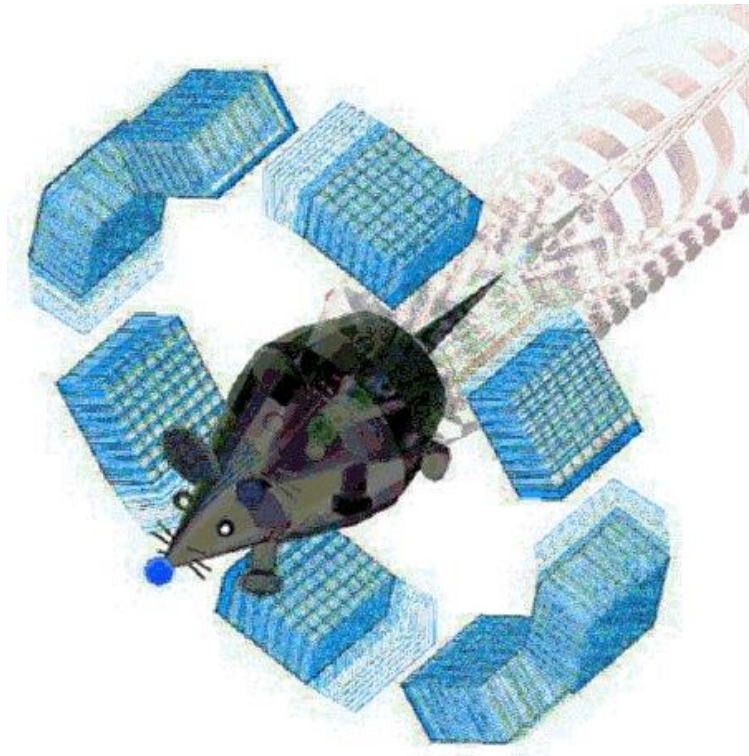


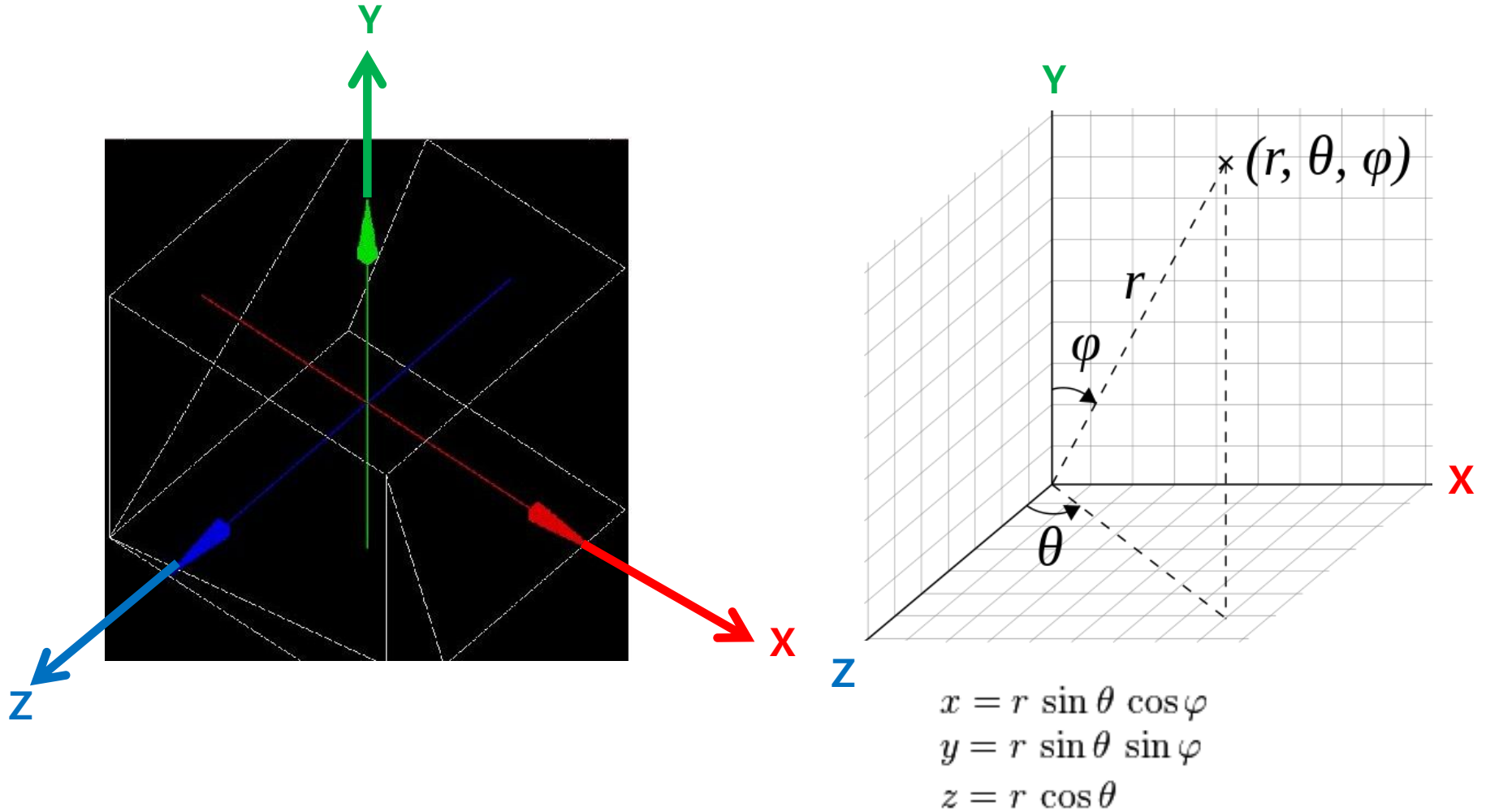
# GATEv6.2 NCAT 4D Phantom simulation with ECAT7 PET



2015.06.23.  
강한규

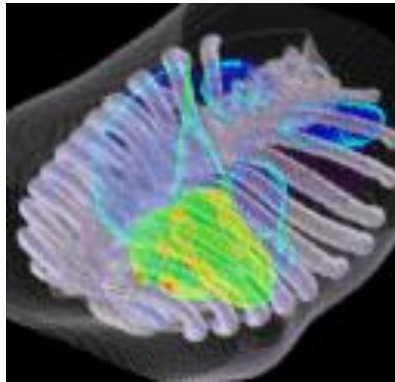
<http://wiki.opengatecollaboration.org/UsersGuide>

# Spherical coordinates (구좌표계) in GATE



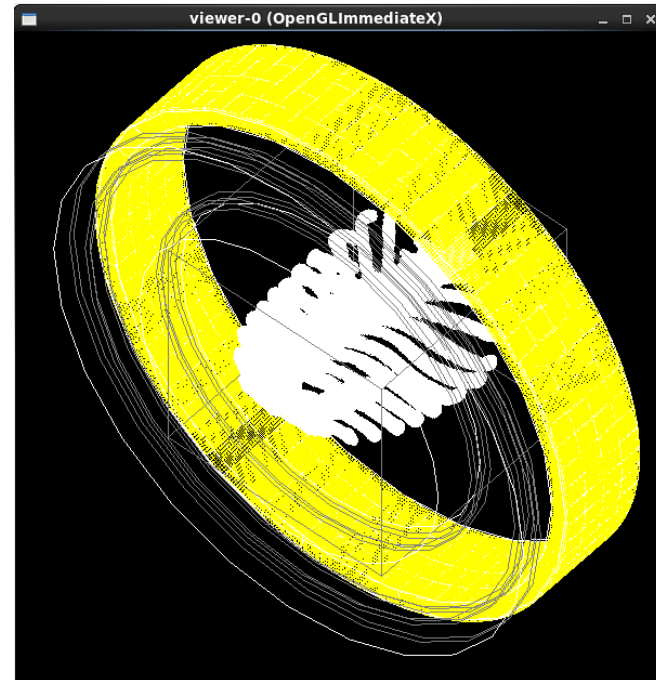
# GATEv6.2 예제 파일 "example\_TimeActivityCurve" 시뮬레이션 방법

NCAT 4D Phantom  
(AMIDE view)



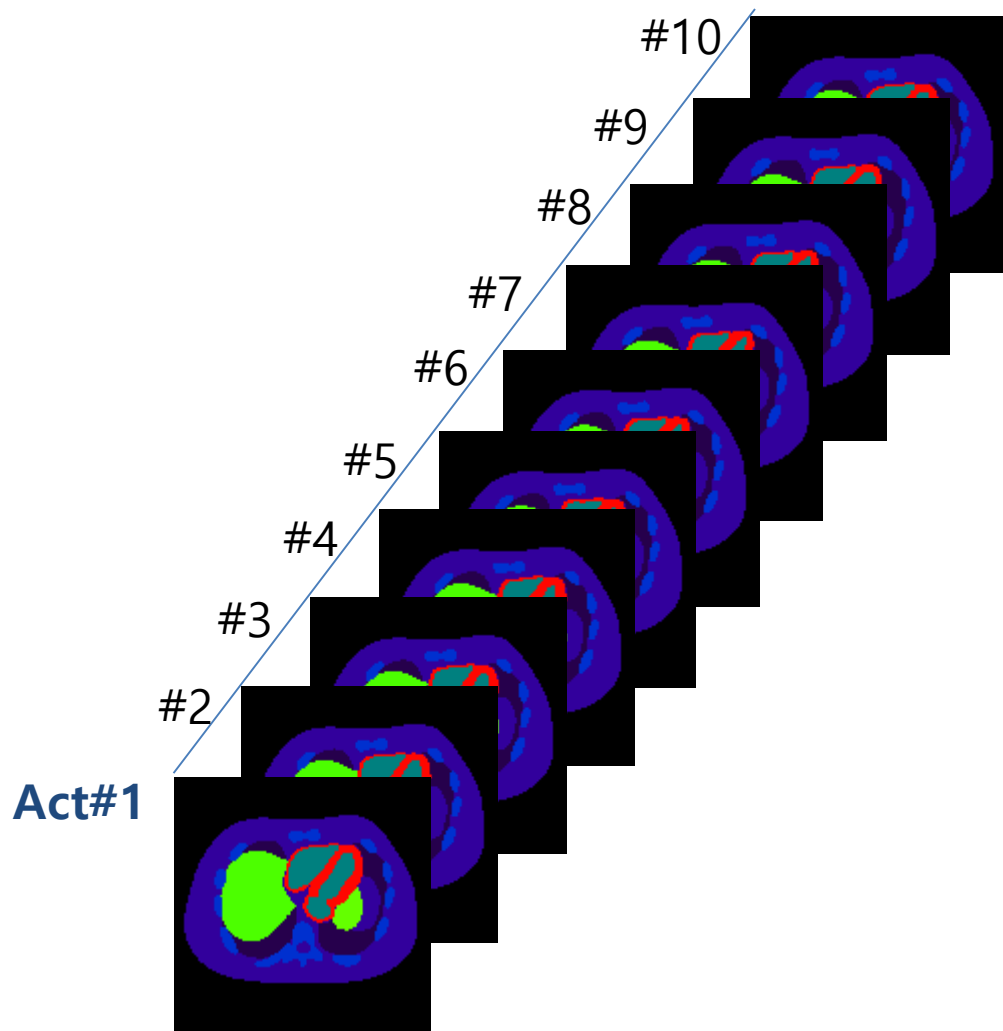
+

PET\_ECAT7\_System  
(OpenGL view)

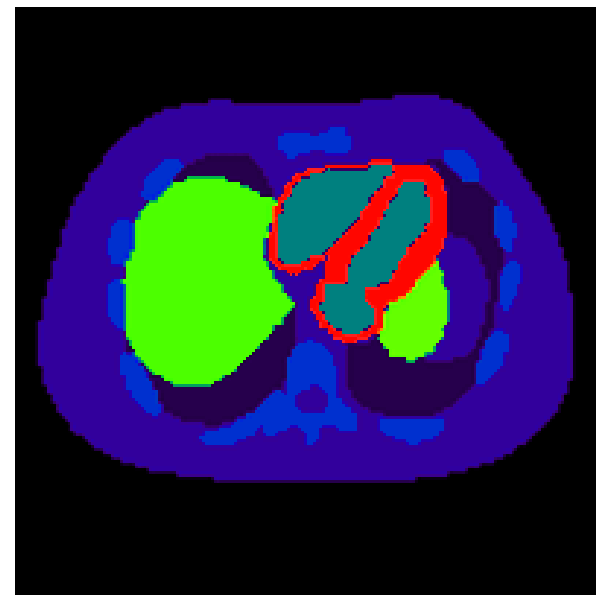


"example\_TimeActivityCurve" 예제파일은 NCAT 4D Phantom CT영상 (voxelized-phantom)에 PET영상(voxelized-source)을 F-18로 이용해서 DoseMap을 AMIDE로 보여줌.

# NCAT 4D Phantom (Activity map) GIF animation

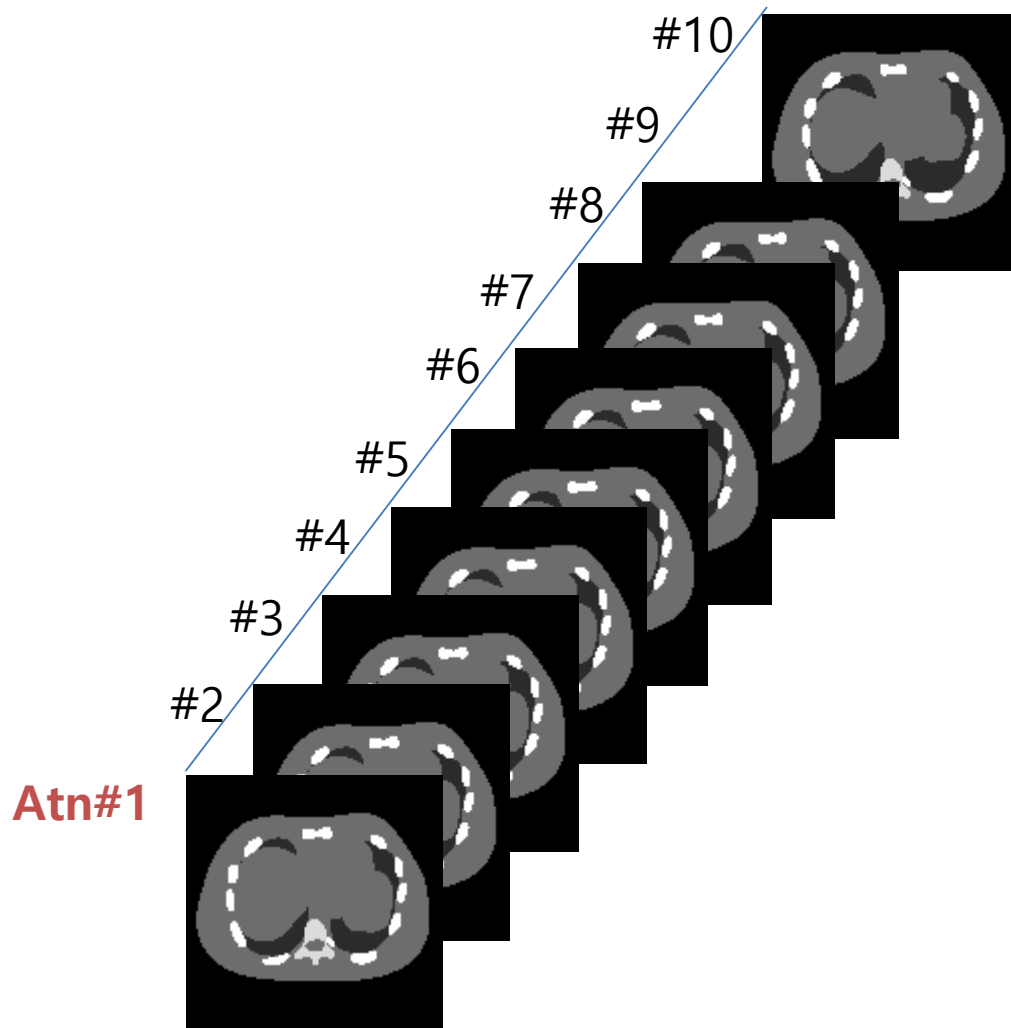


Activity map (PET)



0.1 초/frame

# NCAT 4D Phantom (**Attenuation map**) GIF animation



**Attenuation map (CT)**



0.1 초/frame

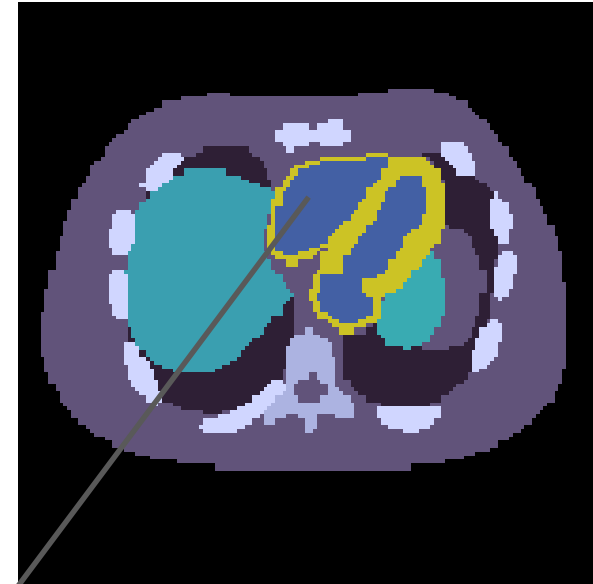
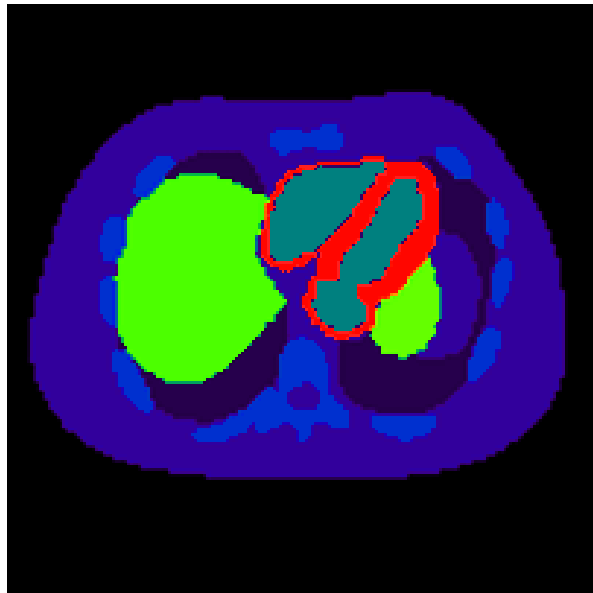
# Activity map (PET) vs. Attenuation map (CT)

NCAT으로 Real-time motion  
시뮬레이션 가능

Activity map (PET)

Attenuation map (CT)

Act + Atn fusion



0.1 초/frame

특정 voxel장기에  
Time Activity Curve(TAC)를  
넣을 수 있음.

# NCAT 4D Phantom 시뮬레이션 과정: 8 steps

- 1 – Verbosity and Visualization (설정값 출력 및 시뮬레이션 화면)
- 2 – Define the NCAT 4D Phantom geometry (환자의 4D CT 영상으로 모션 반영)
- 3 – Physics processes (physics list설정)
- 4 – Initialization the simulation : Compute cross-section tables for particles,  
Initialization 이후엔 매크로에서 1,2,3의 설정 값을 바꾸지 못함.
- 5 – Source (Ncat voxelized source : 특정 voxel 장기에 Time Activity Curve 반영)
- 6 – Digitizer(PET)
- 7 – Output setup (ROOT, RawSinogram, ECAT7 파일 출력 설정)
- 8 – Start the simulation(시뮬레이션 시작)

# GATE6.2 NCAT 4D Phantom 매크로 파일의 구성

ROOT\_to\_Sinogram\_FBP\_Recon → ROOT -> Sinogram 변환 후 매트랩 코드로 FBP recon

MATLAB\_RawSinogram\_Recon\_Code → 매트랩 코드 RawSinogram output FBP recon

Root\_output → Root output (.root)

RawSinogram\_output → RawSinogram output(.sin, .info, .dim)

ECAT7\_output → ECAT7 sinogram output(.S)

data → {  
✓ Voxelized PET activity source  
✓ Voxelized CT attenuation phantom

GateMaterials.db → Water, PMMA, Lung 등의 물질정보 보관

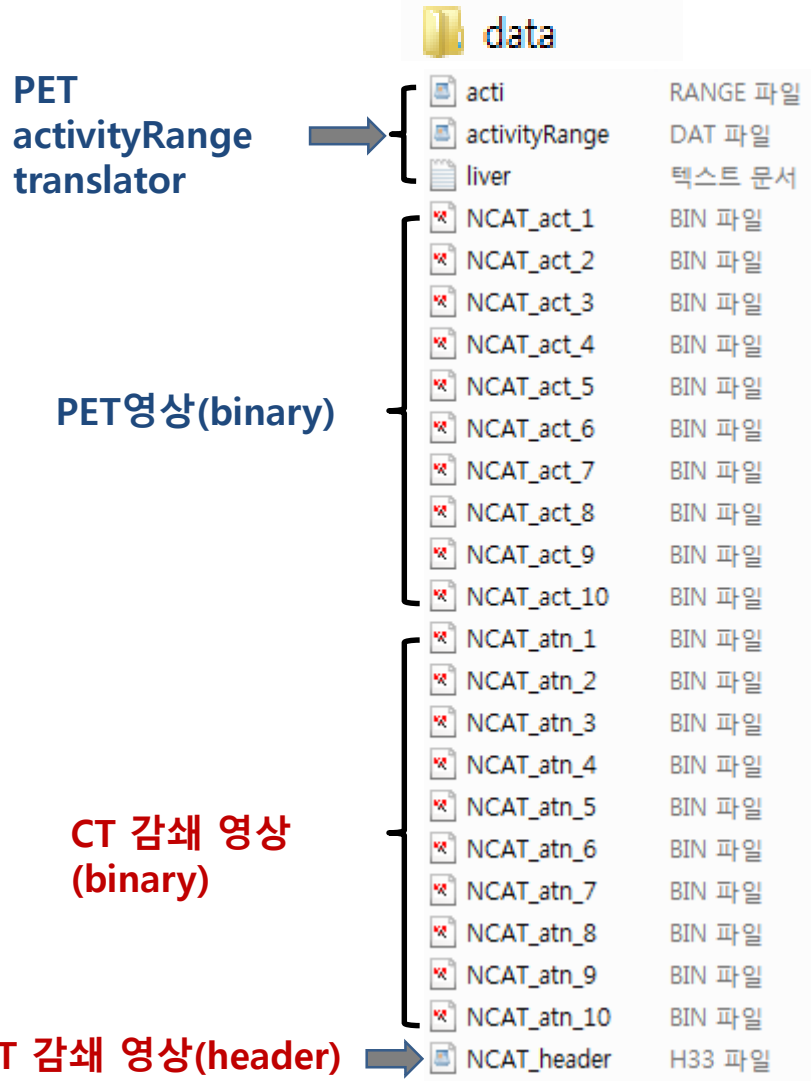
vis → OpenGL viewer의 각도, 줌 설정

Main\_RTVPhantom\_NCAT4D\_PET\_ECAT7\_KangHG → Main Macro

- Verbosity 출력 레벨 설정
- PET ECAT7 system 설정
- NCAT 4D Phantom 설정
  - ✓ Voxelized PET activity source
  - ✓ Voxelized CT attenuation phantom
- Physics list
- Output 설정



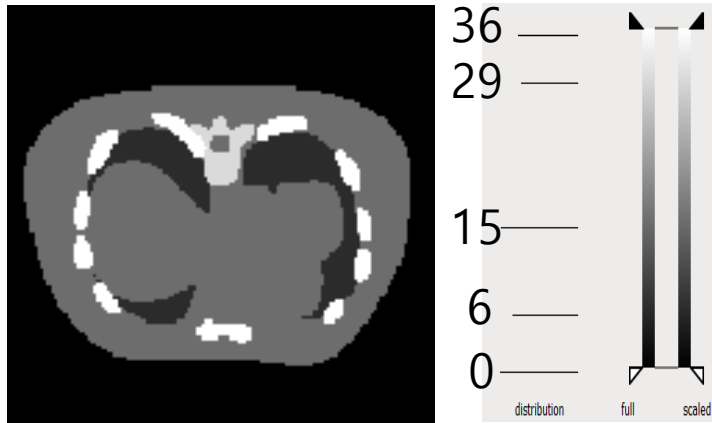
# data폴더의 내용(1)



CT 감쇄 영상 밝기 ->Material (rangeTranslator) → range DAT 파일



# range.dat (CT image->Material)



range DAT 파일

4									
0	0	Air	false	0.0	0.0	0.0	0.0	1.0	
6	6	Lung	false	0.8	0.8	0.8	1.0	1.0	
15	15	Breast	false	0.5	0.5	0.5	1.0	1.0	
29	36	RibBone	true	1.0	1.0	1.0	1.0	1.0	



# 1. Visualisation, Verbosity 설정

```
# Main_RTVPhantom_NCAT4D_PET_ECAT7_KangHG.mac
# First Modified: 2015.04.14 Kang Han-Gyu
(hangyookang@gmail.com)
# Last Modified : 2015.05.28 Kang Han-Gyu
(hangyookang@gmail.com)
# 4D phantom was moved -40 mm along the z-dir.
# NCAT 4D phantom simulation with ECAT7 PET
```

```
#=====
# VISUALISATION
#=====
```

`/vis/disable` ➡ **Visualization을 비활성화 시킴.**  
`#/control/execute mac/Visualisation.mac` ➡ **mac 폴더에 있는 visualisation.mac 매크로 파일 실행-> Visualization 설정**

```
#=====
# GEOMETRY
#-----
```

`/gate/geometry/setMaterialDatabase GateMaterials.db` ➡ **GateMaterials.db파일 가져오기**  
**->Materials.xml파일도 따라서 불러들여짐.**

```
# World
/gate/world/geometry/setXLength 1.5 m
/gate/world/geometry/setYLength 1.5 m
/gate/world/geometry/setZLength 1.5 m
/gate/world/setMaterial Air
/gate/world/vis/setVisible 1
/gate/world/vis/forceWireframe
```

} world 설정

↪ **시뮬레이션을 진행할 world의 x,y,z크기를 설정.**  
**World밖으로 나가는 감마선이나 양전자 등의 입자는 particle tracking이 끝나게 됨.**

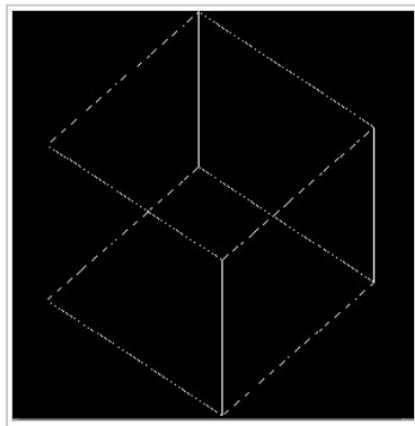
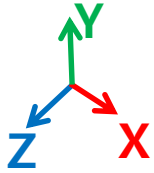


Fig 1.1: World volume.



# 2. Define the Scanner geometry

```
#=====
# GEOMETRY
#=====
```

/gate/geometry/setMaterialDatabase GateMaterials.db ➡ 물질의 정보들이 담긴 GateMaterial.db파일 가져옴.  
 (ex> 물질들의 원자구성, 밀도, 기체, 액체, 고체상태 등)

```
#
# WORLD
#
/gate/world/geometry/setXLength 400. cm
/gate/world/geometry/setYLength 400. cm
/gate/world/geometry/setZLength 400. cm
```

↪ 시뮬레이션을 진행할 world의 x,y,z크기를 설정.  
 World밖으로 나가는 감마선이나 양전자 등의 입자는  
 particle tracking이 끝나게 됨.

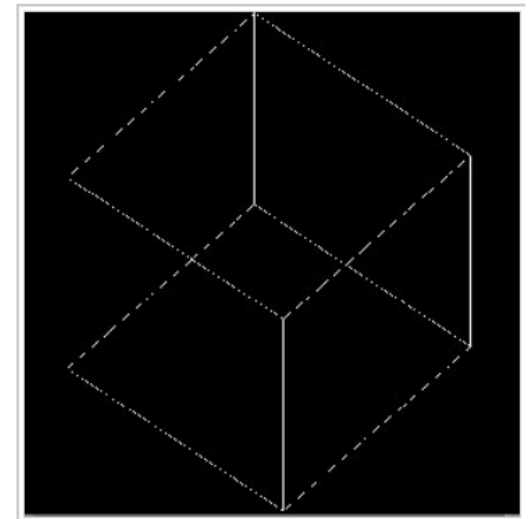
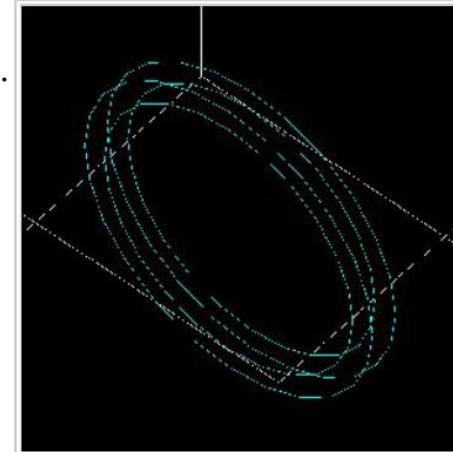


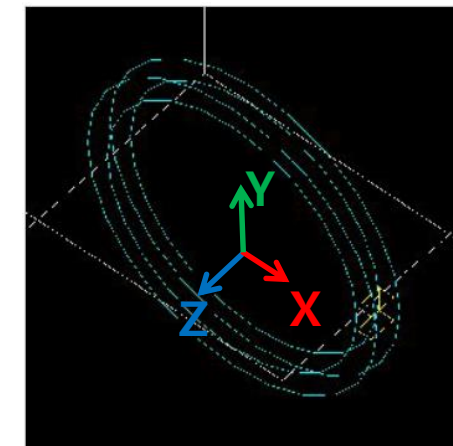
Fig 1.1: World volume.

## 2. Define the Scanner geometry

```
# SYSTEM (ECAT)
/gate/world/daughters/name ecat ➔ World의 하부 볼륨으로 ecat System을 만듦.
/gate/world/daughters/insert cylinder ➔ cylinder모양으로 ecat을 만듦.
/gate/ecat/setMaterial Air ➔ ecat의 물질을 Air로 설정.
/gate/ecat/geometry/setRmax 44.2 cm ➔ ecat의 바깥쪽 반지름을 설정.
/gate/ecat/geometry/setRmin 41.2 cm ➔ ecat의 안쪽 반지름을 설정.
/gate/ecat/geometry/setHeight 15.52 cm ➔ ecat의 z-축 길이를 설정.
/gate/ecat/vis/forceWireframe ➔ ecat의 vis속성을 wireframe으로 설정.
(다른 속성:forceSolid)
```



```
# BLOCK
/gate/ecat/daughters/name vol1 ➔ ecat에 vol1이란 이름을 가지는 하위 volume 생성
/gate/ecat/daughters/insert box ➔ box모양으로 vol1 volume을 만듦.
/gate/vol1/placement/setTranslation 427.0 0.0 0.0 mm ➔ x,y,z방향으로 이동시킴.
/gate/vol1/geometry/setXLength 30.0 mm
/gate/vol1/geometry/setYLength 35.8594 mm
/gate/vol1/geometry/setZLength 38.7 mm
/gate/vol1/setMaterial Air ➔ vol1의 물질을 Air로 설정.
/gate/crystal/vis/setColor yellow ➔ vol1의 색상을 yellow로 설정.
/gate/vol1/vis/forceWireframe
```



## 2. Define the Scanner geometry

```
#          C R Y S T A L
/gate/vol1/daughters/name vol2   ➔ vol1에 vol2이란 이름을 가지는 하위 volume 생성
/gate/vol1/daughters/insert box  ➔ box모양으로 vol2 volume을 만듦.
/gate/vol2/geometry/setXLength 30.0 mm
/gate/vol2/geometry/setYLength 4.4 mm
/gate/vol2/geometry/setZLength 4.75 mm
/gate/vol2/setMaterial BGO      ➔ vol2의 물질을 BGO로 설정.
/gate/vol2/vis/setColor red     ➔ vol2의 색상을 red로 설정.
```

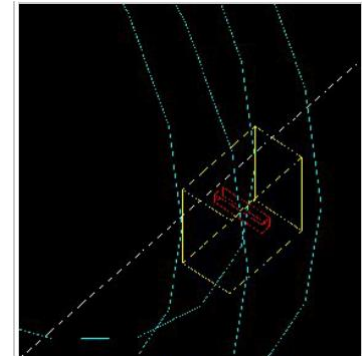


Figure 1.4: crystal, daughter of the block

```
#          R E P E A T   C R Y S T A L
/gate/vol2/repeaters/insert cubicArray ➔ vol2를 cubicArray방식으로 반복시킴.
/gate/vol2/cubicArray/setRepeatNumberX 1   vol2를 y,z방향으로 8개씩 반복 복사 시키
/gate/vol2/cubicArray/setRepeatNumberY 8   고 pitch를 0, 4.4942, 4.85 mm로 설정
/gate/vol2/cubicArray/setRepeatNumberZ 8
/gate/vol2/cubicArray/setRepeatVector 0. 4.4942 4.85 mm
```

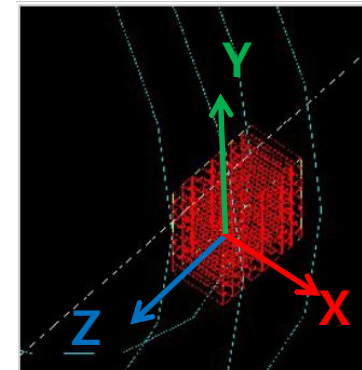
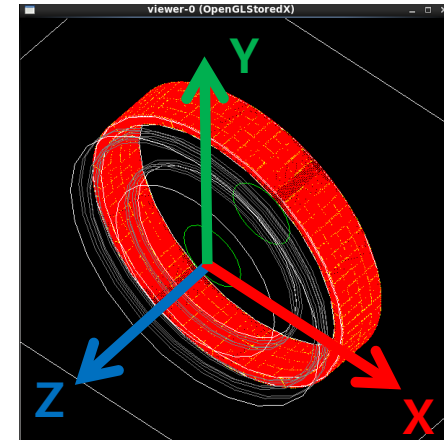


Figure 1.5: matrix of crystals

## 2. Define the Scanner geometry

```
#          R E P E A T   B L O C K
/gate/vol1/repeaters/insert linear ➔ vol1을 linear형식으로 반복시킴.
/gate/vol1/linear/setRepeatNumber 4 ➔ vol1을 4개 반복시켜 복사.
/gate/vol1/linear/setRepeatVector 0. 0. 38.8 mm ➔ 반복 복사의 간격(pitch)
                                                    Z축으로 38.8 mm로 설정.
/gate/vol1/repeaters/insert ring ➔ vol1을 ring형식으로 반복시킴.
/gate/vol1/ring/setRepeatNumber 72 ➔ vol1을 72개로 360도를 채움.
```



- (중요) 섬광결정(vol2)과 블록모듈(vol1)으로부터 감마선 정보(에너지, 반응위치)를 수집하기 위해서는
1. 두 volume을 system에 붙이고(attach)
  2. vol2 volume에 **Crystal Sensitive Detector**를 붙여야 함.

```
# 1. ATTACH SYSTEM
/gate/systems/ecat/block/attach vol1 ➔ vol1을 system/ecat/block에 붙임.
/gate/systems/ecat/crystal/attach vol2 ➔ vol2을 system/ecat/crystal에 붙임.
```

```
# 2. ATTACH CRYSTAL SD
/gate/vol2/attachCrystalSD ➔ vol2에 Crystal Sensitive Detector를 붙임
```



## 2. Define the Scanner geometry

```
# TUNGSTEN SHIELD
```

```
/gate/world/daughters/name carter ➡ carter라는 이름을 가지는 volume 생성
/gate/world/daughters/insert cylinder ➡ cylinder모양으로 volume carter을 만듦.
/gate/carter/setMaterial Air ➡ carter의 물질을 Air로 설정.
/gate/carter/geometry/setRmax 44. cm ➡ carter의 바깥쪽 반지름을 설정.
/gate/carter/geometry/setRmin 28. cm ➡ carter의 안쪽 반지름을 설정.
/gate/carter/geometry/setHeight 9. cm ➡ carter의 Z-축 길이를 설정.
/gate/carter/placement/setTranslation 0.0 0.0 12.5 cm ➡ x,y,z방향으로 이동(Translation)시킴.
/gate/carter/vis/forceWireframe ➡ carter의 vis속성을 wireframe으로 설정.
```

```
/gate/carter/daughters/name carter1 ➡ carter에 carter1이란 이름을 가지는 하위 volume 생성
/gate/carter/daughters/insert cylinder ➡ box모양으로 carter1 volume을 만듦.
/gate/carter1/setMaterial Tungsten ➡ carter1의 물질을 Tungsten으로 설정.
/gate/carter1/geometry/setRmax 30.0 cm ➡ carter1의 바깥쪽 반지름을 설정.
/gate/carter1/geometry/setRmin 29.0 cm ➡ carter1의 안쪽 반지름을 설정.
/gate/carter1/geometry/setHeight 3.0 cm ➡ carter1의 Z-축 길이를 설정.
/gate/carter1/vis/setColor grey ➡ carter1의 vis색상을 grey로 설정.
/gate/carter1/placement/setTranslation 0.0 0.0 -3 cm ➡ x,y,z방향으로 이동(Translation)시킴.
```

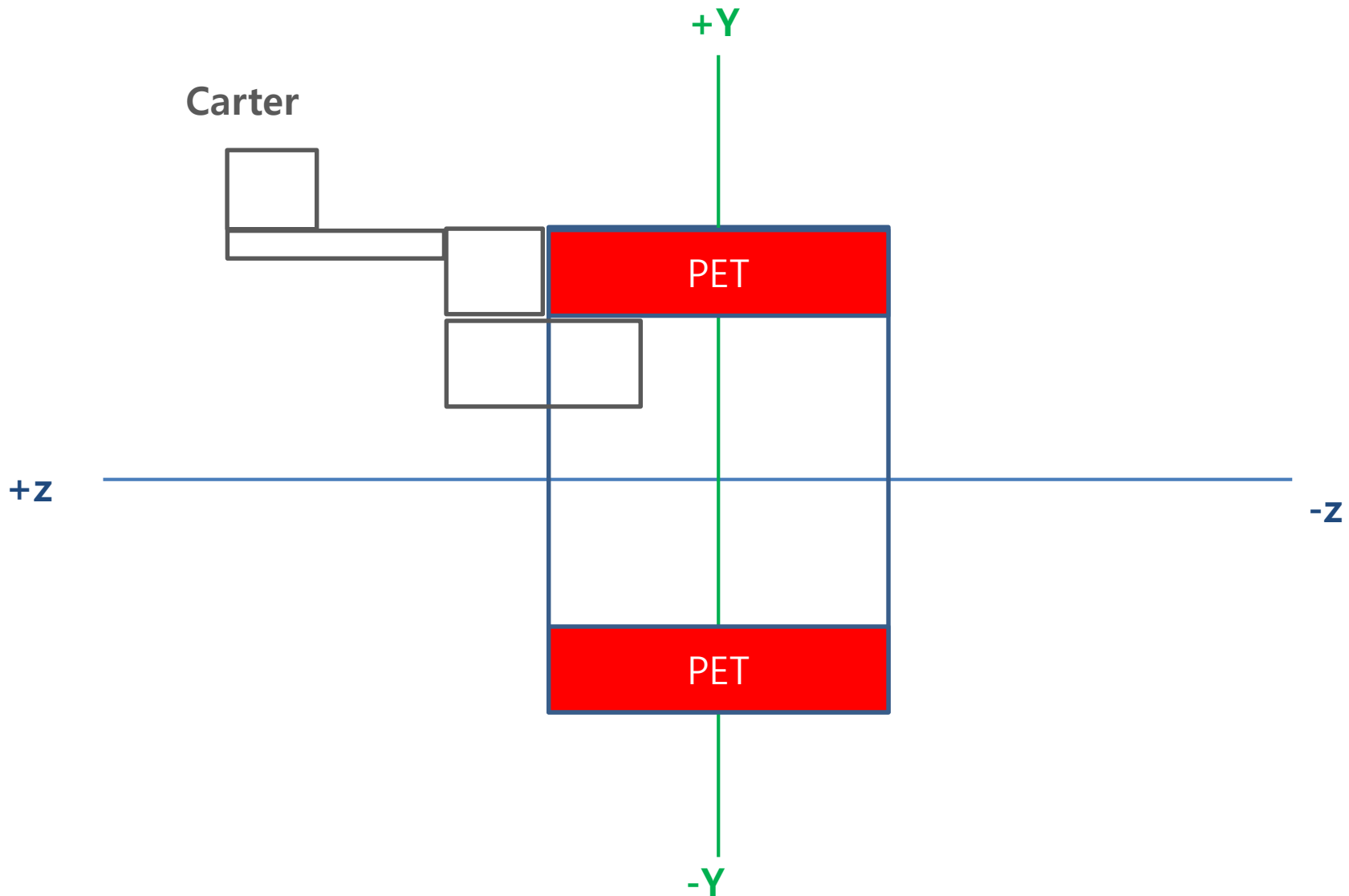
```
/gate/carter/daughters/name carter2 ➡ carter에 carter2이란 이름을 가지는 하위 volume 생성
/gate/carter/daughters/insert cylinder ➡ box모양으로 carter2 volume을 만듦.
/gate/carter2/setMaterial Tungsten ➡ carter2의 물질을 Tungsten으로 설정.
/gate/carter2/geometry/setRmax 40.7 cm ➡ carter2의 바깥쪽 반지름을 설정.
/gate/carter2/geometry/setRmin 30.0 cm ➡ carter2의 안쪽 반지름을 설정.
/gate/carter2/geometry/setHeight 1.0 cm ➡ carter2의 Z-축 길이를 설정.
/gate/carter2/vis/setColor grey ➡ carter2의 vis색상을 grey로 설정.
/gate/carter2/placement/setTranslation 0.0 0.0 -2 cm ➡ x,y,z방향으로 이동(Translation)시킴.
```

## 2. Define the Scanner geometry

/gate/carter/daughters/name carter3 ➡ **carter**에 **carter3**이란 이름을 가지는 하위 volume 생성  
 /gate/carter/daughters/insert cylinder ➡ box모양으로 **carter3** volume을 만듦.  
 /gate/carter3/setMaterial **Tungsten** ➡ **carter3**의 물질을 **Tungsten**으로 설정.  
 /gate/carter3/geometry/setRmax 40.7 cm ➡ **carter3**의 바깥쪽 반지름을 설정.  
 /gate/carter3/geometry/setRmin 39.7 cm ➡ **carter3**의 안쪽 반지름을 설정.  
 /gate/carter3/geometry/setHeight 3.0 cm ➡ **carter3**의 **Z-축** 길이를 설정.  
 /gate/carter3/vis/setColor grey ➡ **carter3**의 vis색상을 **grey**로 설정.  
 /gate/carter3/placement/setTranslation 0.0 0.0 0 cm ➡ **x,y,z**방향으로 이동(Translation)시킴.

/gate/carter/daughters/name carter4 ➡ **carter**에 **carter4**이란 이름을 가지는 하위 volume 생성  
 /gate/carter/daughters/insert cylinder ➡ box모양으로 **carter4** volume을 만듦.  
 /gate/carter4/setMaterial **Tungsten** ➡ **carter4**의 물질을 **Tungsten**으로 설정.  
 /gate/carter4/geometry/setRmax 43.7 cm ➡ **carter4**의 바깥쪽 반지름을 설정.  
 /gate/carter4/geometry/setRmin 40.7 cm ➡ **carter4**의 안쪽 반지름을 설정.  
 /gate/carter4/geometry/setHeight 1.0 cm ➡ **carter4**의 **Z-축** 길이를 설정.  
 /gate/carter4/vis/setColor grey ➡ **carter4**의 vis색상을 **grey**로 설정.  
 /gate/carter4/placement/setTranslation 0.0 0.0 1 cm ➡ **x,y,z**방향으로 이동(Translation)시킴.

# Carter (Tungsten shielding)



## 2. Voxelized phantom 설정

```

#=====
#  Voxellized phantom (Ncat)
#=====
/gate/world/daughters/name Ncat ➡ world의 하위볼륨으로 Ncat생성 (Ncat은 keyword임).
/gate/world/daughters/insert regularMatrix ➡ Ncat에 Voxelized volume을 삽입
#/gate/world/daughters/insert compressedMatrix

/gate/Ncat/geometry/insertReader interfile ➡ Interfile Reader를 삽입
/gate/Ncat/verbose 0
/gate/RTVPhantom/insert RTVPhantom ➡ RTVPhantom라는 이름으로 RTPPhantom생성(RTPPhantom은 keyword)
/gate/RTVPhantom/AttachTo Ncat ➡ RTVPhantom을 Ncat에 붙임(attach)
/gate/RTVPhantom/SetNumberOfFrames 10 ➡ RTVPhantom(=Ncat)의 총 frame개수를 설정
/gate/RTVPhantom/SetTimePerFrame 0.1 s ➡ RTVPhantom(=Ncat)의 frame당 시간을 설정
/gate/RTVPhantom/verbose 0
/gate/RTVPhantom/setHeaderFileName data/NCAT_header.h33 ➡ Interfile헤더 파일을 읽어 들여 Ncat에 붙임
/gate/Ncat/interfileReader/insertTranslator range ➡ Ncat에 rangeTranslator 삽입(CT 영상 밝기 범위->Material로 변환)
/gate/Ncat/interfileReader/rangeTranslator/readTable data/range.dat ➡ data폴더의 range.dat파일을 읽어 들임.
/gate/Ncat/interfileReader/rangeTranslator/describe 1
# Ncat was moved -40 mm along the z-dir to locate at the center of the PET FOV (2015.05.28.KangHG)
/gate/Ncat/placement/setTranslation 0. 0. -40. mm ➡ Ncat을 x,y,z방향으로 translation 시킴.
/gate/RTVPhantom/setBaseFileName data/NCAT ➡ NCAT interfile을 data폴더의 NCAT으로 시작하는 파일명으로 지정
#/gate/Ncat/vis/forceSolid
#/gate/Ncat/vis/setColor Blue

/gate/Ncat/attachVoxelPhantomSD ➡ Ncat볼륨에 PhantomSensitiveDetector를 attach시킴.
#/gate/geometry/update

```

### 3. Physics 설정

```
#=====
#           Physics
#=====
```

```
/gate/physics/addProcess PhotoElectric
/gate/physics/processes/PhotoElectric/setModel StandardModel
```

```
/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel StandardModel
```

```
/gate/physics/addProcess GammaConversion
/gate/physics/processes/GammaConversion/setModel StandardModel
```

```
/gate/physics/addProcess ElectronIonisation
/gate/physics/processes/ElectronIonisation/setModel StandardModel e-
/gate/physics/processes/ElectronIonisation/setModel StandardModel e+
/gate/physics/processes/ElectronIonisation/setStepFunction e+ 0.2 0.1 mm
/gate/physics/processes/ElectronIonisation/setStepFunction e- 0.2 0.1 mm
```

```
/gate/physics/addProcess Bremsstrahlung
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e-
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e+
```

```
/gate/physics/addProcess PositronAnnihilation ➡ 전자쌍 소멸
```

### 3. Physics 설정

# Below 3 lines are WRONG! KangHG(2015.04.14)

#-----

# Before -> MultipleScattering (X) WRONG

# After -> eMultipleScattering (O) OK

#-----

#/gate/physics/addProcess MultipleScattering

#/gate/physics/processes/MultipleScattering/setGeometricalStepLimiterType e- distanceToBoundary

#/gate/physics/processes/MultipleScattering/setGeometricalStepLimiterType e+ distanceToBoundary

# I added "e" at the prefix of the MultipleScattering! KangHG(2015.04.14)

/gate/physics/addProcess eMultipleScattering

/gate/physics/processes/eMultipleScattering/setGeometricalStepLimiterType e- distanceToBoundary

/gate/physics/processes/eMultipleScattering/setGeometricalStepLimiterType e+ distanceToBoundary

/gate/physics/processList Enabled

/gate/physics/processList Initialized

### 3. Physics Cut 설정

```
#=====
# CUTS
#=====
```

```
/gate/physics/Gamma/SetCutInRegion    crystal 1.0 cm
/gate/physics/Electron/SetCutInRegion  crystal 1.0 cm
/gate/physics/Positron/SetCutInRegion  crystal 1.0 cm
```

```
#/gate/physics/Gamma/SetCutInRegion    phantom 0.1 mm
#/gate/physics/Electron/SetCutInRegion  phantom 0.1 mm
#/gate/physics/Positron/SetCutInRegion  phantom 0.1 mm
```

```
#/gate/physics/SetMaxStepSizeInRegion  phantom 0.01 mm
```

```
#/gate/physics/Gamma/SetCutInRegion    RTVPhantom 0.1 mm
#/gate/physics/Electron/SetCutInRegion  RTVPhantom 0.1 mm
#/gate/physics/Positron/SetCutInRegion  RTVPhantom 0.1 mm
```

```
#/gate/physics/SetMaxStepSizeInRegion  RTVPhantom 0.01 mm
```

## 4 – Initialization the simulation :

### (Compute cross-section tables for particles)

```
#=====
# INITIALISATION
#=====
```

```
/gate/run/initialize
# Enable the following lines to display available and enabled processes
# /gate/physics/processList Available
# /gate/physics/processList Enabled
```

**Initialization 이후엔 매크로에서 아래 설정 값을 바꾸지 못함.**

1. **Scanner의 geometry**
2. **Phantom의 geometry**
3. **Physics process**



## 5. Source (voxelized-source) : 511 keV

```
#=====
```

```
# Cubic voxelized-source (PET image)
```

```
#=====
```

```
/gate/source/addSource voxel voxel ➡ voxel 이라는 이름의 source를 voxel로 생성
```

```
/gate/source/verbose 0
```

```
/gate/source/voxel/reader/insert interfile
```

```
/gate/RTVPhantom/AttachToSource voxel ➡ RTVPhantom을 voxel source에 붙임
```

```
/gate/source/voxel/interfileReader/translator/insert range ➡ voxel 에 rangeTranslator를 삽입
```

```
/gate/source/voxel/interfileReader/rangeTranslator/readTable data/activityRange.dat ➡ rangeTranslator 사용
```

```
/gate/source/voxel/interfileReader/SetTimeActivityTablesFrom data/acti.range ➡ TAC table을 읽어 들임.
```

```
/gate/source/voxel/interfileReader/SetTimeSampling 0.1 s ➡ Voxelized-source의 frame당 시간을 설정
```

```
/gate/source/voxel/interfileReader/rangeTranslator/describe 1
```

```
# 4D phantom was moved -40 mm along the z-dir.
```

```
/gate/source/voxel/setPosition -200. -200. -240. mm ➡ Voxel source를 x,y,z방향으로 translation 시킴.
```

```
/gate/source/voxel/setType backtoback ➡ backtoback source를 voxel로 생성
```

```
/gate/source/voxel/gps/particle gamma ➡ Particle type을 gamma로 설정
```

```
/gate/source/voxel/setForcedUnstableFlag true } 반감기를 6686.2 s로 설정(F-18)
```

```
/gate/source/voxel/setForcedHalfLife 6586.2 s }
```

```
/gate/source/voxel/gps/energytype Mono } gamma는 511 keV의 mono energy로 설정
```

```
/gate/source/voxel/gps/monoenergy 0.511 MeV }
```

```
/gate/source/voxel/gps/confine NULL ➡ gamma source를 다른 볼륨에 confine하지 않음.
```

```
/gate/source/voxel/gps/angtype iso ➡ gamma source를 isotropic하게 방출시킴.
```

```
/gate/source/voxel/dump 0
```

```
/gate/source/voxel/verbose 0
```

# 5. Source (rangeTranslator)

## Voxelized-source

/gate/source/voxel/interfileReader/rangeTranslator/readTable

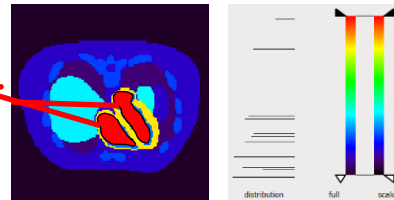
activityRange DAT 파일

```
1
246 246 0.
```

/gate/source/voxel/interfileReader/SetTimeActivityTablesFrom

acti RANGE 파일

```
1
246 246 data/liver.txt
```

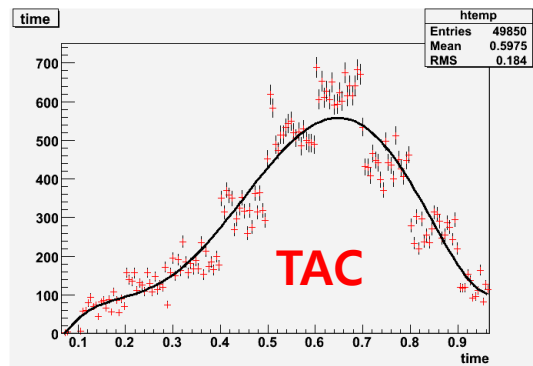


심근 혈액 풀 영상  
(activity)

liver 텍스트 문서

Time bin개수 → 11

```
0.0 0.01
0.1 1
0.2 2
0.3 3
0.4 5
0.5 8
0.6 10
0.7 7
0.8 4
0.9 2
1 1
```



Time [sec] Activity [Bq]

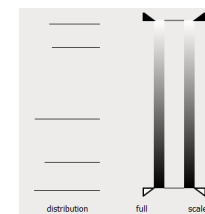
## Voxelized-phantom

range DAT 파일

```
4
0 0 Air false 0,0 0,0 0,0 0,0 1,0
6 6 Lung false 0,8 0,8 0,8 1,0 1,0
15 15 Breast false 0,5 0,5 0,5 1,0 1,0
29 36 RibBone true 1,0 1,0 1,0 1,0 1,0
```



감쇄 영상  
(attenuation)



NCAT\_header H33 파일

```
!matrix size [1] := 128
!matrix size [2] := 128
!name of data file := data/NCAT_atn_1.bin
!number format := unsigned integer
imagedata byte order := LITTLEENDIAN
scaling factor (mm/pixel) [1] := +3,125000e+00
scaling factor (mm/pixel) [2] := +3,125000e+00
!number of slices := 128
slice thickness (pixels) := +3,125000e+00
```

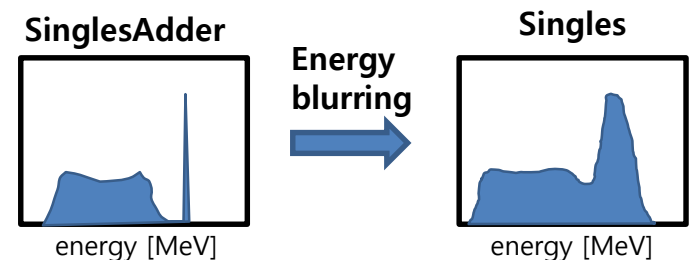
## 6. Digitizer (energy thresholder)

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

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

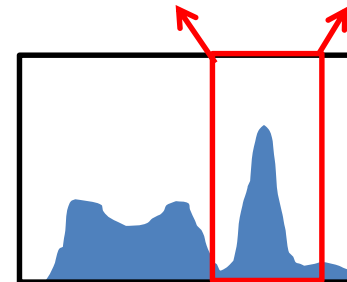
```
# ENERGY BLURRING
/gate/digitizer/Singles/insert blurring
/gate/digitizer/Singles/blurring/setResolution 0.26 }
/gate/digitizer/Singles/blurring/setEnergyOfReference 511. keV
```

Pulse들이 Singles로 갈 때 511 keV에 대하여 Detector의 에너지 분해능(Energy resolution)을 26%로 반영시킴.



```
# ENERGY CUT
/gate/digitizer/Singles/insert thresholder
/gate/digitizer/Singles/thresholder/setThreshold 350. keV
/gate/digitizer/Singles/insert upholder
/gate/digitizer/Singles/upholder/setUphold 750. keV
```

Singles의 에너지 창을 설정  
 ▪최소 350 keV, 최대 750 keV



Thresholder= 350 keV    upholder= 750 keV

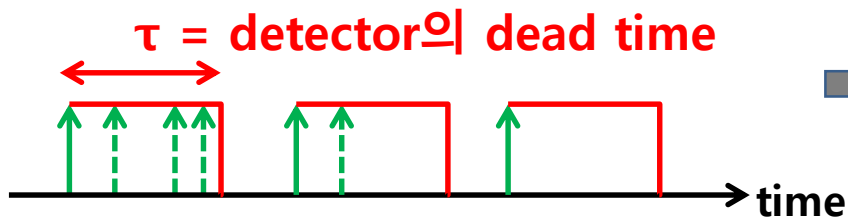
## 6. Digitizer (Dead time)

# DEAD TIME

/gate/digitizer/Singles/insert **deadtime** → Singles 검출 시 **deadtime** 사용  
 /gate/digitizer/Singles/deadtime/setDeadTime **3000. ns** → Deadtime을 3000 ns으로 설정  
 /gate/digitizer/Singles/deadtime/setMode **paralysable** → Deadtime 마비성으로 설정  
 /gate/digitizer/Singles/deadtime/chooseDTVVolume **vol1** → **vol1(block)**에 대해 deadtime 설정

$\tau$  = Detector의 **dead time**.  
 Detector(검출기 블록)은 주로  
 PMT, SiPM임.

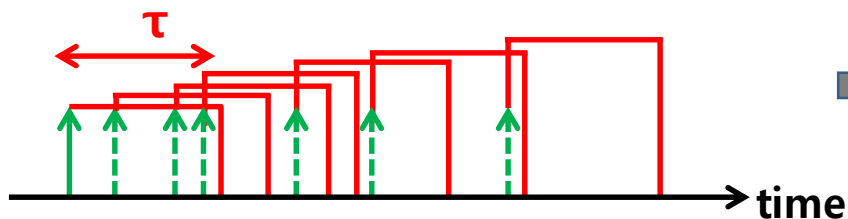
비 마비성 (Non-paralysable dead time)



들어온 감마선 : 7개  
 검출된 감마선 : 3개

마비성 (Paralysable dead time)

:dead time이 뒤이은 감마선에 의해 계속 늘어남)



들어온 감마선 : 7개  
 검출된 감마선 : 1개

## 6. Digitizer (Coincidence sorter)

```
#          COINCI  SORTER
```

```
/gate/digitizer/Coincidences/setWindow 10. ns ➡ 동시계수 시간 창( $\tau$ ), 주로 5~10 ns
/gate/digitizer/Coincidences/setOffset  0. ns ➡ 동시계수 시간창의 time shift (prompt의 경우 0)
/gate/digitizer/Coincidences/describe
```

```
/gate/digitizer/name delay ➡ 지연계수 시간 창을 삽입.
```

```
/gate/digitizer/insert coincidenceSorter
```

```
/gate/digitizer/delay/setWindow 10. ns ➡ 지연계수 시간 창( $\tau$ ), 동시계수 시간 창과 같게 해줌.
```

```
/gate/digitizer/delay/setOffset 500. ns ➡ 지연계수 시간창의 time shift (delay의 경우 동시계수 시간창 보다 길게 해줌.)
```

```
/gate/digitizer/delay/describe ➡ delay의 결과를 출력화면에 반환시킴.
```

```
/gate/digitizer/name finalCoinc ➡ finalCoinc 라는 coincidence sorter를 생성
```

```
/gate/digitizer/insert coincidenceChain ➡ coincidenceChain을 finalCoinc에 사용.
```

```
/gate/digitizer/finalCoinc/addInputName delay
/gate/digitizer/finalCoinc/addInputName Coincidences } finalCoinc에서 delay와 Coincidences를
                                                         서로 그룹화시킴.
```

```
/gate/digitizer/finalCoinc/usePriority true ➡
```

```
/gate/digitizer/finalCoinc/describe ➡ finalCoinc의 결과를 출력화면에 반환시킴.
```

# Verbose

```
#=====
```

```
# Verbose
```

```
#=====
```

```
#/gate/verbose Physic 0
```

```
#/gate/verbose Cuts 0
```

```
#/gate/verbose Actor 0
```

```
#/gate/verbose SD 0
```

```
#/gate/verbose Actions 0
```

```
#/gate/verbose Step 0
```

```
#/gate/verbose Error 0
```

```
#/gate/verbose Warning 0
```

```
#/gate/verbose Output 0
```

```
#/gate/verbose Core 0
```

```
/run/verbose 0
```

```
/event/verbose 0
```

```
/tracking/verbose 0
```

## 7. Output setup (ROOT 출력 파일의 포맷 설정)

```
# SETUP - ROOT FILE
```

```
/gate/output/root/enable ➡ ROOT 파일 출력 enable
```

ROOT 출력 파일명.  
확장자 **.root**는 자동으로  
붙어서 나옴.(binary)

```
/gate/output/root/setFileName Root_output/2015_05_28_TAC_NCAT_Slice60sec_Acq1sec_test00
```

```
/gate/output/root/setRootSinglesAdderFlag 0 ➡ Adder에 의한 결과파일 생성x
```

```
/gate/output/root/setRootSinglesReadoutFlag 0 ➡ readout에 의한 결과파일 생성x
```

```
/gate/output/root/setRootHitFlag 0 ➡ Hit에 의한 결과파일 생성x
```

```
/gate/output/root/setRootSinglesFlag 1 ➡ Singles 결과파일 생성
```

```
/gate/output/root/setRootCoincidencesFlag 1 ➡ Coincidences결과파일 생성 (필수)
```

출력 변수 : 검출된 감마선의 에너지, 동위원소의 위치, 검출기의 x,y,z  
좌표상에서의 감마선 반응 위치, **Sinogram** 등

파일명.root ➡ Ana.cc파일로  
sinogram만 추출 ➡ Sinogram.txt ➡ 매트랩 FBP 코드로  
영상 재구성

readout에 의한 결과파일 생성x

# 7. Output setup (Raw 2D sinogram 출력 파일)

System으로 **ecat system** 또는 **ecatAccel system**이  
선택되어 있는 경우에 사용가능.

2D sinogram 출력 파일명.

- 파일명.ima (binary, uint16의 2D sinogram)
- 파일명.dim (ASCII, sinogram의 size 정보)
- 파일명.info (ASCII, \*.ima를 읽기 위한 최소한의 정보, ex>Ring difference, radial position)

/gate/output/sinogram/enable → Sinogram output 을 enable시킴.

/gate/output/sinogram/setFileName ./RawSinogram\_output/2015\_05\_28\_RawSinogram\_NCAT\_Acq1sec\_test00

/gate/output/sinogram/setTangCrystalBlurring 1.8 mm → Tangential 방향으로 crystal blurring

/gate/output/sinogram/setAxialCrystalBlurring 1.8 mm → Axial 방향으로 crystal blurring

/gate/output/sinogram/verbose 2

#/gate/output/sinogram/RawOutputEnable false → Sinogram output 을 disable시킴.

/gate/output/sinogram/RawOutputEnable true → Sinogram output 을 enable시킴.

/gate/output/sinogram/StoreDelayed → Delay창의 sinogram 출력.

/gate/output/sinogram/StoreScatters → Scatter된 sinogram 출력.

/gate/output/sinogram/setInputDataName finalCoinc → finalCoinc라는 coincidence sorter로 sinogram 출력

/gate/output/sinogram/describe → Sinogram output의 세부사항을 화면에 출력해줌.

example\_PET의 경우 #radial bin=288, #azimuthal bin = 288, #Sinogram = 1024임  
(매트랩 FBP 코드로 영상재구성 가능) PET\_Ecat\_Sinogram\_ima\_binary\_read\_2015\_02\_02\_KangHG.m



## 7. Output setup (ECAT7 3Dsinogram 출력 파일)

```
#      S E T U P - ECAT7 output sinogram
```

```
/gate/output/ecat7/enable
```

```
/gate/output/ecat7/verbose 2
```

```
/gate/output/ecat7/setFileName
```

```
/gate/output/ecat7/setFileName ./ECAT7_output/2015_05_28_ECAT7_NCAT_Slice1sec_Acq1sec_test00
```

```
/gate/output/ecat7/describe
```

```
/gate/output/ecat7/mashing 2
```

```
/gate/output/ecat7/span 9
```

```
/gate/output/ecat7/maxringdiff 22
```

```
/gate/output/ecat7/system 962
```

```
/gate/output/ecat7/IsotopeCode F-18
```

```
/gate/output/ecat7/IsotopeHalfLife 6586.2 second
```

```
/gate/output/ecat7/IsotopeBranchingFraction 1.0
```



ECAT7 출력 파일명.  
확장자 .S는 자동으로  
붙어서 나옴.(binary)

ECAT7 의 형태로 **3D sinogram**이 나옴. (STIR 소프트웨어로 영상재구성 가능)  
STIR는 open소스이고 웹사이트에서 간단한 등록만 하면 다운로드 가능.

# Random number generation

```
# R A N D O M
```

```
# Random engines : JamesRandom, Ranlux64, MersenneTwister(Default)
```

```
#/gate/random/setEngineName Ranlux64
```

```
#/gate/random/setEngineName JamesRandom
```

```
/gate/random/setEngineName MersenneTwister
```

} Random engines의 종류 3가지

- Ranlux64
- JamesRandom
- MersenneTwister** (Default)

```
#/gate/random/setEngineSeed default ➡ CLHEP internal seed를 가져다가 씬(항상 같은 값임)
```

```
#/gate/random/setEngineSeed auto ➡ GATE run마다 random number seed가 새로 생성
```

```
/gate/random/setEngineSeed 1021 ➡ 사용자가 manual로 [0~900,000,000]사이의 정수값을 입력
```

```
#/gate/random/resetEngineFrom fileName ➡ CLHEP internal seed를 가져다가 씬(항상 같은 값임)
```

```
#/gate/random/verbose 1
```

## 8. Start the acquisition (시뮬레이션 시작과 스캔 시간 설정)

```
# =====  
#           START ACQUISITION  
#           TIME PARAMETERS  
# =====
```

Number of Run =  
 $(\text{setTimeStop} - \text{setTimeStart}) / \text{setTimeSlice}$

```
# EXPERIMENT
```

```
#
```

```
/gate/application/setTimeSlice 60. s → 한 슬라이스의 시간  
/gate/application/setTimeStart 0. s → 시뮬레이션 시작 시간  
/gate/application/setTimeStop 60. s → 시뮬레이션 종료 시간
```

```
#
```

```
# LET'S RUN THE SIMULATION!
```

```
#
```

```
/gate/application/startDAQ → GATE 시뮬레이션 시작!
```

**GATEv6.2 예제 파일**

**“Main\_RTVPantom\_NCAT4D\_PET\_ECAT7\_KangHG.mac”  
시뮬레이션 끝!**

# STIR를 이용한 ECAT7 영상 재구성 순서

GATE ECAT7 PET 시뮬레이션



✓ ECAT7\_sinogram.S 출력파일 생성



STIR

명령어> ./ifheaders\_for\_ecat7 ECAT7\_sinogram.S



✓ ECAT7\_sinogram\_S\_f1g1d0b0.hs (ecat header) 생성



vi에디터로 OSMAPOSL\_with\_projector\_pair.par 파일에서 아래와 같이 지정  
input file := ECAT7\_sinogram\_S\_f1g1d0b0.hs



STIR

명령어> ./OSMAPOSL OSMAPOSL\_with\_projector\_pair.par



✓ Output.ahv (header : ASCII)

✓ Output.hv (header : ASCII)

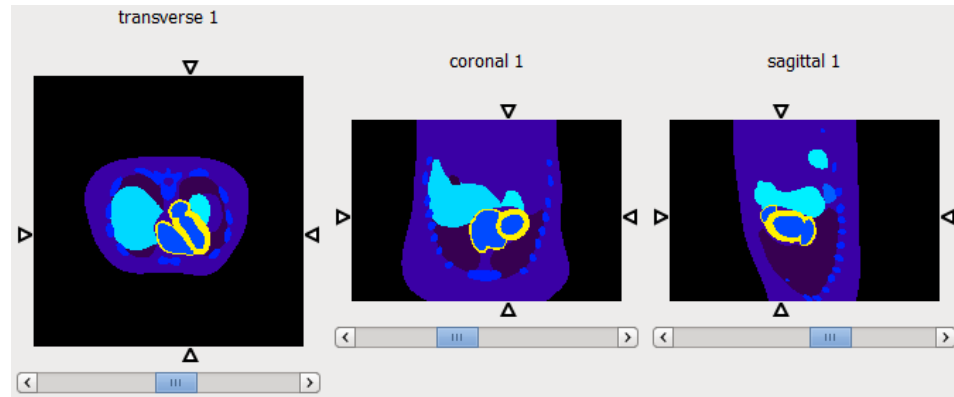
✓ Output.v (Recon 영상 : binary)



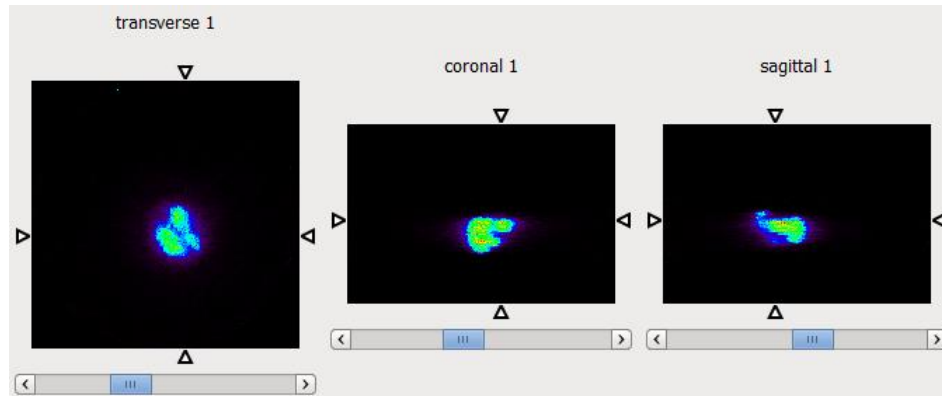
AMIDE로 "import raw data"를  
이용해서 Recon된 영상 확인!

# voxelized-source영상 vs. Recon된 PET영상

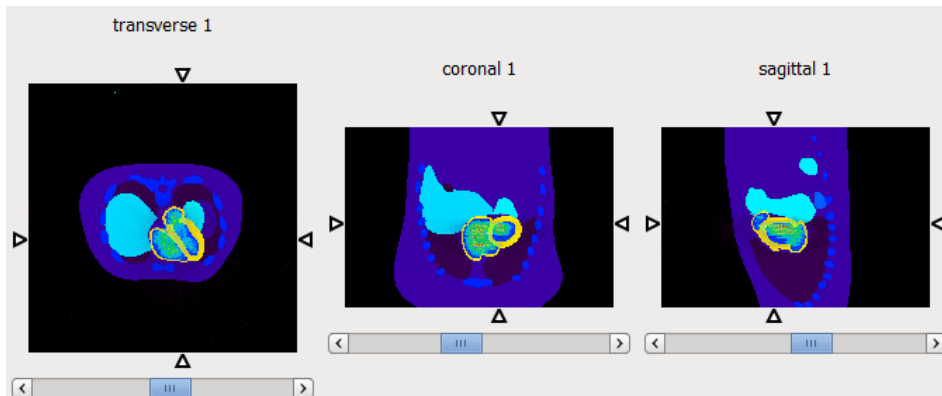
Voxelized-source 영상



STIR로 Recon 된 PET영상  
(Ventricle blood pool 영상)



voxelized source와  
Recon 된 PET 영상의  
융합



# 결론

- NCAT 4D Phantom을 이용하여 voxelized-source(activity)와 voxelized-phantom(attenuation)을 시간에 따른 모션 및 activity 변화를 시뮬레이션 가능
- STIR를 이용해서 얻은 심실의 blood pool영상과 voxelized-source(activity)간에 영상정합이 잘 이루어짐.