VICTRE-MO: Model Observers for VICTRE

Rongping Zeng

I. Summary

This package, named *VICTRE-MO*, contains open-source model observer functions to perform location-known lesion detection tasks. The codes were written in Mathworks' MATLAB language. Channelized Hotelling observer (CHO) and convolution CHO with Laguerre-Gauss channels were implemented in this package.

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.

III. Functions

Two model observer functions were provided in this package, named "conv_LG_CHO_2d" for processing 2D images and "conv_LG_CHO_3d" for processing 3D images. Example codes to call and run the two functions were also provided to help users understand how to correctly apply the functions for their own detection task.

- **Conv_LG_CHO_2d.m**
  function [snr, t_sp, t_sa, chimg, tplimg, meanSP, meanSA, meanSig, k_ch] = **conv_LG_CHO_2d** (trimg_sa, trimg_sp, testimg_sa, testimg_sp, ch_width, nch, b_conv)

<u>Functionality</u>: location-known signal detection in 2D images performed by a LG channelized model observer.

```
%Inputs
%   trimg_sa: the signal-absent training set;
%   trimg_sp: the signal-present training set;
%   testimg_sa: the signal-absent testing set;
%   testimg_sp: the signal-present testing set;
%   ch_width: channel width parameter;
%   nch: number of channels;
%   b_conv: 1 or 0 to indicate whether to apply convolutional
%           LG channels. Default is 1.
%Outputs
%   snr: the detectibility SNR
%   t_sp: t-scores of signal-present cases
%   t_sa: t-scores of signal-absent cases
%   chimg: The channel matrix, Nx x Ny x nch, where Nx x Ny is
%          the image size
%   tplimg: the model observer template, Nx x Ny
%   meanSP: the average signal-repsent image
%   meanSA: the average signal-absent image
%   meanSig: the average signal image
%   k_ch: data covariance matrix, nch x nch
```

Note: In MATLAB, run this example code ***"run_ffdm_2dCHO.m"*** to learn how to use it:

- **Conv_LG_CHO_3d.m**
  function [snr, t_sp, t_sa, chimg, tplimg, meanSP, meanSA, meanSig, k_ch] = **conv_LG_CHO_3d** (trimg_sa, trimg_sp, testimg_sa, testimg_sp, ch_width, nch, b_conv)

<u>Functionality</u>: location-known signal detection in 3D images performed by a LG channelized model observer.

```
%Inputs
%   trimg_sa: the signal-absent training set;
%   trimg_sp: the signal-present training set;
%   testimg_sa: the signal-absent testing set;
%   testimg_sp: the signal-present testing set;
%   ch_width: channel width parameter;
%   nch: number of channels to be used;
%   b_conv: 1 or 0 to indicate whether to apply convolutional
%           LG channels. Default is 1.
%
%Outputs
%   snr: the detectibility SNR
%   t_sp: t-scores of signal-present cases
%   t_sa: t-scores of signal-absent cases
%   chimg: The channel matrix, Nx x Ny x Nz x nch, where Nx x
%          Ny x Nz is the image size
%   tplimg: the model observer template, Nx x Ny x Nz
%   meanSP: the average signal-repsent image
%   meanSA: the average signal-absent image
%   meanSig: the average signal image
%   k_ch: data covariance matrix
```

Note: In MATLAB, run this example code ***“run_dbt_3dCHO.m”*** to learn how to use it.

IV. References

Users can refer to these two papers for general LG-CHO[1, 2] and the concept of convolutional CHO[3]:

[1] B. D. Gallas and H. H. Barrett, "Validating the use of channels to estimate the ideal linear observer," *J. Opt. Soc. Am. A,* vol. 20, pp. 1725-1738, 2003.

[2] R. Zeng, S. park, P. Bakic, and K. Myers, J., "Evaluating the sensitivity of the optimization of acquisition geometry to the choice of reconstruction algorithm in digital breast tomosynthesis through a simulation study," *Physics in Medicine and Biology,* vol. 60, p. 1259, 2015.

[3] I. Diaz, C. K. Abbey, P. A. S. Timberg, M. P. Eckstein, F. R. Verdun, C. Castella*, et al.*, "Derivation of an Observer Model Adapted to Irregular Signals Based on Convolution Channels," *IEEE Transactions on Medical Imaging,* vol. 34, pp. 1428-1435, 2015.