

Exercises Instructions

Introduction

This material is intended for practical demonstration using STIR on PET Image Reconstruction.

Software is available as source code or precompiled on Linux, Mac and Windows:

<http://stir.sourceforge.net/PETMRTraining2015/>

Simulated data will be prepared during the exercises. These are based on 2 sets of images:

- Thorax phantom data are obtained from the recent open access article: *Tsoumpas et al 2013 Phys Med Biol*.
We have two respiratory gated positions of a thorax FDG PET phantom along with the corresponding CTAC image.
- Brain data are obtained from BrainWeb.
We have a segmented brain-map.

You will probably only want to run either the brain or the thorax data (except for the motion correction exercise which is currently only for the thorax).

The input data are stored in the folders called EX_*, **but you will need to run the scripts from the “main” exercises folders** (open a terminal, cd to where you extracted the exercises, and always cd back after every exercise).

See the appendices at the end of this document for some information to get started. Note that in the text below we're using AMIDE for display. ImageJ would work as well (see the end of the document).

Authors:

Elisabetta Grecchi, Irene Polycarpou, Charalampos Tsoumpas, Kris Thielemans

Exercise: Data Simulation Brain

(Always run scripts from the exercises directory)

This is a simple simulation of a brain phantom. Scatter is set to zero. Randoms are constant.

Read and run script:

```
./run_simulations_brain.sh
```

You will need to extract the sinograms in an “image” Interfile to be able to load them in AMIDE

```
cd working_folder/brain
```

```
extract_segments my_prompts.hs
```

```
Extract as SegmentByView (0) or BySinogram (1)?[0,1 D:0]: 1
```

Import the extracted sinogram (e.g. my_prompts_seg0_by_sino.hv) using **AMIDE**.

Select and export the central sinogram plane and upload them online.

Go back to main directory

```
cd ..
```

We can also extract profiles through the sinogram to display these in Excel or similar. You could run

```
./evaluate_simulation_brain.sh
```

to extract the segments and create profiles for you.

Exercise: Data Simulation Thorax

(Always run scripts from the exercises directory)

This is a simple simulation of a thorax phantom (2 gates). Scatter is simulated using STIR. Randoms are constant.

Read and run script:

```
./run_simulations_thorax.sh
```

You will need to extract the sinograms in an “image” Interfile to be able to load them in AMIDE

```
cd working_folder/GATE1/
extract_segments my_prompts_g1.hs
Extract as SegmentByView (0) or BySinogram (1)?[0,1 D:0]: 1

cd ../GATE2/
extract_sinograms my_prompts_g2.hs
Extract as SegmentByView (0) or BySinogram (1)?[0,1 D:0]: 1
cd ..
```

Import the extracted sinogram (e.g. `my_prompts_g1seg0_by_sino.hv`) using **AMIDE**.
Select and export the central sinogram plane and upload them online.

Subtract the two sinograms. This can be done in AMIDE or on the command line

```
stir_subtract -s diff.hs GATE1/my_prompts_g1.hs GATE2/my_prompts2.hs
extract_sinograms diff.hs
Extract as SegmentByView (0) or BySinogram (1)?[0,1 D:0]: 1
```

How does the difference look like in sinogram space?

Go back to main directory

```
cd ..
```

We can also extract profiles through the sinogram to display these in Excel or similar. You could run

```
./evaluate_simulation_thorax.sh
```

to extract the segments and create profiles for you.

For the reconstruction exercise, we first need to generate a new simulation data set. This time just a single slice to speed things up. Scatter is also set to zero for simplicity here.

```
./run_simulation_single_slice.sh
```

Output is in `working_folder/single_slice`

Exercise: Image reconstruction part 1

(Always run scripts from the exercises directory)

We will now look at EMLL, OSEM and OSEM with PSF. A sample script is provided to generate results

```
./run_reconstruction_thorax.sh
```

Or

```
./run_reconstruction_brain.sh
```

This will run EMLL for 240 iterations and OSEM for 240 iterations. It will also continue OSEM from there and write images at every subiterations. Finally, PSF reconstruction will also be performed. The script generates some differences images and launches AMIDE.

You could decide to run only a subset of these exercises and modify the script for yourself.

Sample questions to address:

- Is it worth running EMLL? Why not simply use OSEM?
- Does this depend on the number of iterations that you use? And post-filtering?
- When using PSF reconstruction, can you see effects discussed during the lecture?

Exercise: Image reconstruction part 2 (MAP)

This exercise needs results from the previous step. Output is in `working_folder/single_slice` or `working_folder/brain`

We will now look at OSL and OSSPS with a Quadratic Prior. A sample script is provided to generate results

```
./run_reconstruction_thorax_MAP.sh
```

or

```
./run_reconstruction_brain_MAP.sh
```

This will run OSL and OSSPS (continuing from a previous OSEM image).

Sample questions to address:

- Do OSL and OSSPS generate the same results?
- Does this depend on the penalty factor? Noise level? Iteration number? Initialisation (try to remove the initial estimate for instance).

Exercise: Adding Poisson noise

(advanced exercise)

We can make the simulation more realistic by adding noise to the data. An example would be

```
poisson_noise -p my_noisy_data my_prompts.hs 0.1 1
```

Run `poisson_noise` to understand what these arguments mean. Execute a similar command in `working_folder/brain` to create a noisy simulation. Check if you added “enough” noise (use `extract_segments` to display for instance). Adjust the reconstruction parameter files to use your new noisy data (input) and change the filename used for the output. Run some reconstructions to see what noisy does to your images.

Exercise: Scatter Correction

(Always run scripts from the exercises directory)

Note: this exercise is out of scope for the PET-MR course. Don't try it!

Read and run script, e.g.:

```
N=0; ./run_scatter_${N}.sh
```

Each of the following teams runs the corresponding script:

N = 0 = TEAM # modulo 4

Ideal Scatter correction (using 3 scatter correction loops)

N = 1 = TEAM # modulo 4

Calculate scatter by using a smaller energy window than that simulated. This will demonstrate if the scaling technique works. We have selected 425keV.

N = 2 = TEAM # modulo 4

Scatter correction using the scatter estimation from the first gate for both gates. This will demonstrate how sensitive is scatter in choosing different but adjacent gates.

N = 3 = TEAM # modulo 4

Perform reconstruction & attenuation correction by using wrong attenuation map. In the particular exercise we have assigned bone attenuation value to lung attenuation value for the first gate. Then we use this wrong attenuation map located at the first gate to correct for attenuation and scatter for each gates.

Displaying different results per group:

M = 0 = TEAM # modulo 2

Reconstruct the scatter corrected activity with FBP for each of the two gates:

```
cd ./working_folder/GATE1/
```

```
FBP2D FBP2D_full.par
```

```
cd ../GATE2/
```

```
FBP2D FBP2D_full.par
```

```
cd ../..
```

Use **AMIDE** to visualize your `scatter_corrected_image.hv` for each gate and the original simulated image, then select and export the central transverse, coronal or sagittal view and upload them online. Use maximum display value 25.

Can you display the subtraction (e.g. using `stir_subtract`) between the two gates? How much motion do you see?

M = 1 = TEAM # modulo 2

Extract sinograms for display with AMIDE for each of the two gates, e.g.:

```
extract_sinograms scatter_estimate_g2_3.hs
```

```
extract_sinograms my_scatter_g2.hs
```

Select and export the central sinogram plane for those and upload them online.

Can you display the difference of the two sinograms e.g. using `stir_subtract -s`

Exercise: Motion Correction

(Always run scripts from the exercises directory. This exercise depends on the output of `run_simulations_thorax.sh`)

There are 2 scripts:

```
run_MCIR_0.sh
```

Correct for motion using valid motion vectors and the previously calculated scatter background

```
run_MCIR_1.sh
Do not correct for motion
```

Read and run scripts, e.g.:

```
./run_MCIR_0.sh &
./run_MCIR_1.sh &
```

This will run everything in the “background”, i.e. you will be immediately have the prompt back. It takes about 30seconds to complete each reconstruction. If nothing happens, you can confirm the scripts run OK:

```
less working_folder/MCIR/MCIR.log
less working_folder/noMC/noMC.log
```

(quit less by pressing q)

Each of the following teams runs the corresponding script:

Each team uses AMIDE to visualize the corresponding images (e.g. working_folder/MCIR/MCIR_16.hv or a later iteration 32, 48, 64). Choose the iteration depending on the computational speed of the computer. Select a transverse slice showing the myocardium (preferably z=12.15mm). Maximum scale value: 25.

Appendices

Installing STIR

You will need the updated source made available via the web for this course (see the STIR website). You also need a display program, we recommend AMIDE. You will then need to unpack the zip file and add the directory to your path such that they can be found when typing a command.

If you have Linux (or vGate), use the installation script. Otherwise you need to build STIR yourself, install it, and add it to your path

Open a terminal and type something like this all on one line (adjust to where your files are):

- Linux/MacOS:
PATH=~ /STIRShortCourse/bin/:\$PATH
- Cygwin
PATH=~ /STIRShortCourse/bin/:/cygdrive/c/Program Files\ (x86\)/amide/bin/:\$PATH
- Windows command terminal (if Cygwin not available)
PATH %HOMEDRIVE%%HOMEPATH%\Documents\STIRShortCourse\bin;c:\Program Files x86\amide\bin;%PATH%

After installing, try to type in your terminal

```
forward_project
```

You should see a usage message. If you get an error, you probably didn't set-up your path correctly.

File extensions

.hv: Interfile header for image (volume)
.ahv: (ignore) old-style Interfile header for image
.v: raw data of image (in floats)

.hs: Interfile header projection data (sinograms)
.s: raw data of projection data (in floats)

.par: STIR parameter file

.sh: Shell script (sequence of commands)

.bat: Windows batch file

.log: log file (used to record output of command)

Commands for evaluation:

The following is a list of commands that are used during the exercises

```
extract_segments projdata.hs
```

convert projection data into an (Interfile) image e.g. for display

```
list_projdata_info projdata.hs
```

Writes basic information about geometry etc.
Use without arguments for extra options.

```
list_image_info image.hv
```

Writes basic information about image geometry and values.

```
list_image_values prof.txt input_image \  
    min_plane max_plane  min_row max_row min_col max_col
```

(note: the backslash “\” is used in shell scripts for “line continuation”, i.e. when everything does not fit on one line)

list_image_values writes values to a text file (for import in Excel et al).
Indices need to be in the STIR convention (plane starts from 0, col,row are centred around 0). Use list_image_info to find ranges.

Note: there is currently a bug in list_image_values that row (x) and column (y) have to be given in that order (i.e. it's z,x,y while should have been z,y,x)

Image display

Several display programs can be used. AMIDE reads the interfile volumes directly. ImageJ and others can use import of raw floats (i.e. the .v file). Settings are for instance.

Image type: 32-bit Real

Width ?

Height: ?

Offset: 0

Number of images ?

Gap between images: 0

White is 0: Ticked

Little endian: Ticked

You will have to find the data sizes from the header (the .hv file), or by using list_image_info.