup

To run and test your setup, following these three steps.

Step 1: Download necessary files.

- The ReconDBT package in Github at https://github.com/DIDSR/ReconDBT/ReconDBT.tar.gz.
- Dr. Fessler's CT reconstruction package. You can either download the whole package from http://web.eecs.umich.edu/~fessler/irt/fessler.tgz for convenience, or only the codes in these four individual folders from http://web.eecs.umich.edu/~fessler/irt/irt/ that are used by ReconDBT to save storage space: "/fbp", "/systems", "/utilities" and "/mex/v7".

Step 2: Run "setpath_dbt" in MATLAB.

This step includes the path of the reconstruction packages into the MATLAB workspace. Users need to modify the paths in "setpath_DBT.m" to reflect the actual paths where you place the ReconDBT package and Fessler's CT reconstruction codes.

Step 3: Run "test_ReconDBT" to test your setup.

Run "test_ReconDBT" in MATLAB. If no error occurs, then you should have correctly set up the package. Otherwise, go back to step 2 and check if the paths are correct in "setpath_DBT.m".

At the end of the run, two figures will appear to display a reconstruction slice that looks like the image below in Figure 1.

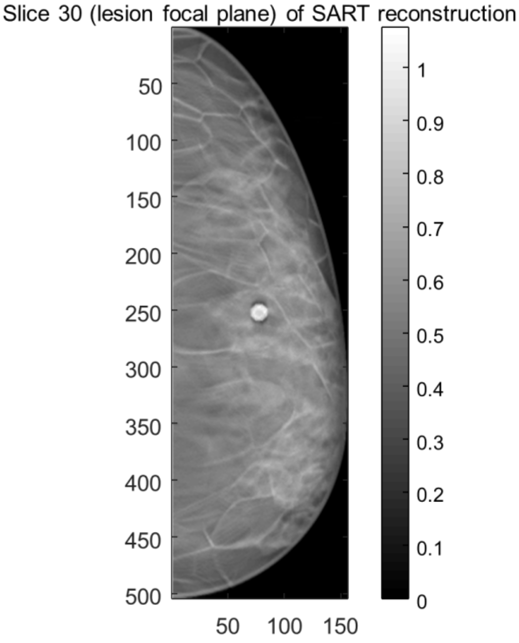


Figure 1. Slice images that are displayed at the end of the run of "test_ReconDBT.m". The image contains a bright lesion at the center.

IV. Usage of the reconstruction functions

There are three reconstruction algorithms implemented in this package:

- Filter back projection (FBP), in “FBP_dbt.m”
- Simultaneous algebraic reconstruction technique (SART), in “SART_dbt.m”
- Maximum-likelihood reconstruction (ML), in “ML_dbt.m.

Users can refer to the appendix in reference [1] for a description of the three reconstruction algorithms implemented in the package.

Two example reconstruction codes: “DBT_recon_circular_example.m” and “DBT_recon_noncircular_example.m”, are provided to help users learn how to prepare inputs for properly running the reconstruction functions. Both example codes take the pre-generated projection data by the code “gen_dbtproj_example.m” to reconstruct 3D volumes with the reconstruction functions. Users can directly find the description of parameters in the codes or in this document as described next.

➢ Reconstruction of circular-arc trajectory DBT

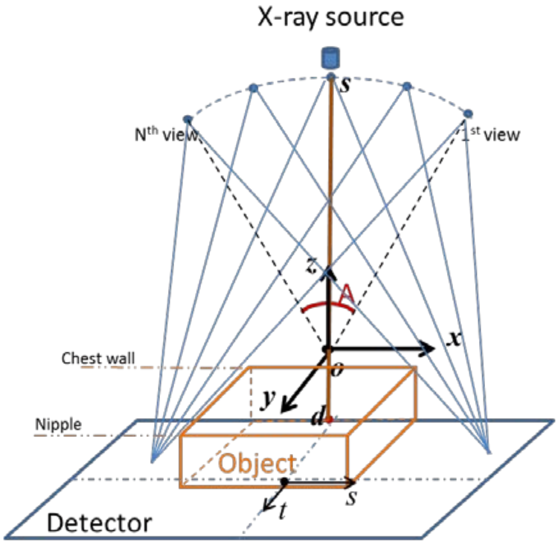
Circular-arc trajectory is the most common scan geometry used in commercial DBT systems, in which the x-ray tube rotates around a pivot point above the detector and fires x-ray at evenly spaced angles. The scan geometry is mainly defined by the number of projection views and the angular span. If the projection data is collected or simulated from an arc-trajectory DBT, use “DBT_recon_circular_example.m” as your template and modify the parameter values for your own reconstruction.

Three categories of inputs are needed by the DBT reconstruction functions: 1) the DBT projection data, 2) the DBT system geometry parameters and 3) the reconstruction volume parameters.

Table 1 lists of the input parameters and Figure 2 plots the arrangement of the origin and dimensions of the 3D coordinate system related to the DBT system.

Table 1: Input parameters for arc trajectory DBT.

Parameter category	**Variable names used in the code**	**Comments**
Projection data	*g*	*(For FBP and SART reconstruction functions)* The DBT projection views as line integrals of attenuation, a 3D array of size *ns* × *nt* × *na*.
	(proj, I0)	*(For ML reconstruction function)* "*proj*" is the DBT projection views as photon counts and *I0* is the air photon counts. Both are 3D array of size ns × nt × na. *I0* can be set to a single value for the case of uniform exposure over all detector pixels.
DBT scan geometry	*(dso, dod)*	Distance from the source to the rotation center and distance from the rotation center to the detector. Unit in cm.
	orbit	The angular span in degree.
	na	The number of projection views.
	(ds, dt)	Detector pixel size in cm. "s" represents the x-ray tube moving direction and "t" the perpendicular direction to "s" direction. Unit in cm.
	(ns, nt)	The number of detector pixels along the "s" and "t" directions
	(offset_s, offset_t)	The offset of the detector center in the "s" and "t" dimensions relative to the origin of the 3D coordinate system. Unit in detector pixel. Default value is "0".
	*d_objbottom_det_*	Distance from the bottom of the object being scanned to the detector. Unit in cm. Default value is "0" indicating the object is placed right on the detector without a gap.
Reconstruction volume	*(drx, dry, drz)*	The voxel size of the reconstructed volume in x, y and z dimensions. Unit in cm.
	(nrx, nry, nrz)	The number of voxels contained in the reconstructed volume.
	(offset_x, offset_y, offset_z)	The offset of the reconstructed volume center in the x, y and z dimensions relative to the origin. Unit in voxels.



Circular arc trajectory is defined by:

- The angular span A
- The number of views N
- The distance from source to rotation center d_{so}
- The distance from the source to detector d_{sd}

The 3D coordinate system:

- The origin 'o' is the rotation center of the x-ray source
- +*x*: toward right
- +*y*: toward the nipple
- +z: toward the x-ray source

Figure 2: Coordinate system defined for the circular-arc trajectory DBT system

After the input parameter values are provided, use the following lines to perform the reconstruction, as provided in “DBT_recon_circular_example.m”:

```
%Generate the system structures using the input parameters
```

```
igr = image_geom('nx', nrx, 'ny',nry, 'nz', nrz, 'dx',drx, 'dz', drz,'offset_y',
offset_y,'offset_z', offset_z,'down', 1);


btg = bt_geom('arc', 'ns', ns, 'nt', nt, 'na', na, 'ds', ds, ‘down', 1,
'orbit', orbit, ‘offset_s', 0, 'offset_t', offset_t,'dso', dso, 'dod', dod,
'dfs',inf);

Gtr = Gtomo_syn(btg,igr);
```

```
%For FBP reconstruction. User controllable variable is the filter function ‘filter’.
```

```
filter = ‘hann75’
xfbp = fbp_dbt(Gtr,btg,igr,g,filter);
```

```
%For SART reconstruction. User controllable variables are the number of iterations
‘niter’ and the step size ‘stepsize’.
```

```
xbp = BP(Gtr, g); %initialization from a simple back projection volume
niter = 2;
stepsize=0.5;
[xart,cost] = SART_dbt(Gtr,g,xbp,niter,stepsize);
```

```
%For ML reconstruction. User controllable variables are the number of iterations
‘niter’ and the step size ‘stepsize’.
```

```
xbp = BP(Gtr, g); %initialization from a simple back projection volume
niter=3;
stepsize=2;
[xmlt,cost] = ML_dbt(Gtr,proj,xbp,I0,3,2);
```

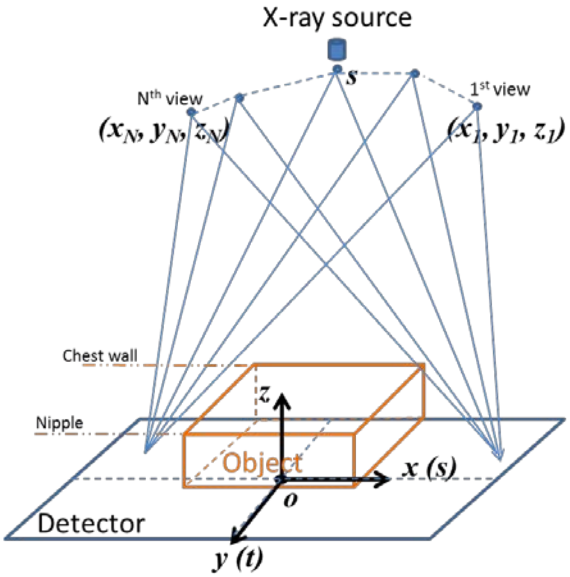
➢ Reconstruction of non-circular arc trajectory DBT

For non-circular arc trajectory, the x-ray source may move along a line or other arbitrary curves, or a circular trajectory but nonuniformly spaced projection angles. The scan geometry is mainly defined by the x-ray source positions corresponding to each projection view. This may be useful for researchers investigating experimental DBT systems with atypical tube moving designs. If the projection data is collected or simulated from non-circular arc trajectory DBT, use “DBT_recon_noncircular_example.m” as your template and modify the parameter values for your own reconstruction.

Table 2 for a list of the input parameters and Figure 3 for the arrangement of the origin and dimensions of the 3D coordinate system related to the DBT system. Note that only SART and ML are used for non-arc trajectory DBT reconstruction. FBP is not applicable to this type of DBT system.

Table 2: Input parameters for non-arc trajectory DBT.

Parameter category	**Variable names used in the code**	**Comments**
Projection data	*g*	*(For SART reconstruction functions)* The DBT projection views as line integrals of attenuation, a 3D array of size *ns* × *nt* × *na*.
	(proj, I0)	*(For ML reconstruction function)* "*proj*" is the DBT projection views as photon counts and *I0* is the air photon counts. Both are 3D array of size ns × nt × na. *I0* can be set to a single value for the case of uniform exposure over all detector pixels.
DBT scan geometry	*src_vec*	The x-ray source coordinates for each projection view, a na × 3 matrix. Each element values are in the unit of cm
	na	The number of projection views.
	(ds, dt)	Detector pixel size in cm. "s" represents the x-ray tube moving direction and "t" the perpendicular direction to "s" direction. Unit in cm.
	(ns, nt)	The number of detector pixels along the "s" and "t" directions
	(proj_offset_s, proj_offset_t)	The offset of the projection view center in the "s" and "t" dimension relative to the origin of the 3D coordinate system. Unit in detector pixel. Default value is "0".
	d_objbottom_det	Distance from the bottom of the object being scanned to the detector. Unit in cm. Default value is "0" indicating the object is placed right on the detector without a gap.
Reconstruction volume	*(drx, dry, drz)*	The voxel size of the reconstructed volume in x, y and z dimensions. Unit in cm
	(nrx, nry, nrz)	The number of voxels contained in the reconstructed volume.
	(offset_x, offset_y, offset_z)	The offset of the reconstructed volume center in the x, y and z dimensions relative to the origin. Unit in voxels.



Non-circular arc trajectory is defined by the coordinates of the source positions, $\{(x_i, y_i, z_i)\}$, $i=1,\dots N$, where N is the number of projection views.

The 3D coordinate system:

- The origin 'o' is the center of the detector
- +x: toward right
- +y: toward the nipple
- +z: toward the x-ray source

Figure 3. Coordinate system defined for the non-circular arc trajectory DBT system

After the input parameter values are provided, use the following lines to perform the reconstruction, as provided in “DBT_recon_noncircular_example.m”:

```
%Generate the system structures using the input parameters
```

```
igr = image_geom('nx', nrx, 'ny',nry, 'nz', nrz, 'dx',drx, 'dz', drz,...
       'offset_x', offset_x, 'offset_y', offset_y,'offset_z', offset_z,'down', 1);

dgeo.ns = ns;
dgeo.nt = nt;
dgeo.ds = ds;
dgeo.dt = dt;

Gtr = Gtomo_syn_noncircular(src_vec,igr, dgeo, projview_offset);
```

```
%For SART reconstruction. User controllable variables are the number of iterations
‘niter’ and the step size ‘stepsize’.
```

```
xbp = BP(Gtr, g); %initialization from a simple back projection volume
niter = 2;
stepsize=0.5;
[xart,cost] = SART_dbt(Gtr,g,xbp,niter,stepsize);
```

```
% For ML reconstruction. User controllable variables are the number of iterations
‘niter’ and the step size ‘stepsize’.
```

```
xbp = BP(Gtr, g); %initialization from a simple back projection volume
niter=3;
stepsize=2;
[xmlt,cost] = ML_dbt(Gtr,proj,xbp,I0,3,2);
```

References:

[1] R Zeng, S Park, P Bakic, K Myers, “Evaluating the sensitivity of the optimization of acquisition parameters to the choice of reconstruction algorithm in digital breast tomosynthesis through a simulation study”, Phys. in Med. Biol., 60, 1259-1288, 2015

[2] Jeffrey A Fessler, "Michigan Image Reconstruction Toolbox,"
http://web.eecs.umich.edu/~fessler/code/index.html