
APPENDIX B

MATLAB CODE

B.1 Description of Functions and Scripts

Below is a list of the functions and scripts used in this thesis. The list is organized in a hierarchical order such that each primary function is listed followed by the subfunctions it calls in the next hierarchy level. The subfunctions are listed in alphabetical order, not the order in which they are called.

1. **csMRI.m** – Performs compressed sensing (CS) recovery techniques on undersampled real valued MRI images (k -space data is simulated by taking the Fourier transform of the image).
 - (a) **addNoise.m** – Adds Gaussian white noise to an image for a specified signal-to-noise ratio (SNR).
 - (b) **errorMetrics.m** – Computes image quality metrics such as PSNR and MSSIM for image comparison.
 - i. **double2int.m** – Converts a grayscale image of double precision in the range 0.0-1.0 back to its original integer dynamic range.
 - ii. **ssim_wang.m** – Computes the MSSIM quality score of an image relative to a reference. This function is the same exact function as `ssim.m` [49], but renamed in order not to conflict with the MATLAB function `ssim.m` found in the image processing toolbox [54].
 - (c) **imThresh.m** – Thresholds a real valued image so that pixel values lie in a specified range.

- (d) **my_im2col.m** – Breaks up image into overlapping image patches. This function is available in the K-SVD MATLAB toolbox [55].
 - (e) **myKSVD.m** – Performs the K-SVD algorithm. It is a slightly modified version of `KSVD2.m` contained in the DLMRI-Lab toolbox [50]. The functionality of `myKSVD.m` is identical to that of `KSVD2.m`, except that it returns a few additional output parameters useful for data analysis. The user can replace `myKSVD.m` with `KSVD2.m`.
 - (f) **OMPerrn.m** – Performs orthogonal matching pursuit algorithm. Allows both sparsity and error constraints. Modified version of `OMPerrn.m` found in the DLMRI-Lab toolbox [50].
 - (g) **overcompleteDCTdict.m** – Creates an overcomplete 2D DCT dictionary with zero mean atoms. This function is based off of `denoiseImageDCT.m` found in the K-SVD MATLAB toolbox [55].
 - (h) **pcaDictionary.m** – Generates a sparsifying dictionary using principle component analysis (PCA). This code was adopted from `DLMRI.m` found in the DLMRI-Lab toolbox [50].
2. **genSamplingMask.m** – Generates a k -space sampling mask with random Cartesian trajectories (phase encodes) chosen from a Normal or Laplace distribution.
 - (a) **laplacePDF.m** – Evaluates the Laplace distribution at specified points.
 3. **csMRIscript_demo.m** – Script that demonstrates how to set parameters for using `csMRI.m`.
 4. **LearnPriorDict_demo.m** – Script that demonstrates how image priors are learned for brain scans via the K-SVD algorithm.
 5. **mnc2mat.m** – Script that loads all `.mnc` files (containing 3D volume data) in a specified location into 3D arrays and stores the array as a `.mat` file.
 - (a) **myloadminc.m** – Loads minc format (`.mnc`) input file into MATLAB. This function is a modified version of `loadminc.m` [56].