General Description:
Software for reconstructing 2D/3D x-ray and neutron tomography datasets. The data can be undersampled, of poor contrast, noisy, and contain various artifacts. This is Matlab and C-omp implementation of iterative model-based algorithms with unconventional data fidelities and with various regularization terms (TV and higher-order LLT). The main optimization problem is solved using FISTA framework [1]. The presented algorithms are FBP, FISTA (Least-Squares), FISTA-LS-TV(LS-Total Variation), FISTA-GH(Group-Huber)-TV, and FISTA-Student-TV. More information about the algorithms can be found in papers [2,3]. Please cite [2] if the algorithms or data used in your research.

Requirements/Dependencies:
MATLAB (www.mathworks.com/products/matlab/)
ASTRA toolbox (https://github.com/astra-toolbox/astra-toolbox)
C/C++ compiler (run compile_mex in Matlab first to compile C-functions)

Package Contents:
Demos:
-Demo1: Synthetic phantom reconstruction with noise, stripes and zingers
-Demo2: Synthetic phantom reconstruction with noise, stripes, zingers, and the missing wedges
-DemoRD1: Real data reconstruction from sino_basalt.mat (see Data)
-DemoRD2: Real data reconstruction from sino3D_dendrites.mat (see Data)
Data:
- phantom_bone512.mat - a synthetic 2D phantom obtained from high-resolution x-ray scan
- sino_basalt.mat – 2D neutron (PSI) tomography sinogram (slice across a pack of basalt beads)
- sino3D_dendrites.mat – 3D (20 slices) x-ray synchrotron dataset (DLS) of growing dendrites
Main modules:
- FISTA_REC.m – Matlab function to perform FISTA-based reconstruction
- FISTA_TV.c – C-omp function to solve for the weighted TV term using FISTA
- SplitBregman_TV.c – C-omp function to solve for the weighted TV term using Split-Bregman
- LLT_model.c – C-omp function to solve for the weighted LLT [3] term using explicit scheme
- studentst.m – Matlab function to calculate Students t penalty with 'auto-tuning'
Supplementary:
- zing_rings_add.m Matlab script to generate proj. data, add noise, zingers and stripes
- add_wedges.m script to add the missing wedge to existing sinogram
- my_red_yellowMAP.mat – nice colormap for the phantom
- RMSE.m – Matlab function to calculate Root Mean Square Error
- subplot_tight – visualizing better subplots
- ssim_index – ssim calculation

Practical advices:
- Full 3D reconstruction provides much better results than 2D. In the case of ring artifacts, 3D is almost necessary
- Depending on data it is better to use TV-LLT combination in order to achieve piecewise-smooth solution. The DemoRD2 shows one possible example when smoother surfaces required.
- **L** (Lipshitz constant) if tweaked can lead to faster convergence than automatic values
- Convergence is normally much faster when using Fourier filtering before backprojection
- Students't penalty is generally quite stable in practice, however some tweaking of **L** might require for the real data
- You can choose between SplitBregman-TV and FISTA-TV modules. The former is slower but requires less memory (for 3D volume U it can take up to 6 x U), the latter is faster but can take more memory (for 3D volume U it can take up to 11 x U). Also the SplitBregman is quite good in improving contrast.

Compiling:

It is very important to check that OMP support is activated for the optimal use of all CPU cores.

In Windows enable OMP support, e.g.:

edit C:\Users\Username\AppData\Roaming\MathWorks\MATLAB\R2012b\mexopts.bat

In the file, edit 'OPTIMFLAGS' line as shown below (add /openmp entry at the end):

OPTIMFLAGS = /O2 /Oy- /DNDEBUG /openmp

define and set the number of CPU cores

PROC = feature('numCores');

PROCS = num2str(PROC);

setenv('OMP_NUM_THREADS', PROCS); % you can check how many CPU's by: getenv OMP_NUM_THREADS

In Linux in terminal: export OMP_NUM_THREADS = (the numer of CPU cores)

then run Matlab as normal

to compile with OMP support from Matlab in Windows run:

mex *.cpp CFLAGS="\$CFLAGS -fopenmp -Wall" LDFLAGS="\$LDFLAGS -fopenmp"

to compile with OMP support from Matlab in Linux ->rename *cpp to *c and compile:

mex *.c CFLAGS="\$CFLAGS -fopenmp -Wall" LDFLAGS="\$LDFLAGS -fopenmp"

References:

[1] Beck, A. and Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*,*2*(1), pp.183-202.

[2] Kazantsev, D., Bleichrodt, F., van Leeuwen, T., Kaestner, A., Withers, P.J., Batenburg, K.J., 2017. A novel tomographic reconstruction method based on the robust Student's t function for suppressing data outliers, IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING (to appear)

[3] Lysaker, M., Lundervold, A. and Tai, X.C., 2003. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Transactions on image processing*, *12*(12), pp.1579-1590.

[4] Paleo, P. and Mirone, A., 2015. Ring artifacts correction in compressed sensing tomographic reconstruction. *Journal of synchrotron radiation*, *22*(5), pp.1268-1278.

[5] Sidky, E.Y., Jakob, H.J. and Pan, X., 2012. Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle-Pock algorithm. *Physics in medicine and biology*, *57*(10), p.3065.

D. Kazantsev / Harwell Campus / 16.03.17

any questions/comments please e-mail to daniil.kazantsev@manchester.ac.uk