Imaging applications

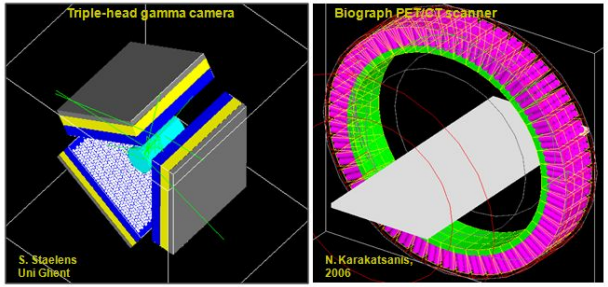
Maxime Chauvin on behalf of the OpenGATE collaboration
maxime.chauvin@inserm.fr

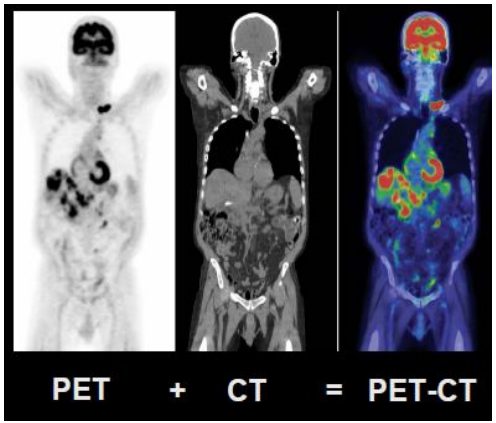
Rencontres du Vietnam, Medical Physics School - Quy Nhon, Vietnam - 2019

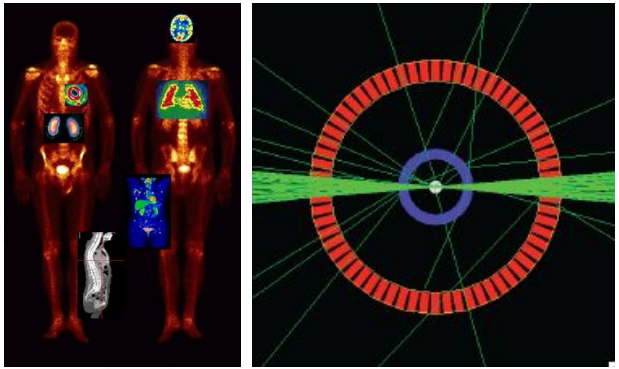
Introduction

Nuclear and radiation imaging modalities:

- PET
- SPECT
- Compton Camera
- Prompt gamma
- CT
- CBCT …







Simulation components

- Complex **imaging systems**: PET, SPECT, Compton Camera, CT, CBCT
  - Clinical, pre-clinical
  - Crystal, digitizer, dead time, collimator
  - Rotation, module duplication, timing
- **Patient geometry** description
  - Voxelized phantom
  - Stoichiometric calibration from CT HU
- **Patient source** description
  - Activity spatial distribution as voxelized source
  - From SPECT, PET images

Complex Imaging Systems

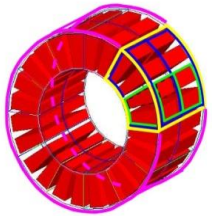
Concept of System

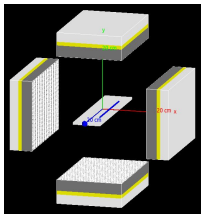
- A system is a key-concept of GATE
- Provides a template of a predefined geometry to simulate a scanner
  - described by key components
  - organized in a certain way, called a tree level structure
  - each component has its own specific role or ordering
- Can be used to model several scanners sharing the same general geometrical characteristics

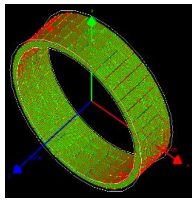
http://wiki.opengatecollaboration.org/index.php/Users_Guide:Defining_a_system

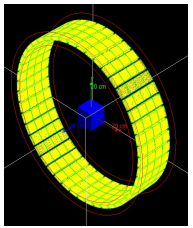
List of available systems

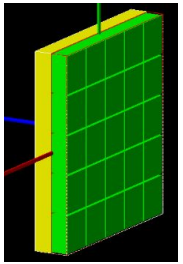
Scanner	SPECThead	CTscanner	CylindricalPET	Ecat	OpticalSystem
scanner	SPECThead	CTscanner	CylindricalPET	Ecat	OpticalSystem
level1	crystal	module	rsector	block	crystal
level2	pixel	cluster	module	crystal	pixel
level3		pixel	crystal		
level4			layer0		
level5			layer1		





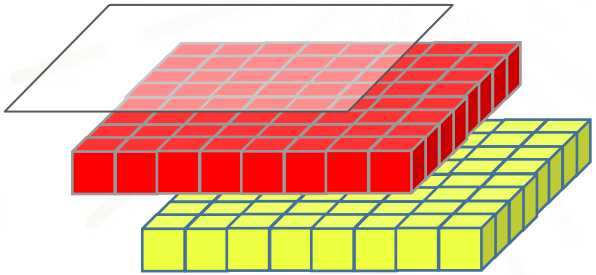


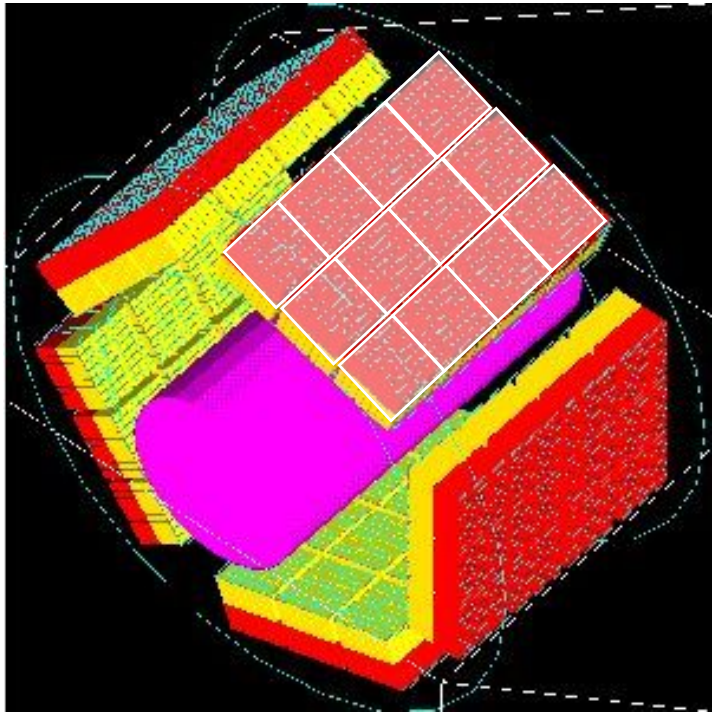


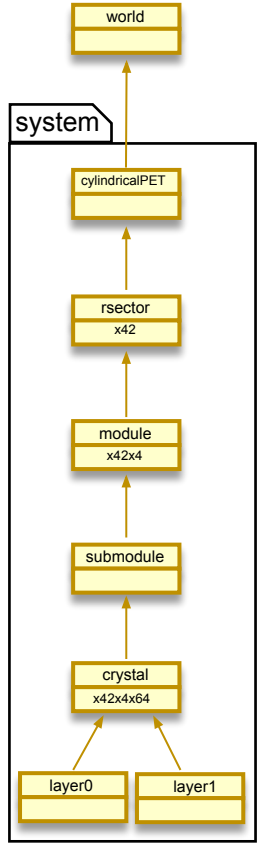


Example of cylindricalPET system

- 5 rsectors
- 4 modules (repeated along Z axis),
- 3 submodules (repeated along Y axis),
- 64 crystals (8 x 8)
- 2 layers (red and yellow)







Defining a System

1. The first volume (after the World) to be created will encapsulate the whole system :

```
/gate/world/daughters/name                SPECThead
/gate/world/daughters/insert              box
/gate/SPECThead/geometry/setXLength       6.5 cm
/gate/SPECThead/geometry/setYLength       19 cm
/gate/SPECThead/geometry/setZLength       28 cm
/gate/SPECThead/setMaterial               Air
```

! Hard-coded name !

From the list of available systems:
scanner - SPECTHead - cylindricalPET - ecat - CPET - OPET - OpticalSystem

Defining a System

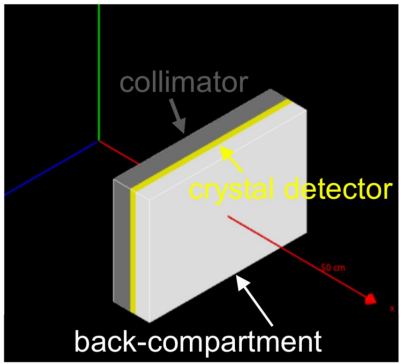
2. Create a crystal volume as daughter of the SPECThead system (only one block of NaI, the pixelization will be provided by the output module)

```
/gate/SPECThead/daughters/name                    crystal_scintillator
/gate/SPECThead/daughters/insert                  box
/gate/crystal_scintillator/geometry/setXLength    1 cm
/gate/crystal_scintillator/geometry/setYLength    19 cm
/gate/crystal_scintillator/geometry/setZLength    28 cm
/gate/crystal_scintillator/setMaterial            NaI
```

3. Create other volumes if needed :

```
/gate/SPECThead/daughters/name              collimator
/gate/SPECThead/daughters/insert            box
...
/gate/collimator/setMaterial                Lead
```

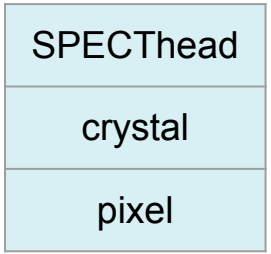
```
/gate/SPECThead/daughters/name       back-compartment
/gate/SPECThead/daughters/insert     box
...
/gate/back-compartment/setMaterial   Lead
```



Defining a System

4. Add each geometric component with respect to the tree level structure
   → attach each volume (crystal_scintillator) to its corresponding component in the chosen system (crystal for SPECThead)

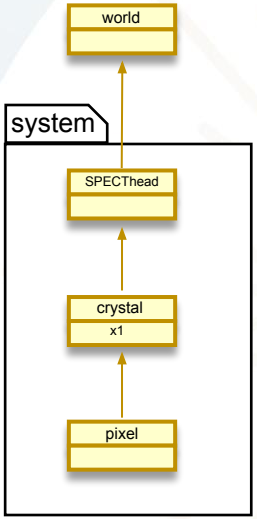
```
/gate/systems/SPECThead/crystal/attach    crystal_scintillator
```



5. Declare one given volume as a sensitive detector
   → all interactions occurring inside this volume are recorded and could be further analyzed

```
/gate/crystal_scintallator/attachCrystalSD
```

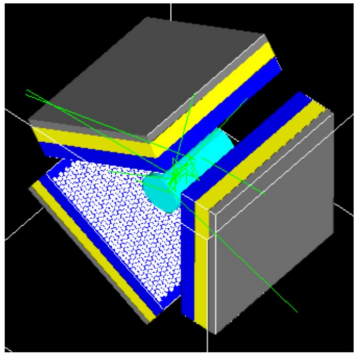
Defining a SPECT System




```
# 1. Create the encompassing volume
/gate/world/daughters/name SPECThead
/gate/world/daughters/insert box
...
# 2. Repeat the head to get several ones
/gate/SPECThead/repeaters/insert ring
/gate/SPECThead/ring/setRepeatNumber 3
...
# 3. Create the crystal volume
/gate/SPECThead/daughters/name my_crystal
/gate/SPECThead/daughters/insert box
...
# 4. Create the collimator
/gate/SPECThead/daughters/name my_collimator
/gate/SPECThead/daughters/insert box
...
# 5. Attach to components (here only the crystal)
/gate/systems/SPECThead/crystal/attach my_crystal
...
# 6. Attach to sensitive detector
/gate/my_crystal/attachCrystalSD
```

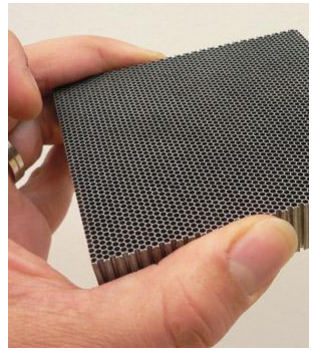
Create and repeat volumes

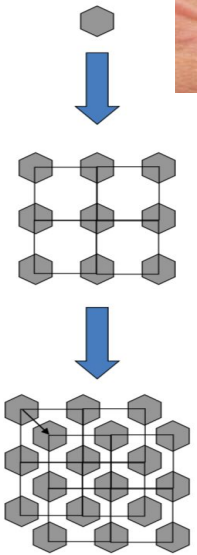
Attach system components and sensitive detectors



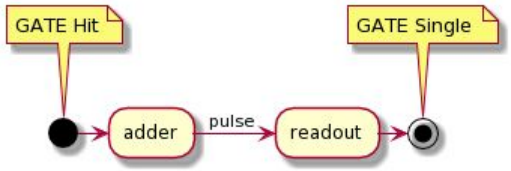
Modeling a Collimator

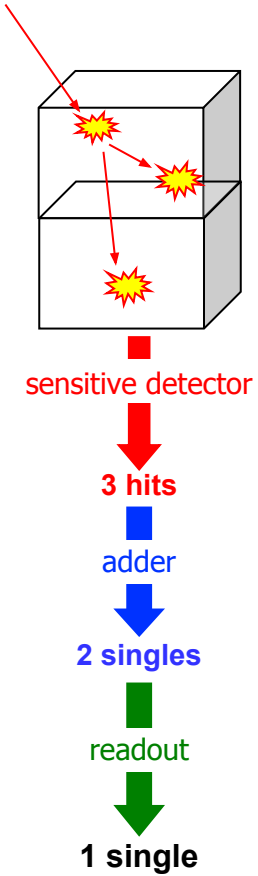
```
# 4. Create the collimator
/gate/SPECThead/daughters/name              my_collimator
/gate/SPECThead/daughters/insert            box
...

# Insert the first hole of air in the collimator
/gate/my_collimator/daughters/name          hole
/gate/my_collimator/daughters/insert        hexagone
/gate/hole/geometry/setHeight               2.7 cm
/gate/hole/geometry/setRadius               0.122 cm

# Repeat the hole in an array
/gate/hole/repeaters/insert                 cubicArray
/gate/hole/cubicArray/setRepeatNumberX      69
/gate/hole/cubicArray/setRepeatNumberY      84
/gate/hole/cubicArray/setRepeatNumberZ      1
/gate/hole/cubicArray/setRepeatVector       0.774 0.447 0. cm

# Repeat linearly these holes
/gate/hole/repeaters/insert                 linear
/gate/hole/linear/setRepeatNumber           2
/gate/hole/linear/setRepeatVector           0.387 0.2235 0. cm
```





Digitizer





➔ Adder

- Sum all the hits that occur within the same crystal
- Should be the first module of a digitizer chain
- Regroups **hits** per volume into a pulse

```
/gate/digitizer/Singles/insert      adder
```

★ Energy: total energy in volume
★ Position: energy-weighted centroid

➔ Readout

- Artificial geometry to mimic the fact that a detector can measure pulse from different volumes
- Regroups pulses per block (group of sensitive detectors)
- Block is defined by the depth parameter

```
/gate/digitizer/Singles/insert              readout
/gate/digitizer/Singles/readout/setDepth     1
```

★ Energy: total energy in block
★ Position: position of pulse with maximum energy (winner-takes-all)

Other digitizer modules

- **Blurring** modules to apply a blur on:
  - the deposited energy (Gaussian blurring of the E spectrum)

```
/gate/digitizer/Singles/insert blurring
/gate/digitizer/Singles/blurring/setResolution 0.10
/gate/digitizer/Singles/blurring/setEnergyOfReference 140. keV
```

  - the detected position (to model the uncertainty on detected position)

```
/gate/digitizer/Singles/insert sblurring
/gate/digitizer/Singles/setSpresolution 2.0 mm
```

- Module to apply **threshold** on the detected energy

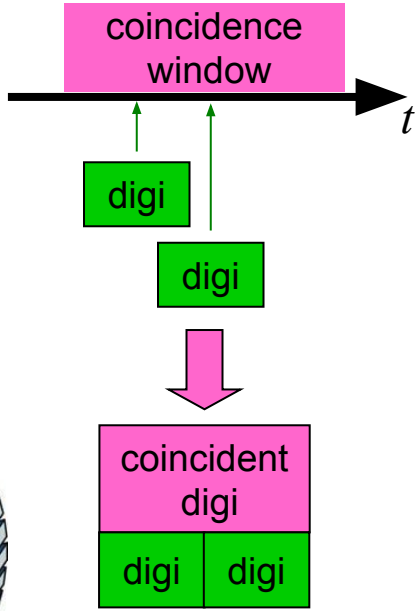
```
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold 80. keV
```

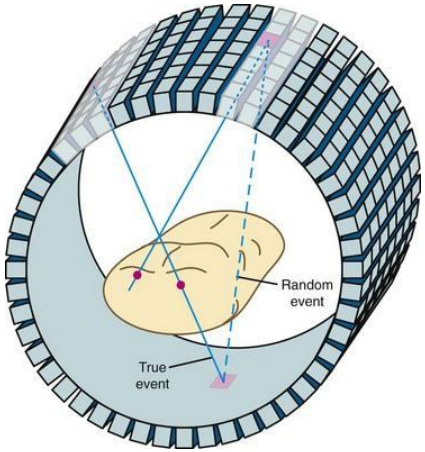
- Module to apply **uphold** on the detected energy

```
/gate/digitizer/Singles/insert upholder
/gate/digitizer/Singles/upholder/setUphold 160. keV
```

Coincidence sorting

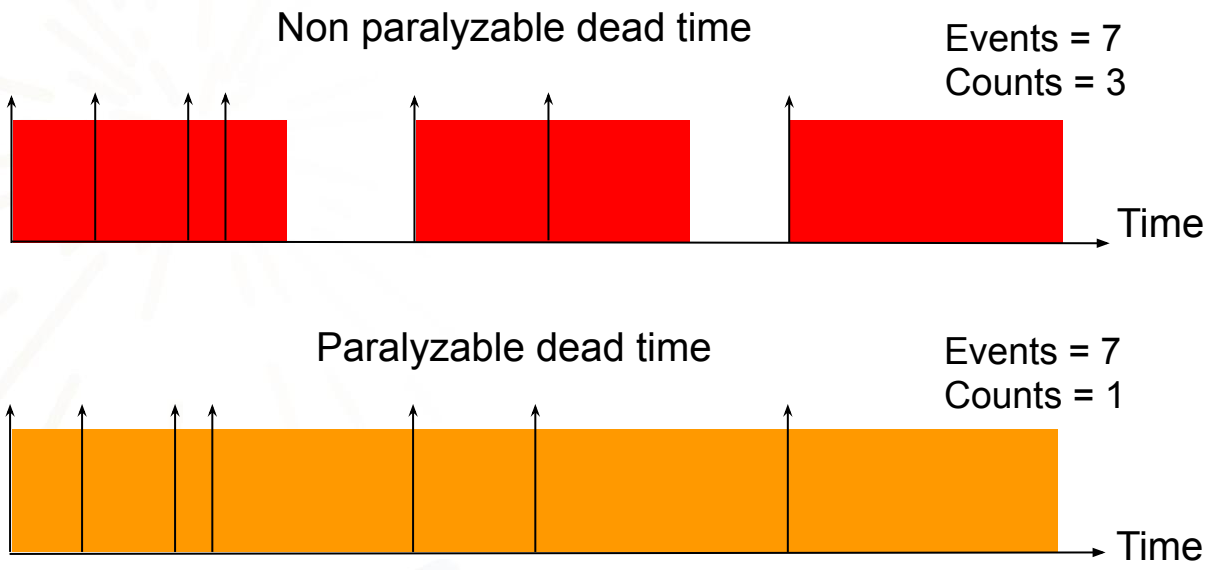
- For PET systems: sort the *Singles* list for coincidences
- Principle
  - The coincidence sorter processes the *Singles* list…
  - Looks for coincidences by searching in this list for a set of *Singles* that are detected within a given time interval (the so called 'coincident events')
  - … and creates *coincident events*
- Can store ‘random’ coincidences for analysis





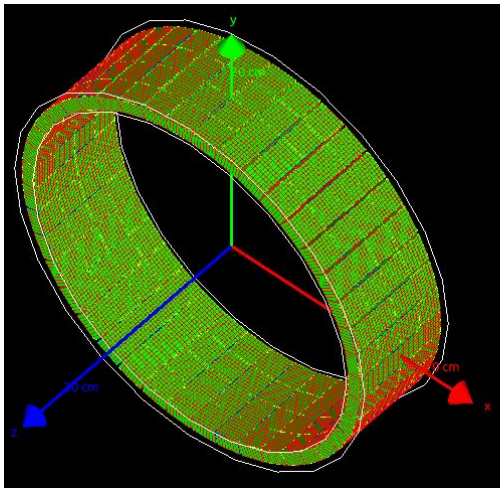
Dead Time Module

- Dead time is applied on the Singles, before the coincidence sorting
- Two dead time models are available : paralyzable and non-paralyzable



Digitizer example

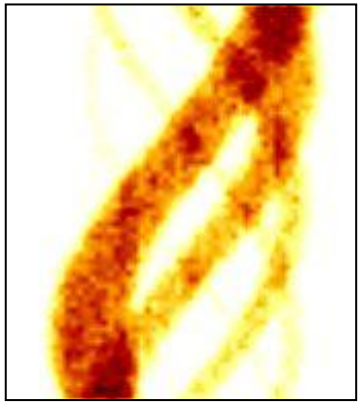
```
# ADDER
/gate/digitizer/Singles/insert adder

# READOUT
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 1
/gate/digitizer/Singles/insert blurring
/gate/digitizer/Singles/blurring/setResolution 0.26
/gate/digitizer/Singles/blurring/setEnergyOfReference 511. keV

# ENERGY CUT
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold 250. keV
/gate/digitizer/Singles/insert upholder
/gate/digitizer/Singles/upholder/setUphold 750. keV

# DEAD TIME
/gate/digitizer/Singles/insert deadtime
/gate/digitizer/Singles/deadtime/setDeadTime 1000. ns
/gate/digitizer/Singles/deadtime/setMode nonparalysable
/gate/digitizer/Singles/deadtime/chooseDTVolume block

# COINCIDENCE SORTER
/gate/digitizer/Coincidences/setWindow 10. ns
```



Simulation output

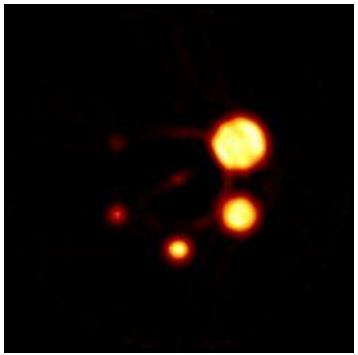
- PET: sinogram

```
/gate/output/sinogram/enable
/gate/output/sinogram/setFileName          my_sinogram.root
```



- SPECT: projection

```
/gate/output/projection/enable
/gate/output/projection/setFileName        my_projection.mhd
```

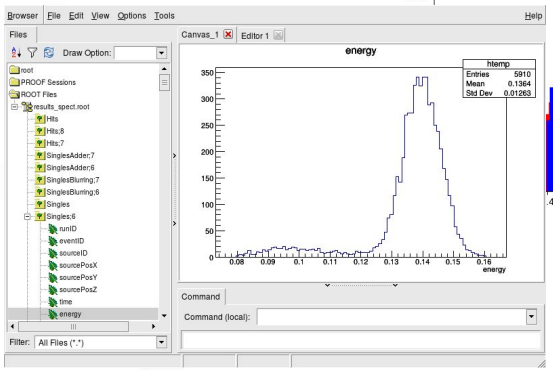


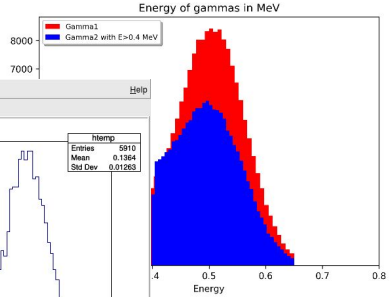
Simulation output

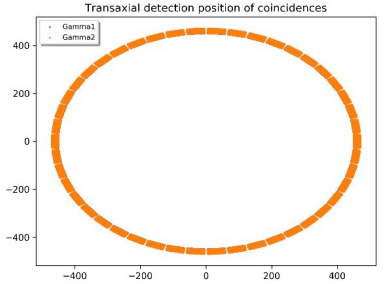
- Store all hits, singles, events for data analysis

```
# ROOT OUTPUT FORMAT
/gate/output/root/enable
/gate/output/root/setFileName                my_output.root
/gate/output/root/setOutFileHitsFlag         1
/gate/output/root/setRootSinglesFlag         1
/gate/output/root/setRootCoincidencesFlag    1
```

- Organized as trees
  - *Hits*
  - *Singles*
  - Coincidences
  - …







ROOT Data Analysis Framework

python™

Patient geometry description

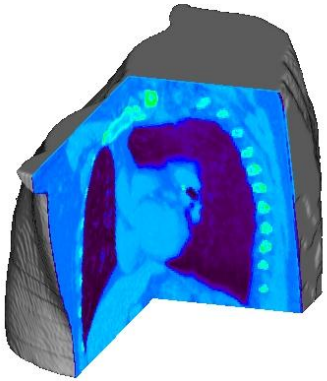
Definition of Voxel Geometries

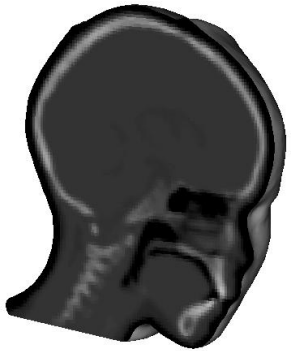
➔ Using digital phantoms or patient CT images

```
/gate/world/daughters/name                    patient
/gate/world/daughters/insert                  ImageNestedParametrisedVolume
/gate/patient/geometry/SetImage               data/patient_CT.mhd
/gate/patient/placement/setTranslation        0 0 0 mm
```

➔ GATE converts pixel or HU values to the Geant4 material composition using:

- a range translator
- a HU convertor





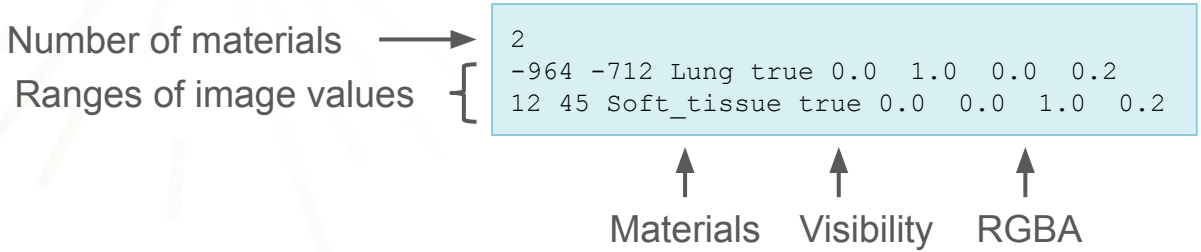
Definition of Voxel Geometries

- Using a range translator:

Text file with a predefined format converting pixel range values into materials

```
/gate/world/daughters/name                          patient
/gate/world/daughters/insert                        ImageNestedParametrisedVolume
/gate/patient/geometry/SetImage                     data/patient_CT.mhd
/gate/patient/geometry/setRangeToMaterialFile       data/range.dat
```

Range.dat:



Definition of Voxel Geometries

- Using a HU convertor:

```
/gate/HounsfieldMaterialGenerator/SetMaterialTable                    data/Schneider2000MaterialsTable.txt
/gate/HounsfieldMaterialGenerator/SetDensityTable                     data/Schneider2000DensitiesTable.txt
/gate/HounsfieldMaterialGenerator/SetDensityTolerance                 0.1 g/cm3
/gate/HounsfieldMaterialGenerator/SetOutputMaterialDatabaseFilename   data/patient-HUmaterials.db
/gate/HounsfieldMaterialGenerator/SetOutputHUMaterialFilename         data/patient-HU2mat.txt
/gate/HounsfieldMaterialGenerator/Generate
```

➔ **Schneider2000MaterialsTable.txt** calibration file to split the HU range into several materials (see [Schneider2000]).
➔ **Schneider2000DensitiesTable.txt** calibration file to convert HU and to mass density (g/cm3).
➔ **SetDensityTolerance** define the density tolerance to avoid generating one material for each HU value.

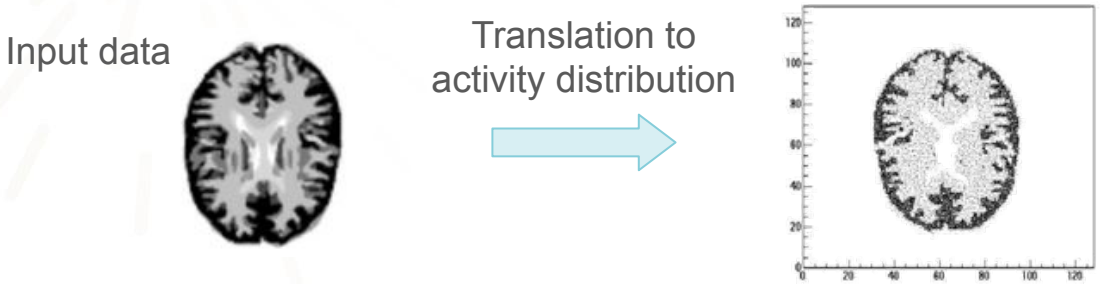
➔ **patient-HUmaterials.db** and **patient-HU2mat.txt** generated are used for the patient geometry:

```
/gate/geometry/setMaterialDatabase                  data/patient-HUmaterials.db
/gate/patient/geometry/setHUToMaterialFile          data/patient-HU2mat.dat
```

Patient source description

Definition of Voxelized Sources

- Digital phantom or patient data can be read as emission distribution (3D image in Interfile, Analyze, MetaImage or DICOM format)
- Each voxel grayscale value of the image is converted into activity using :
  - a range translation table (for digital phantom)
  - a linear scale (for patient data)
- Each voxel is then treated as a source with a specific activity

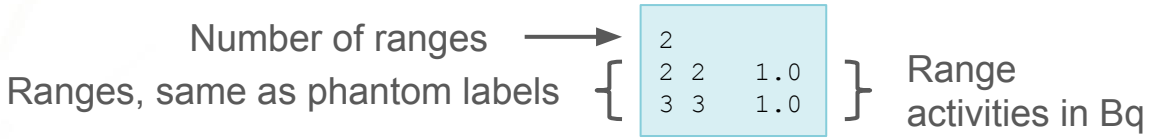


Emission map from digital Hoffman brain phantom

Definition of Voxelized Sources

- Using a range translator:

ActivityRange.dat:




```
# Voxelized Source Based On The Contrast-Detail Phantom
/gate/source/addSource                                  source voxel
/gate/source/source/reader/insert                       image

# Insert The Translator That Will Convert The Image From Digital Values To Activity
Values
/gate/source/source/imageReader/translator/insert       range
/gate/source/source/imageReader/rangeTranslator/readTable  data/ActivityRange.dat
/gate/source/source/imageReader/rangeTranslator/describe   1

# Read Image
/gate/source/source/imageReader/readFile                data/phantom.mhd

# Position Of The Source (its default position is in the 1st quarter, so it has to be
shifted over half its dimensions in the negative direction on each axis)
/gate/source/source/setPosition                         -100 -15 -75 mm
```

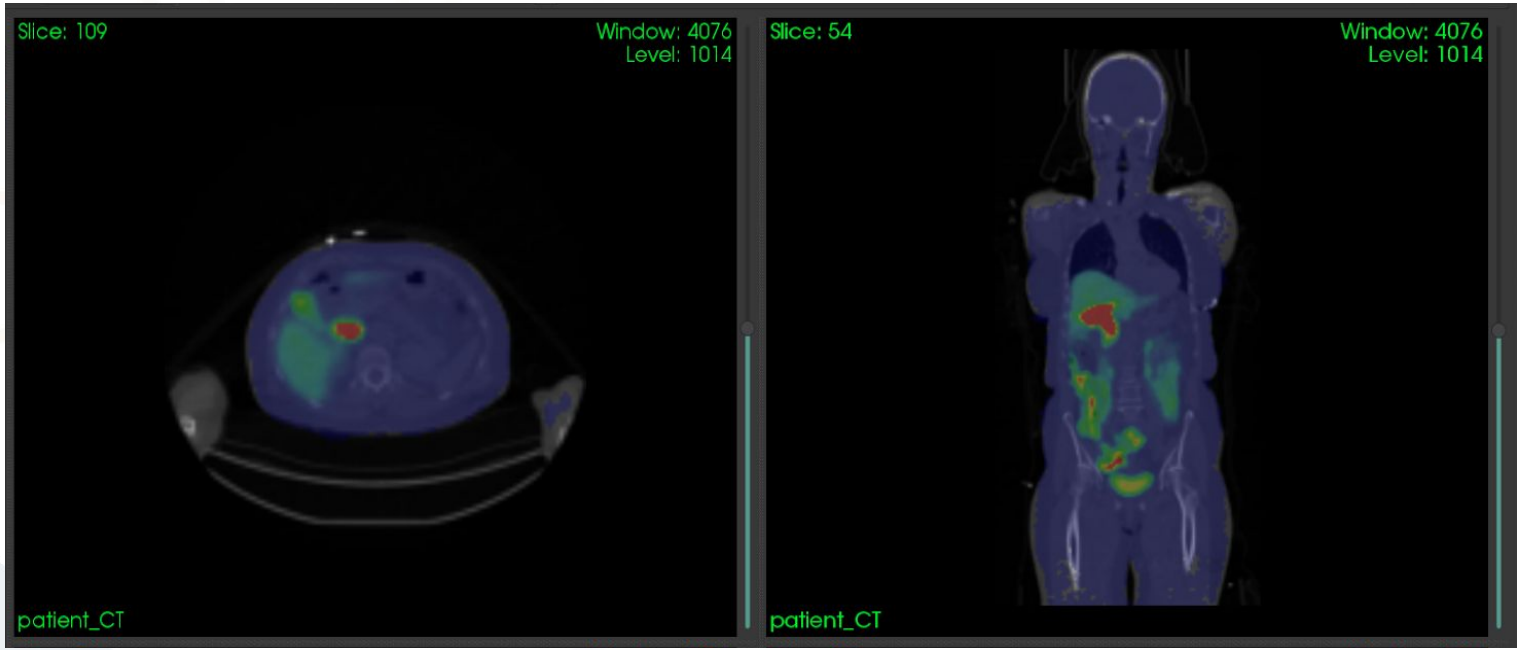
Definition of Voxelized Sources

- Using a linear scale:

```
# Voxelized Source Based On The Contrast-Detail Phantom
/gate/source/addSource                                    source voxel
/gate/source/source/reader/insert                         image

# Insert The Translator To Convert The Image From Digital Values To Activity Values
/gate/source/source/imageReader/translator/insert         linear
/gate/source/source/imageReader/linearTranslator/setScale 1.0 Bq

# Read Image
# SPECT image activity: max = 65535, total = 1512387046
/gate/source/source/imageReader/readFile                  data/patient_SPECT.mhd

# Position Of The Source (its default position is in the 1st quarter, so it has to be shifted
over half its dimensions in the negative direction on each axis)
/gate/source/source/setPosition                           -243.1 -243.1 -486.2 mm

# Technetium-99m, one photopeak : 140 keV
/gate/source/source/gps/particle                          gamma
/gate/source/source/gps/energytype                        UserSpectrum
/gate/source/source/gps/setSpectrumFile                   data/energy_spectrum_Tc99m.txt
/gate/source/source/gps/angtype                           iso
```

Definition of Voxelized Sources

- Using a linear scale:



Hands-on

Hands-on: SPECT imaging

In the folder school_vietnam/Ex2-SPECT you will find a simulation of a SPECT. Warning: this simulation needs a very long computation time to reach acceptable statistics. Some variance reduction techniques exist, but are not shown here. In the different macro files you will find some parts missing that you have to complete:

- spect_head.mac:
  - insert a ring repeater to have 2 heads
- spect_digitizer_Tc99m.mac:
  - set a thresholder at 126 keV and upholder at 154 keV to the digitizer/peak140
- source_voxels.mac:
  - insert a linear translator for the patient image activity and set the scale to 1e-6 Bq
- main.mac:
  - add the HounsfieldMaterialGenerator to create the files patient-HUmaterials.db and patient-HU2mat.txt

- Save all the macro files and execute the simulation with **`Gate --qt mac/main.mac`**
- Look at the folder output to see what files have been produced
- Visualise the results with **`vv output/projection.mhd`**. What do you see ?
- You can use ROOT and the Tree Browser to open output/spect.root and see all the hits and singles
- Increase the activity of the source