pytigre Documentation

Reuben Lindroos, Ander Biguri

Apr 30, 2019

CONTENTS:

CHAPTER

ONE

INDICES AND TABLES

- genindex
- modindex
- search

CHAPTER

TWO

CHAPTER ON MODULES

tigre.algorithms.**sirt**(*proj*, *geo*, *angles*, *niter*, ***kwargs*)

SART_CBCT solves Cone Beam CT image reconstruction using Oriented Subsets Simultaneous Algebraic Reconxtruction Techique algorithm SIRT(PROJ,GEO,ALPHA,NITER) solves the reconstruction problem using the projection data PROJ taken over ALPHA angles, corresponding to the geometry descrived in GEO, using NITER iterations.

param proj (np.ndarray, dtype=np.float32)

Input data, shape = (geo.nDector, nangles)

param geo (tigre.geometry)

Geometry of detector and image (see examples/Demo code)

param angles (np.ndarray , dtype=np.float32)

angles of projection, shape = (nangles,3)

param niter (int)

number of iterations for reconstruction algorithm

param kwargs (dict)

optional parameters

keyword blocksize (int) number of angles to be included in each iteration of proj and backproj for OS_SART

keyword lmbda (np.float64) Sets the value of the hyperparameter.

keyword lmbda_red (np.float64) Reduction of lambda every iteration lambda=lambdared*lambda. Default is 0.99

keyword init (str) Describes different initialization techniques.

"none" : Initializes the image to zeros (default) "FDK" : intializes image to FDK reconstrucition "multigrid": Initializes image by solving the problem in

small scale and increasing it when relative convergence is reached.

"image" [Initialization using a user specified] image. Not recommended unless you really know what you are doing.

keyword InitImg (np.ndarray) Not yet implemented. Image for the "image" initialization.

keyword verbose (Boolean) Feedback print statements for algorithm progress default=True

keyword Quameasopts (list) Asks the algorithm for a set of quality measurement parameters. Input should contain a list or tuple of strings of quality measurement names. Examples:

RMSE, CC, UQI, MSSIM

:keyword OrderStrategy [(str)]

Chooses the subset ordering strategy. Options are:

"ordered" [uses them in the input order, but] divided

"random" : orders them randomply "angularDistance": chooses the next subset with the

biggest angular distance with the ones used

```
>>> import numpy as np
>>> import tigre
>>> import tigre.algorithms as algs
>>> from tigre.demos.Test_data import data_loader
>>> geo = tigre.geometry(mode='cone',default_geo=True,
>>>                          nVoxel=np.array([64,64,64]))
>>> angles = np.linspace(0,2*np.pi,100)
>>> src_img = data_loader.load_head_phantom(geo.nVoxel)
>>> proj = tigre.Ax(src_img,geo,angles)
>>> output = algs.iterativereconalg(proj,geo,angles,niter=50
>>>                                 blocksize=20)
```

tigre.demos.run() to launch ipython notebook file with examples.

tigre.algorithms.**sart**(*proj*, *geo*, *angles*, *niter*, ***kwargs*)

SART_CBCT solves Cone Beam CT image reconstruction using Oriented Subsets Simultaneous Algebraic Reconstruction Techique algorithm SART(PROJ,GEO,ALPHA,NITER) solves the reconstruction problem using the projection data PROJ taken over ALPHA angles, corresponding to the geometry described in GEO, using NITER iterations.

param proj (np.ndarray, dtype=np.float32)

Input data, shape = (geo.nDector, nangles)

param geo (tigre.geometry)

Geometry of detector and image (see examples/Demo code)

param angles (np.ndarray , dtype=np.float32)

angles of projection, shape = (nangles,3)

param niter (int)

number of iterations for reconstruction algorithm

param kwargs (dict)

optional parameters

keyword blocksize (int) number of angles to be included in each iteration of proj and backproj for OS_SART

keyword lmbda (np.float64) Sets the value of the hyperparameter.

keyword lmbda_red (np.float64) Reduction of lambda every iteration lambda=lambdared*lambda. Default is 0.99

keyword init (str) Describes different initialization techniques.

"none" : Initializes the image to zeros (default) "FDK" : intializes image to FDK reconstrucition "multigrid": Initializes image by solving the problem in

small scale and increasing it when relative convergence is reached.

"image" [Initialization using a user specified] image. Not recommended unless you really know what you are doing.

keyword InitImg (np.ndarray) Not yet implemented. Image for the "image" initialization.

keyword verbose (Boolean) Feedback print statements for algorithm progress default=True

keyword Quameasopts (list) Asks the algorithm for a set of quality measurement parameters. Input should contain a list or tuple of strings of quality measurement names. Examples:

RMSE, CC, UQI, MSSIM

:keyword OrderStrategy [(str)]

Chooses the subset ordering strategy. Options are:

"ordered" [uses them in the input order, but] divided

"random" : orders them randomply "angularDistance": chooses the next subset with the

biggest angular distance with the ones used

```
>>> import numpy as np
>>> import tigre
>>> import tigre.algorithms as algs
>>> from tigre.demos.Test_data import data_loader
>>> geo = tigre.geometry(mode='cone',default_geo=True,
>>>                          nVoxel=np.array([64,64,64]))
>>> angles = np.linspace(0,2*np.pi,100)
>>> src_img = data_loader.load_head_phantom(geo.nVoxel)
>>> proj = tigre.Ax(src_img,geo,angles)
>>> output = algs.iterativereconalg(proj,geo,angles,niter=50
>>>                                 blocksize=20)
```

tigre.demos.run() to launch ipython notebook file with examples.

PYTHON MODULE INDEX

t

INDEX

S

T