Project RoboSPECT

3D SPECT

Hierarchical Object Definitions

CrystalCamBundle

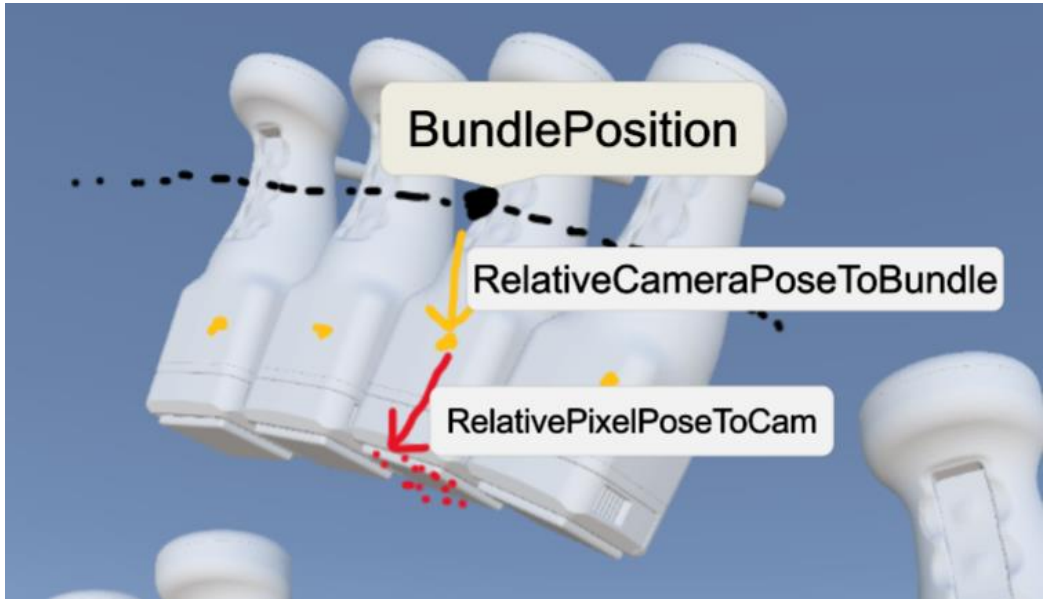
- CrystalCam[]	m_crystalCams
- int	m_numOfCrystalCams
- Vector3	m_position
- Vector3	m_orientation

CrystalCam

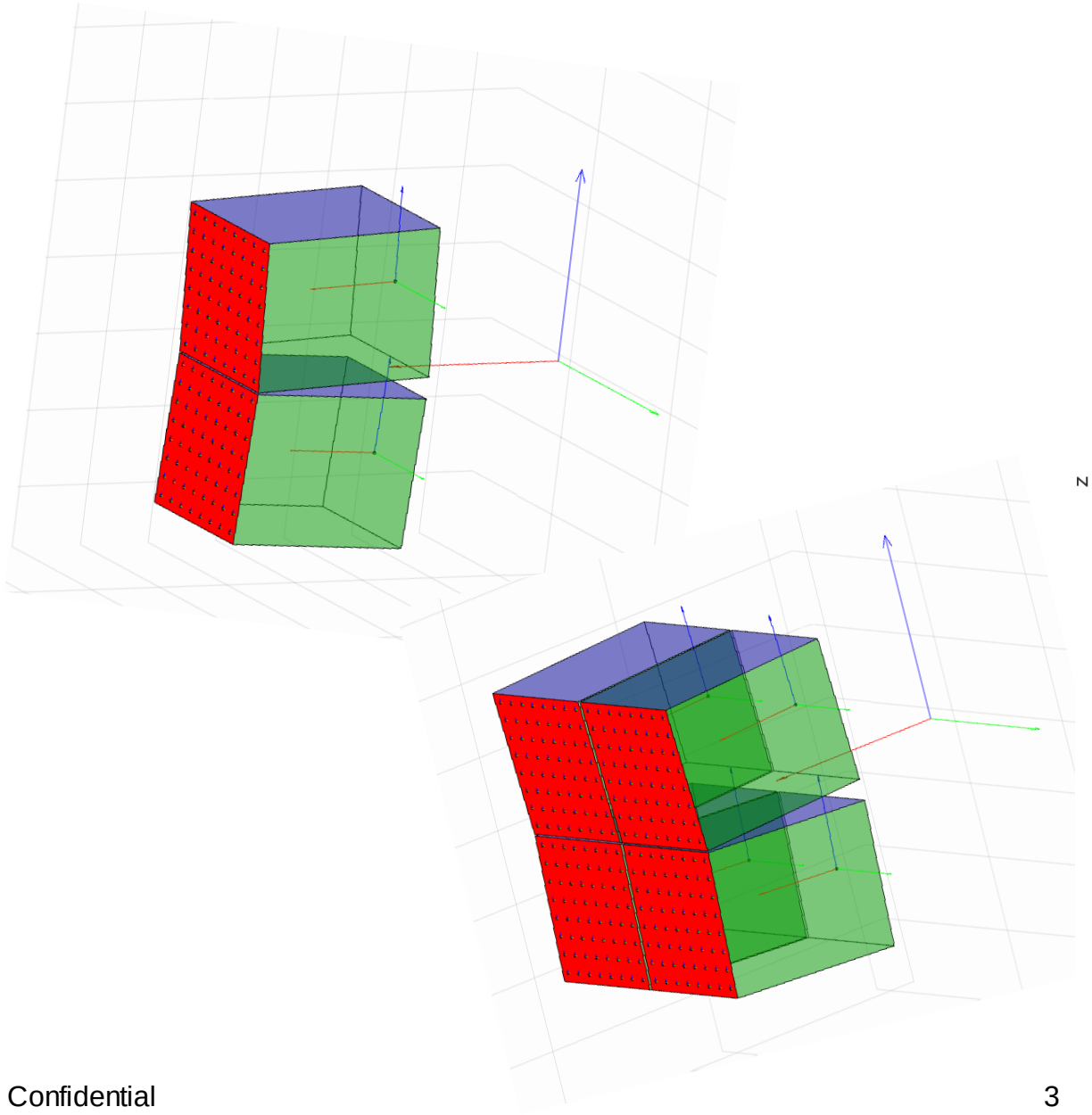
- Pixel[]	m_pixels
- int	m_numOfPixels
- Vector3	m_relativePositionToParentBundle
- Vector3	m_relativeOrientationToParentBundle

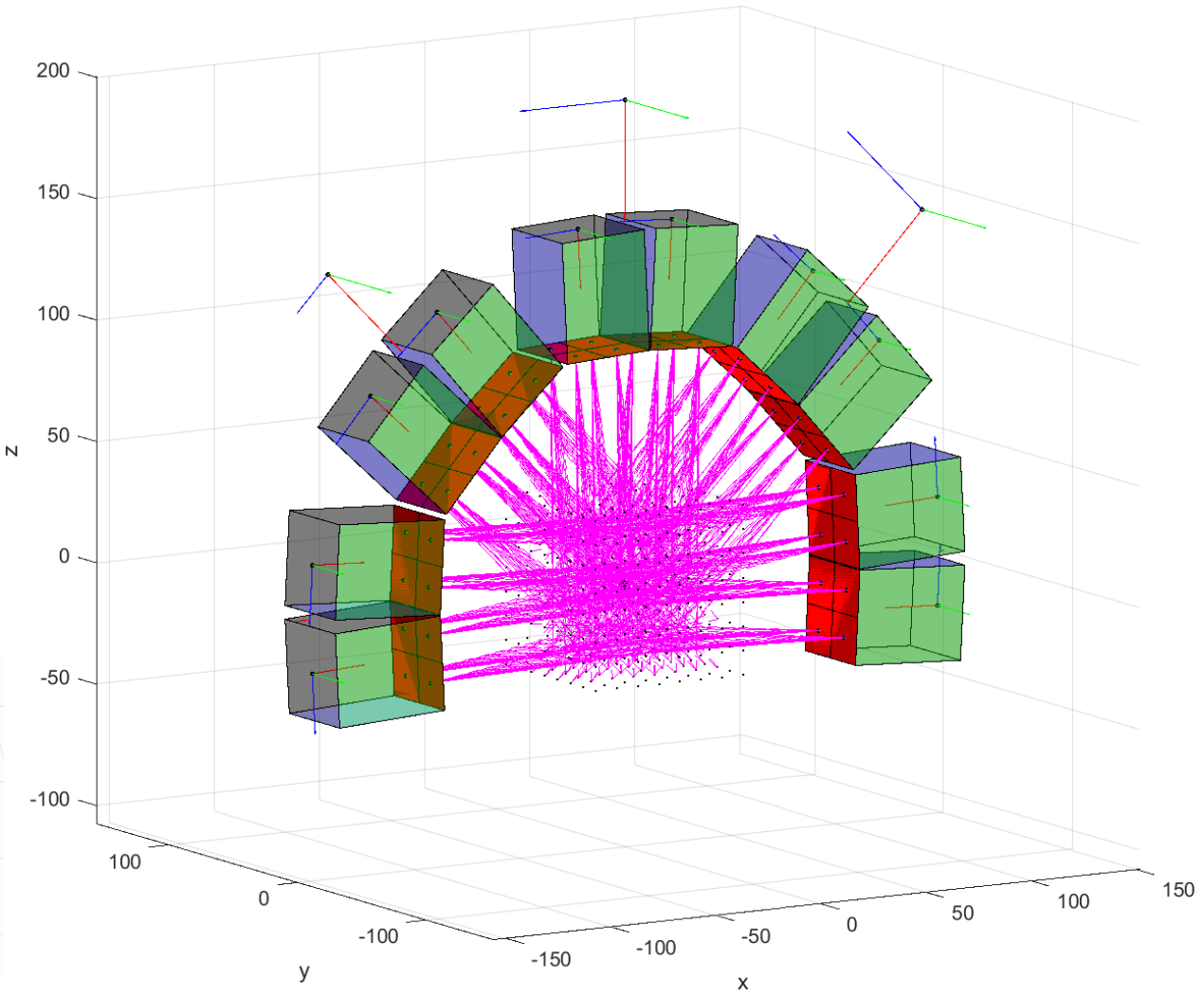
Pixel

- CrystalCam	m_parentCrystalCam
- Vector3	m_relativePositionToParentCamera
- Vector3	m_relativeOrientationToParentCamera



Hierarchical Object Definitions

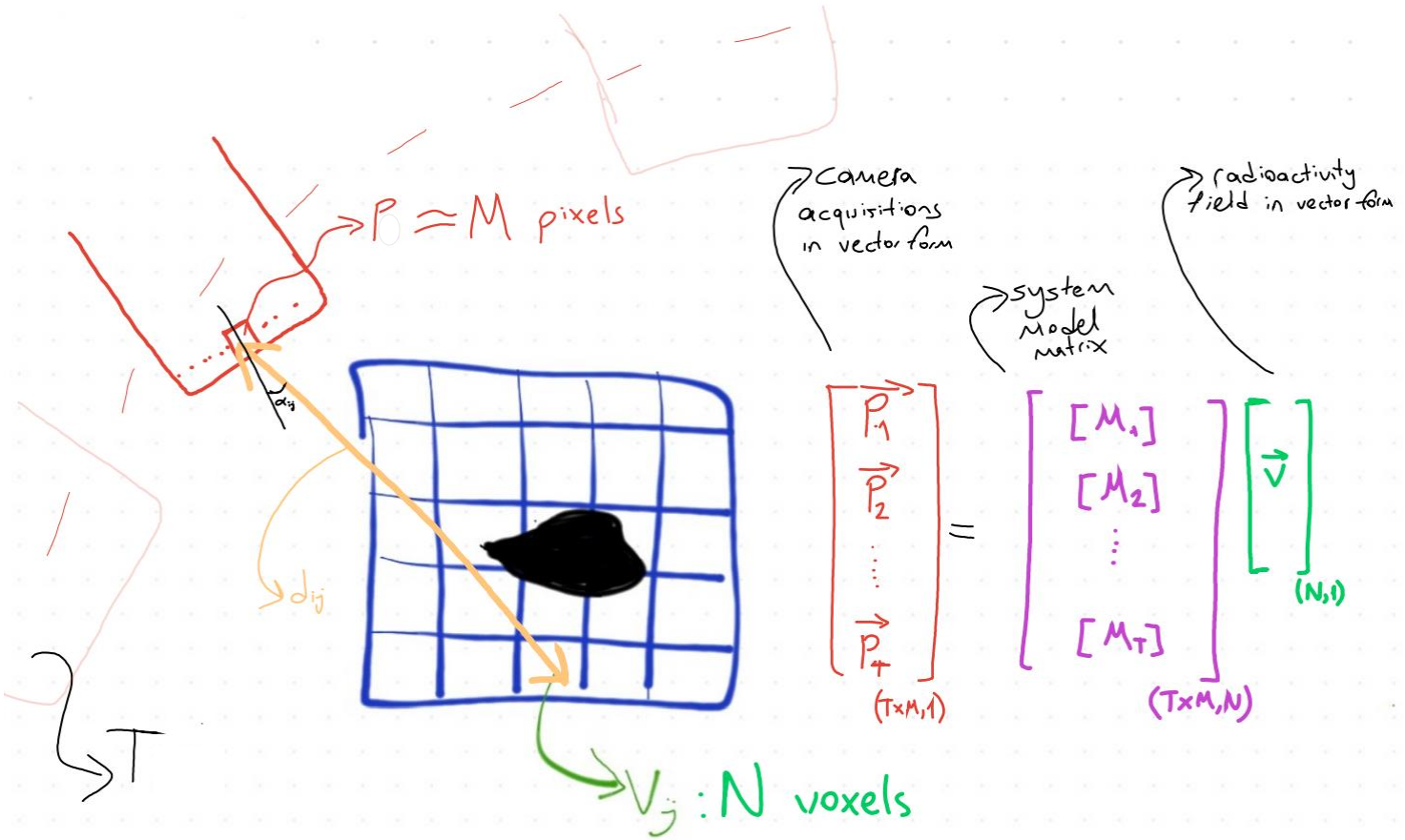




Some References for 3D SPECT

- link PhD dissertation - Robotic freehand SPECT Imaging, Jose Francisco Gardiazabal Schilling
- link PhD dissertation - Tomographic Reconstruction Methods for Optical and Intra–operative Functional Imaging, Tobias Roman Lasser
- link Lecture - Tomographic Image Reconstruction: An Introduction, Erika Garutti & Florian Grüner

Some Theory



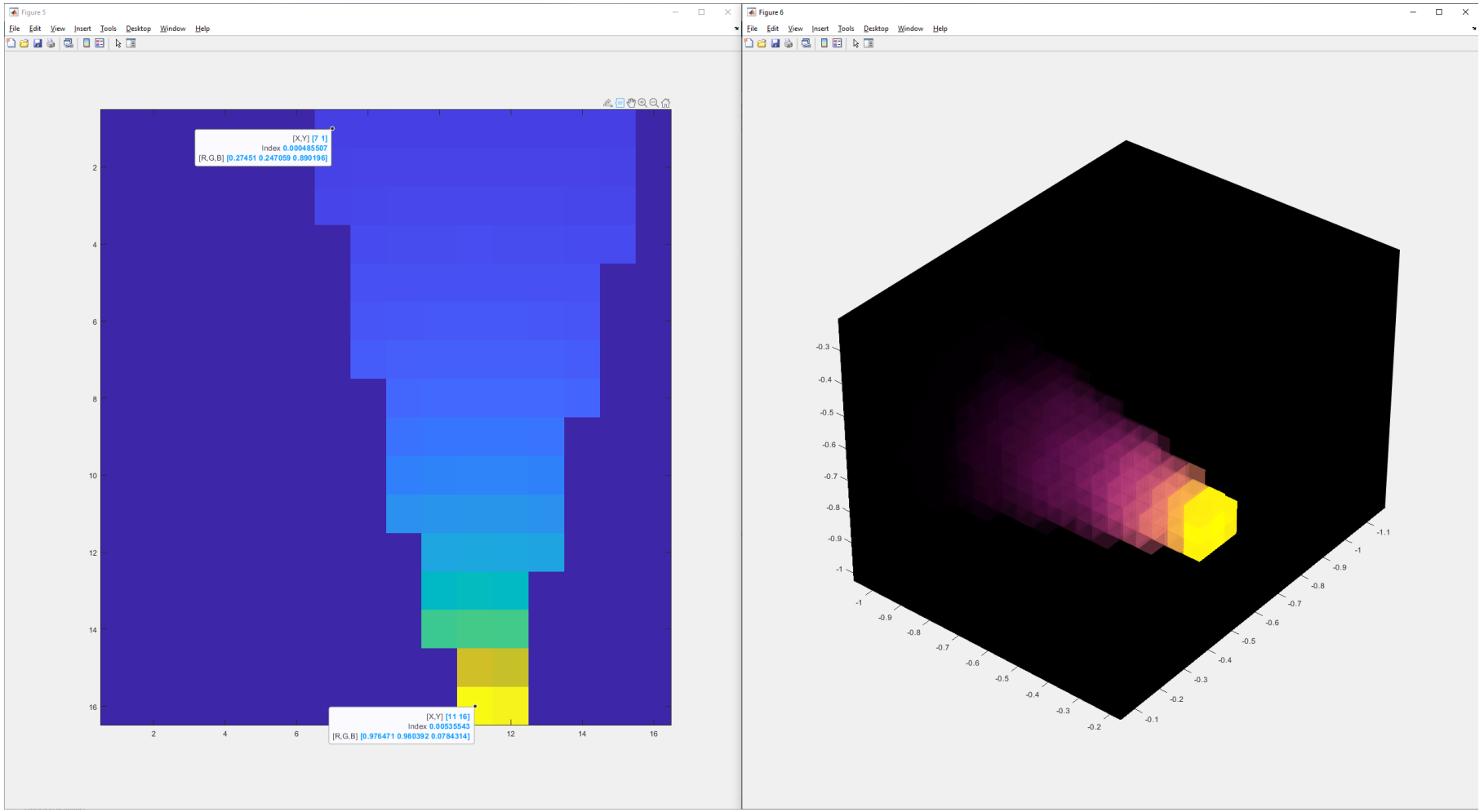
Simple Model

$$M_{ij} = \frac{\cos(\alpha_{ij})}{d_{ij}^2}$$

- Radioactivity contribution from voxel j to detector i

Some Theory

- Contribution field from voxels to a particular pixel;



Some Theory

- $\mathbf{y} = \mathbf{A}\mathbf{x}$ or $y_j = \sum_i A_{ij} x_i$
- Inverse problem: Know y_i, want x_i
- Could do $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$
- But: $\mathbf{A}$ is huge & cannot be inverted
- Solve inverse problem iteratively

- Measurements y_i: independent random variables (**Poisson**)
- Expectation value: $\mu_i = \mathbf{A}_i \cdot \mathbf{x} = \sum_j A_{ij} x_j$
- Probability to measure k given λ: $p(k|\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$

Likelihood:

$$L(\mathbf{x}) = p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{x}) = \prod_i \frac{e^{-\mathbf{A}_i \cdot \mathbf{x}} (\mathbf{A}_i \cdot \mathbf{x})^{y_i}}{y_i!}$$

- Among all possible images $\mathbf{x}$ we choose the one that maximises the probability of producing the data (find most likely image)
- Maximise the log-likelihood, i.e. maximise $\log(p(\mathbf{y}|\mathbf{x}))$

Maximum Likelihood - Expectation Maximisation (ML-EM)

- Objective function to maximise: log-Likelihood (ML)
- Maximisation algorithm: Expectation Maximisation (EM)

- Initial guess for the image (uniform) $x_i^{(0)}$
- Simulate measurements from estimate (forward proj.) $y_j^{\text{simu}} = \sum_k A_{kj} x_k^{(0)}$
- Compare this with actual measurements Ratio $R_j = \frac{y_j}{y_j^{\text{simu}}}$
- Improve image estimate (backward projection) $x_i^{(1)} = x_i^{(0)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} R_j$
- Repeat until convergence

ML-EM

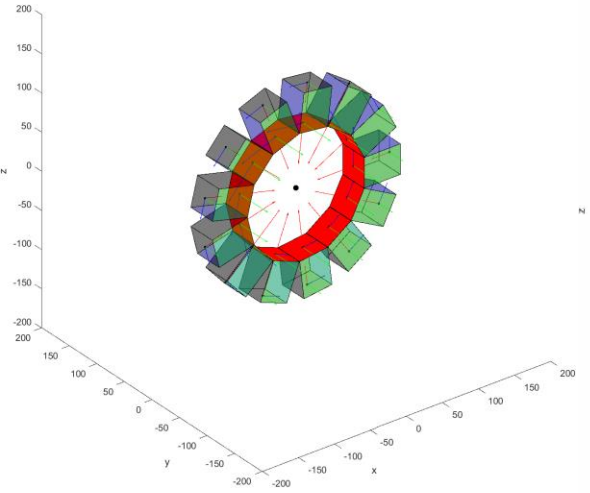
$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

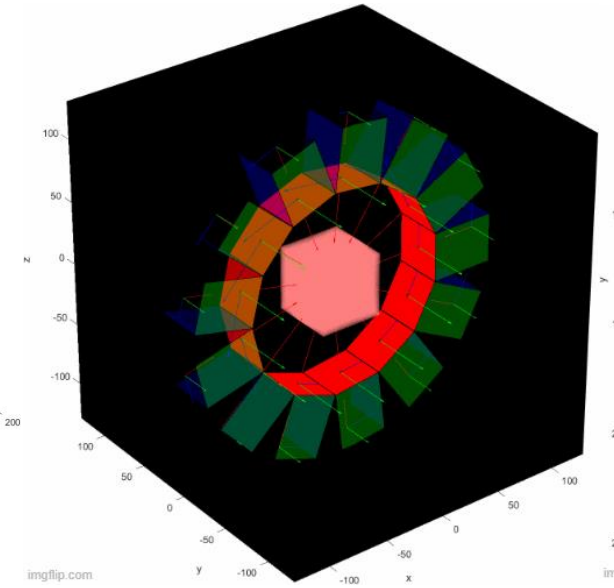
Simple Reconstruction Example | Point source (artificially generated with matlab)

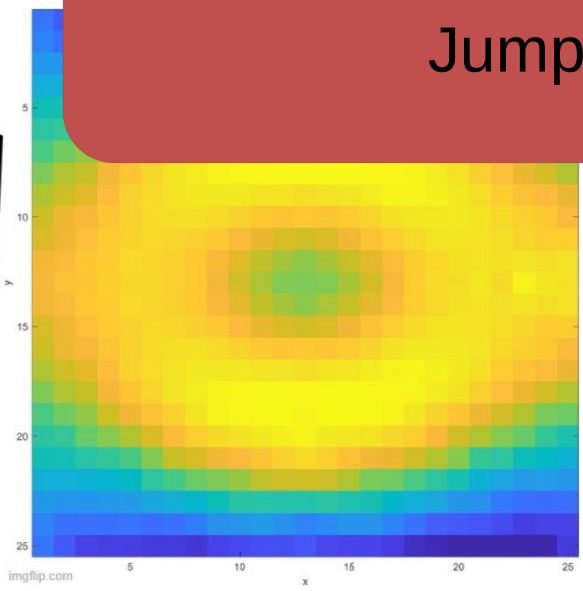
- Full-circular trajectory
- 14 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

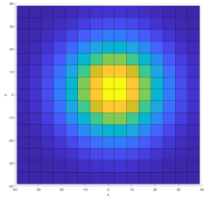
Code had bugs, resolved later
Jump to slide 17







- what every camera would see at every acquisition point;

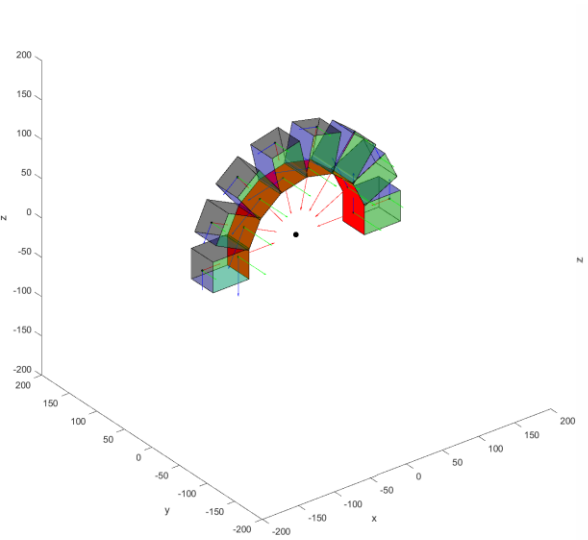


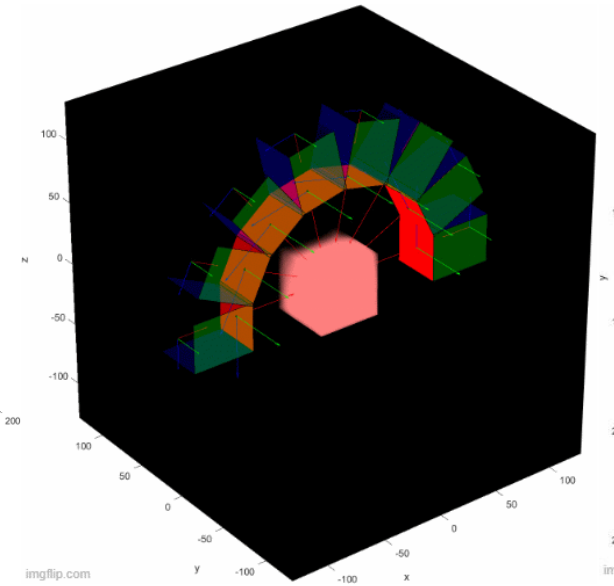
Simple Reconstruction Example | Point source (artificially generated with matlab)

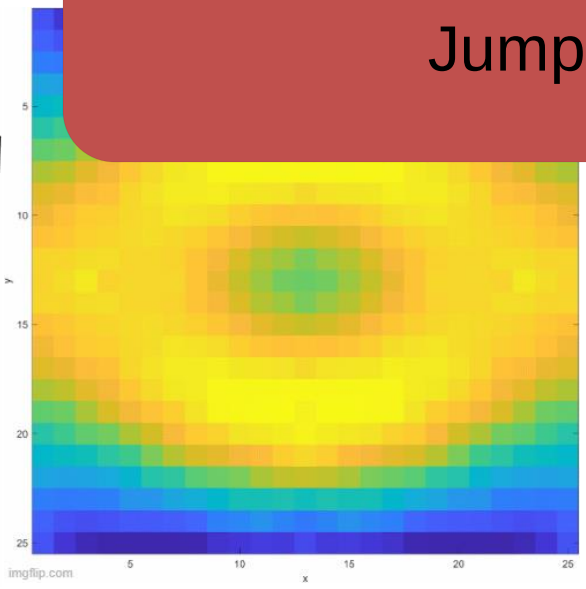
- Semi-circular trajectory
- 8 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

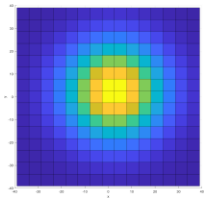
Code had bugs, resolved later,
Jump to slide 17







- what every camera would see at every acquisition point;

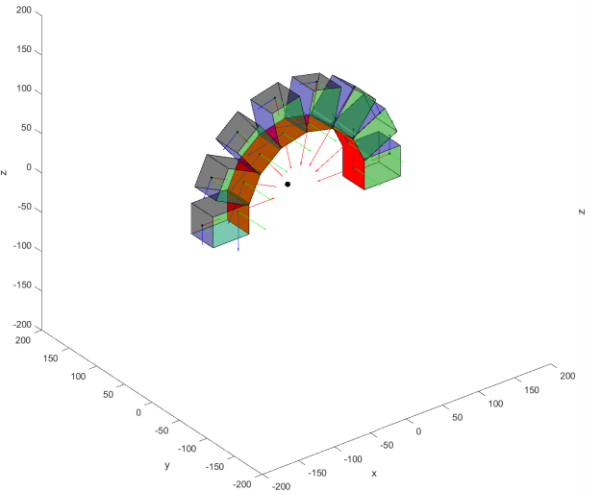


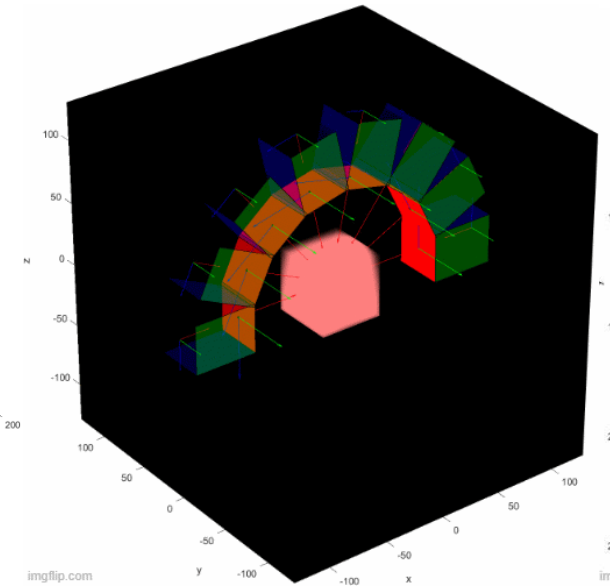
Simple Reconstruction Example | Point source - shifted (artificially generated with matlab)

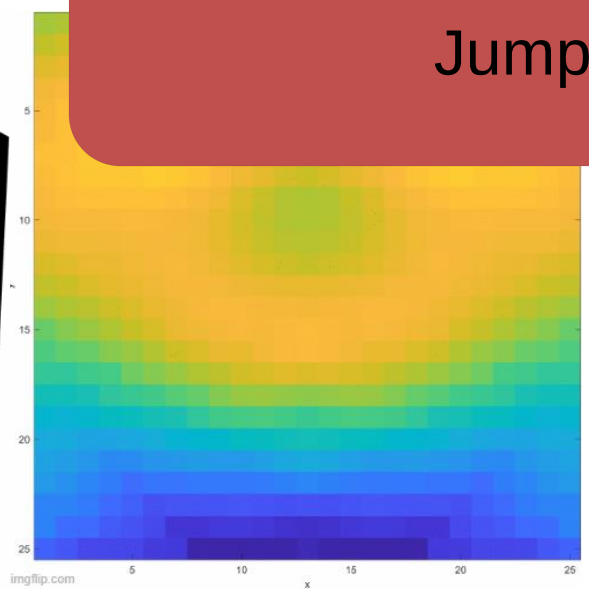
- Semi-circular trajectory
- 8 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

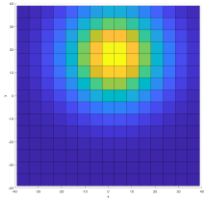
Code had bugs, resolved later,
Jump to slide 17







- what every camera would see at every acquisition point;

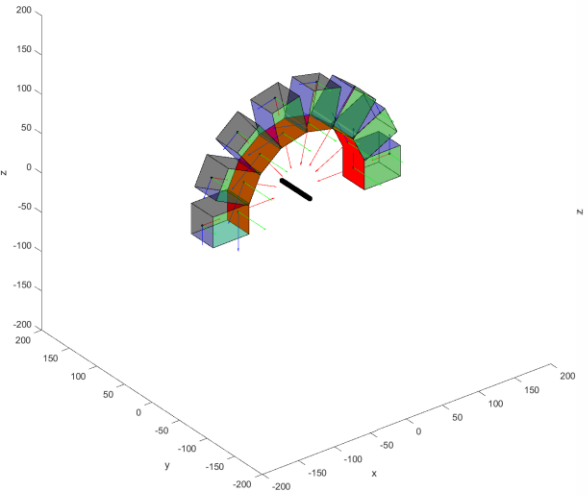


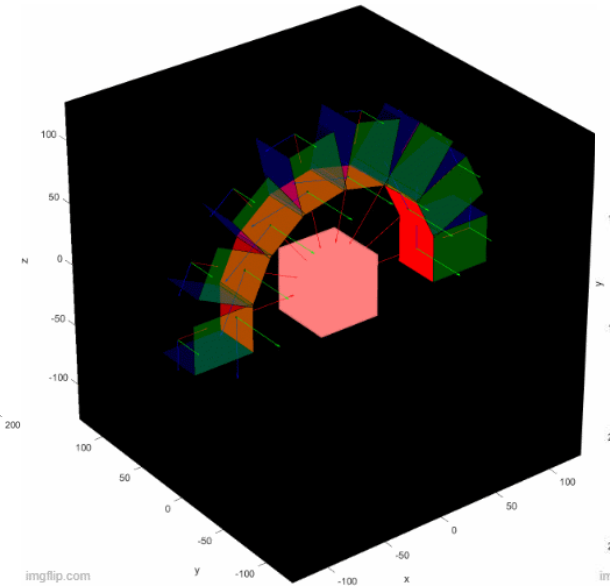
Simple Reconstruction Example | Cylinder source (artificially generated with matlab)

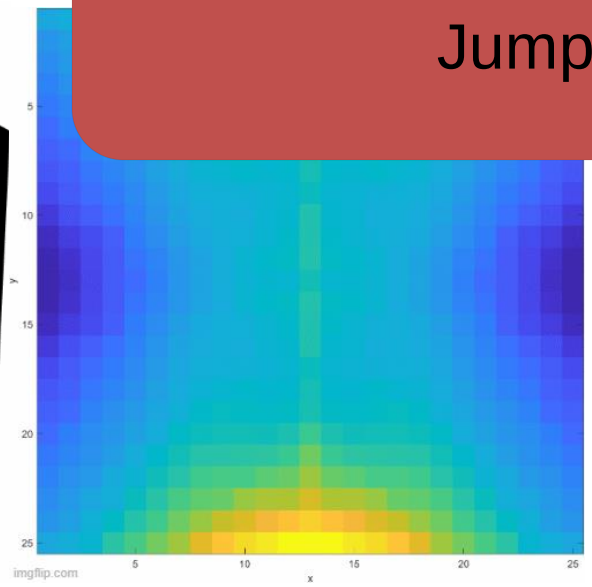
- Semi-circular trajectory
- 8 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

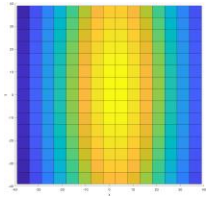
Code had bugs, resolved later,
Jump to slide 17







- what every camera would see at every acquisition point;

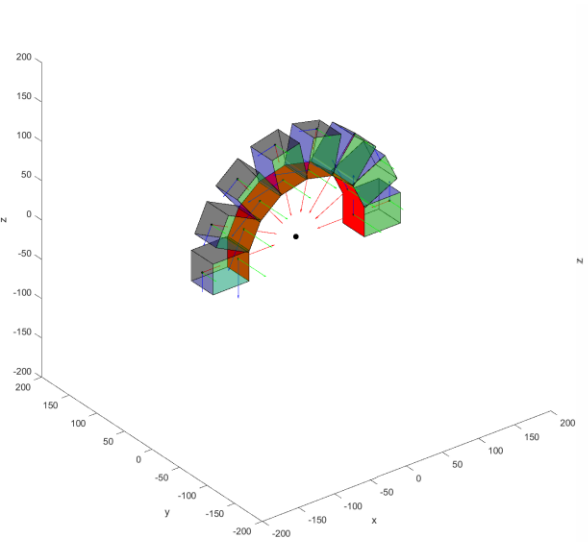


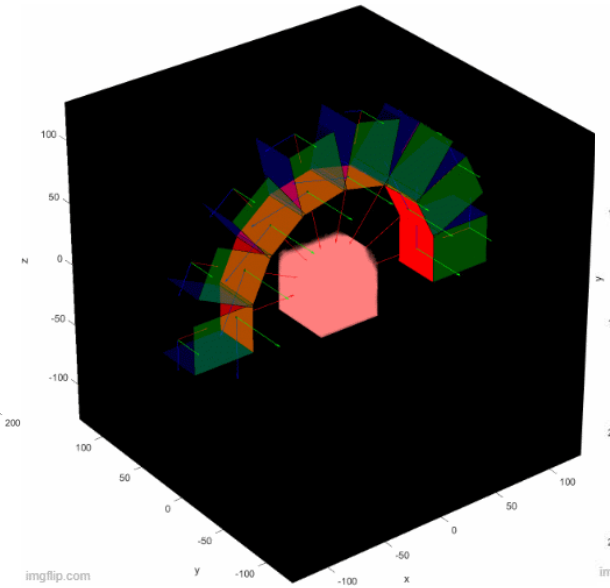
Simple Reconstruction Example | Point source (with Gate simulations)

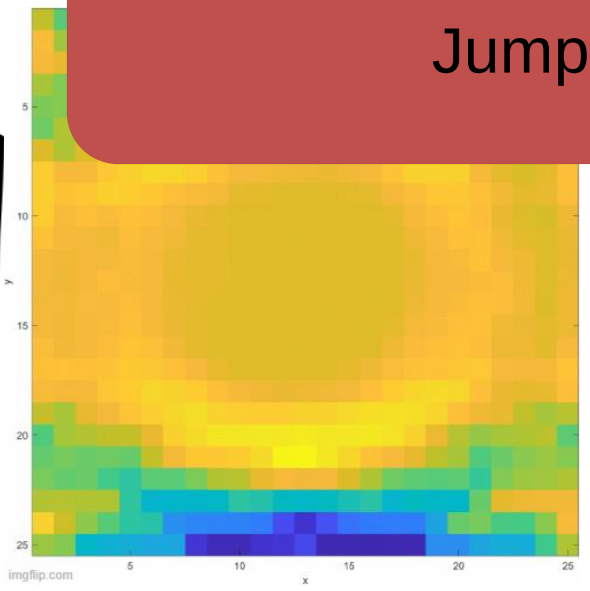
- Semi-circular trajectory
- 8 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

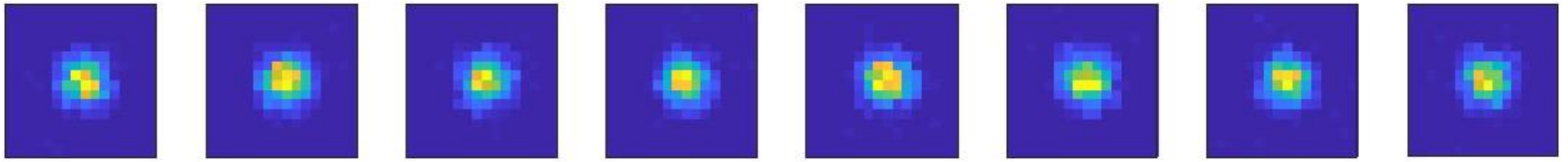
Code had bugs, resolved later,
Jump to slide 17







- 8 samples from Gate simulations, corresponding to acquisition points;

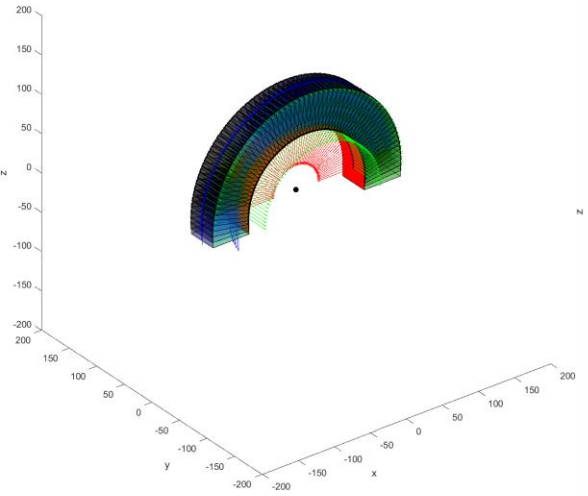


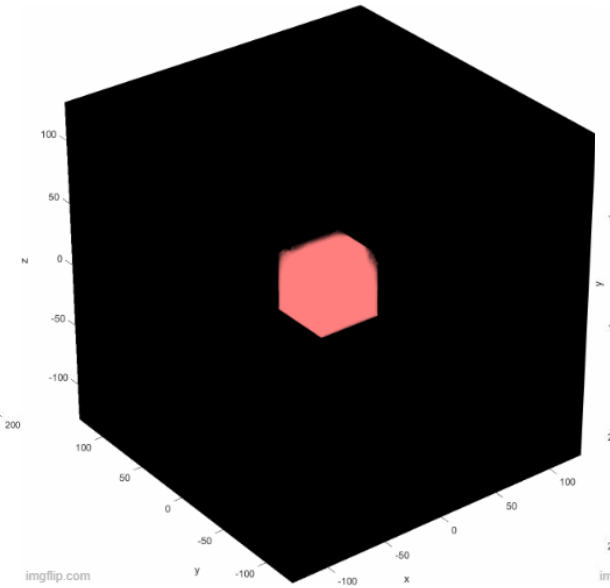
Simple Reconstruction Example | Point source (with Gate simulations)

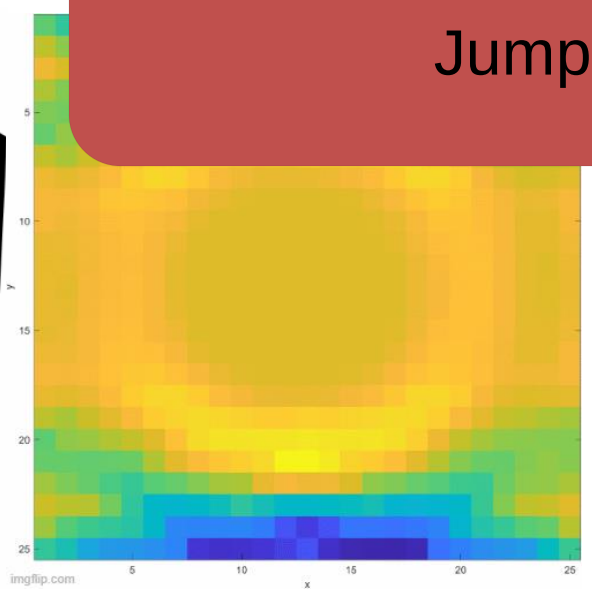
- Semi-circular trajectory
- 60 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

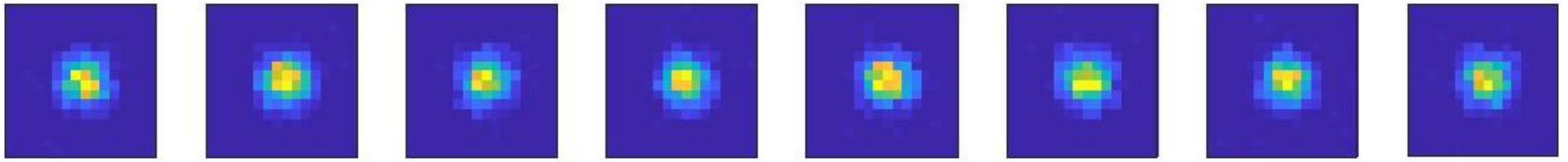
Code had bugs, resolved later,
Jump to slide 17







- .. some 8 samples from Gate simulations, corresponding to acquisition points;

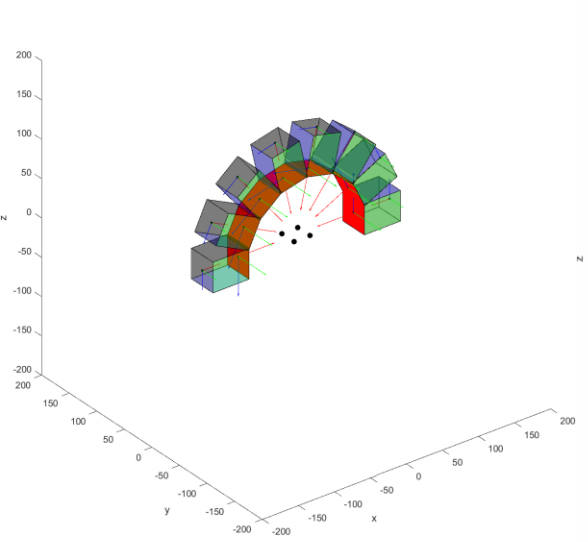


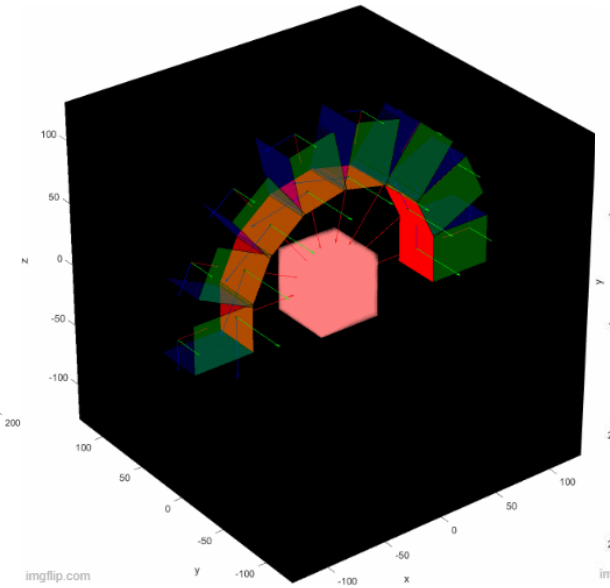
Simple Reconstruction Example | 4-Point sources (with Gate simulations)

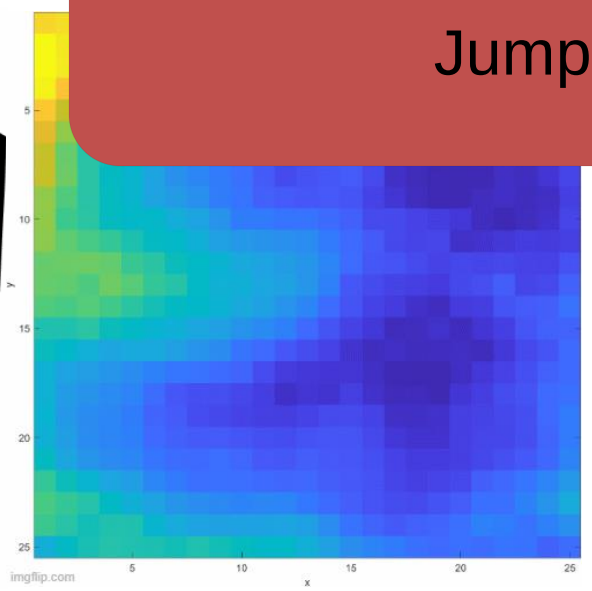
- Semi-circular trajectory
- 8 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

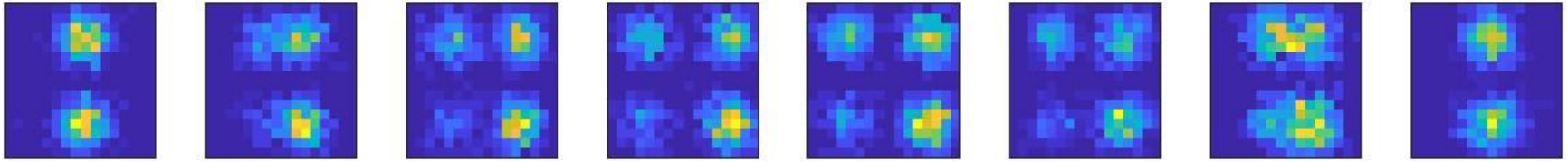
Code had bugs, resolved later,
Jump to slide 17







- 8 samples from Gate simulations, corresponding to acquisition points;

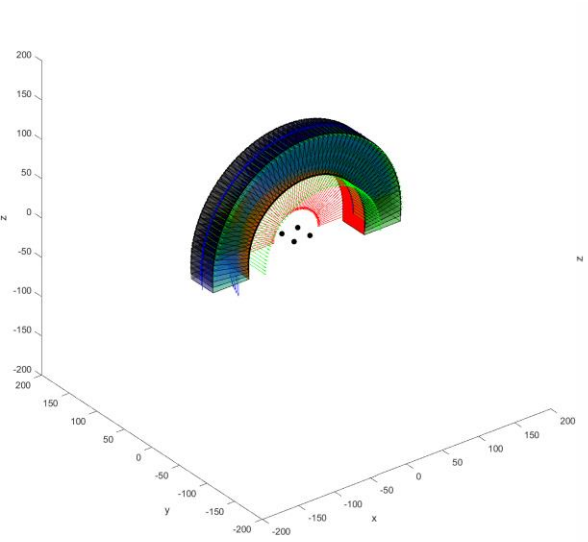


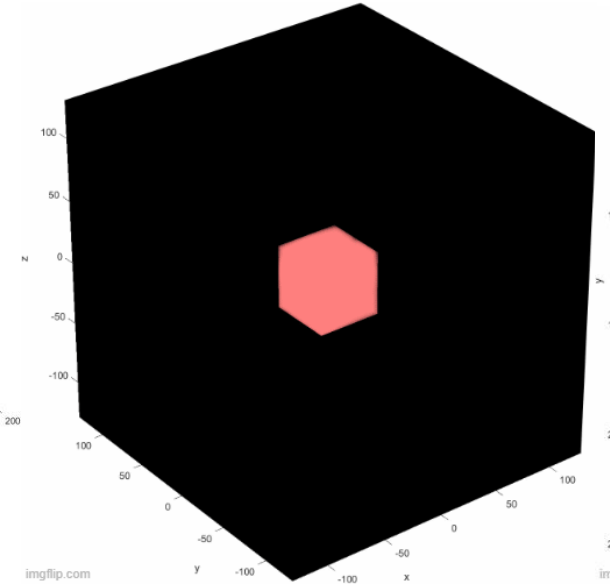
Simple Reconstruction Example | 4-Point sources (with Gate simulations)

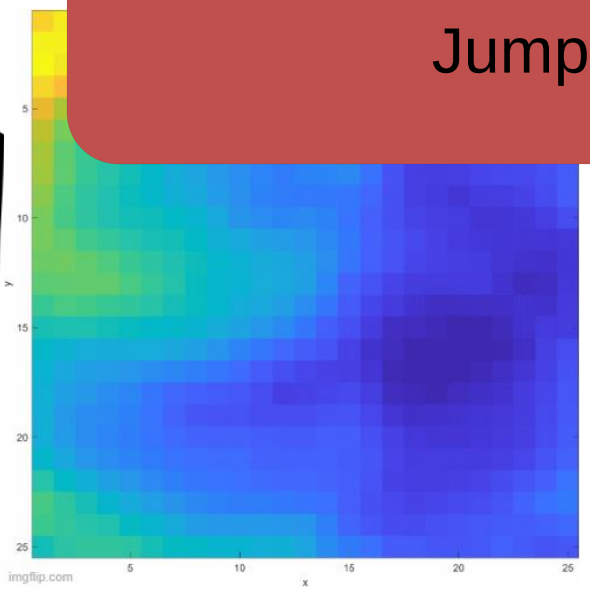
- Semi-circular trajectory
- 60 acquisitions

- Grid size: 50x50x50 mm
- Voxel size: 2 mm

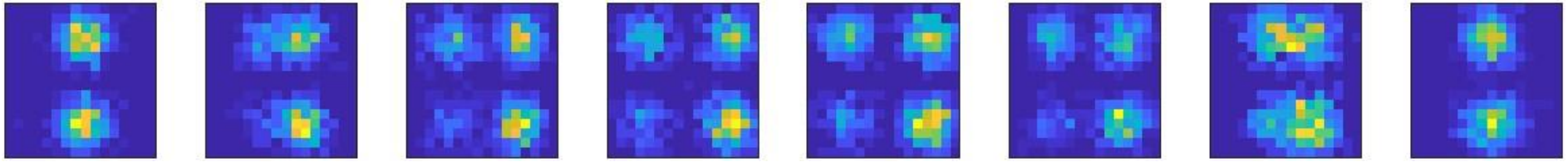
Code had bugs, resolved later,
Jump to slide 17







- .. some 8 samples from Gate simulations, corresponding to acquisition points;



What's next?

- Holes that are seen in the reconstructions should not happen
- There could be a problem with incorrect traversing of arrays or incorrect camera–voxel distances

High Priority;
- [Bugfix] Consider also the 1cm of crystal depth when building the system matrix
- [Debug] Make reconstruction volume bigger, crop it so that we only have meaningful visuals for the region of interest
- [Debug] Work with less complicated scene (3x3 cam)
- [Debug] Compare forward projection with camera acquisitions

Medium Priority;
- Tabulate the system matrix, using Gate simulations

Low Priority;
- Try more complex system modellings
- Gate simulations with the trajectory obtained from NDI tracking around neck
- Export reconstruction as dicom file

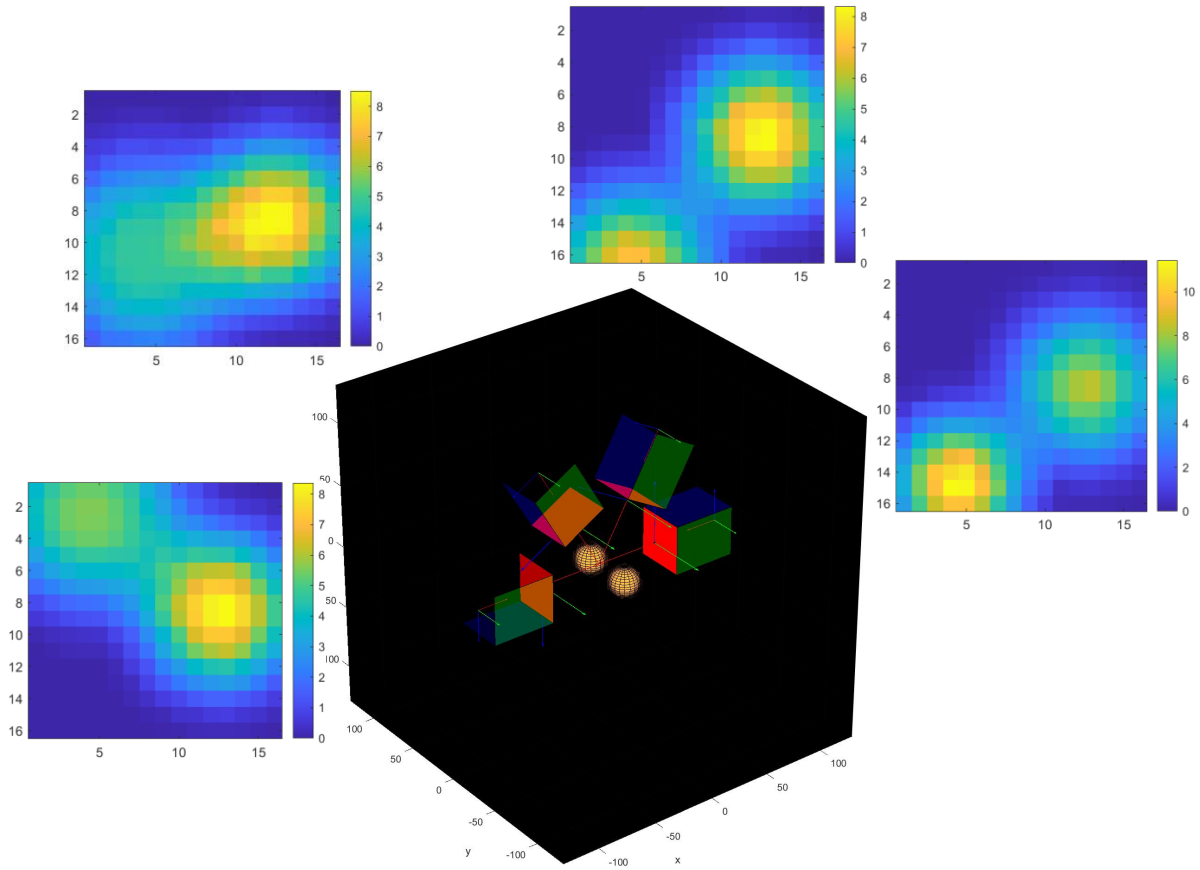
Sanity check

- Experiment workflow

1. Define the radioactivity field **f_ref**
   (known voxel grid as reference)
2. Build the system matrix **H**
   (fixed for the scene)
3. Forward projection to get camera acquisitions **g_ref** = **H** * **f_ref**
   (known cam acqusitions as reference)
4. Use the reference **g_ref** and the same system matrix **H**
   to do the reconstruction **f_recons**
   (solve for **g_ref** = **H** * **f_recons**)

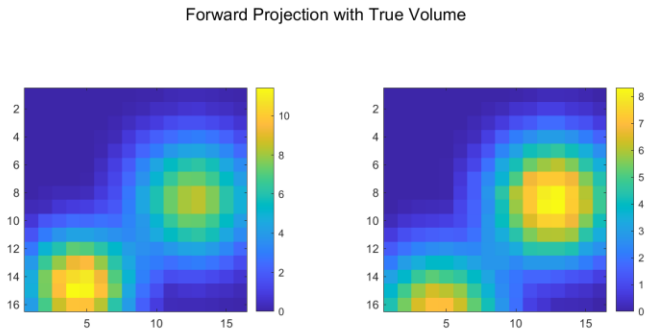
Forward projection on the reference radioactivity field **f_ref**

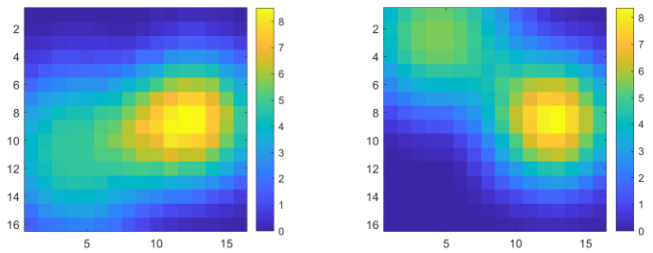
- Reference camera acquisitions: **g_ref**



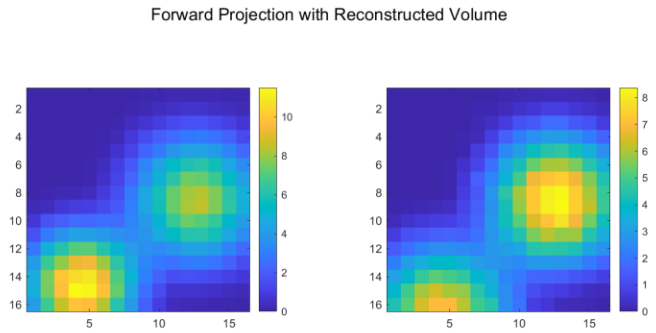
Sanity check

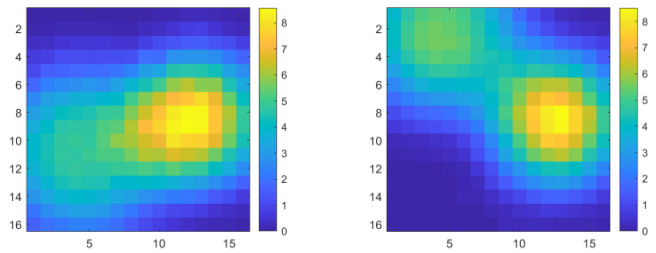
Forward projection comparison



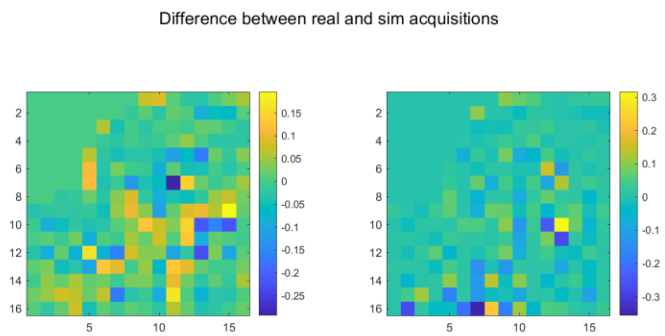


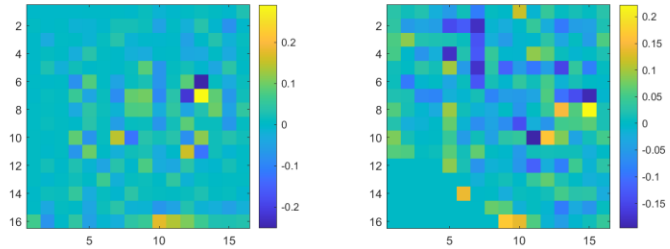
g_ref





g_sim

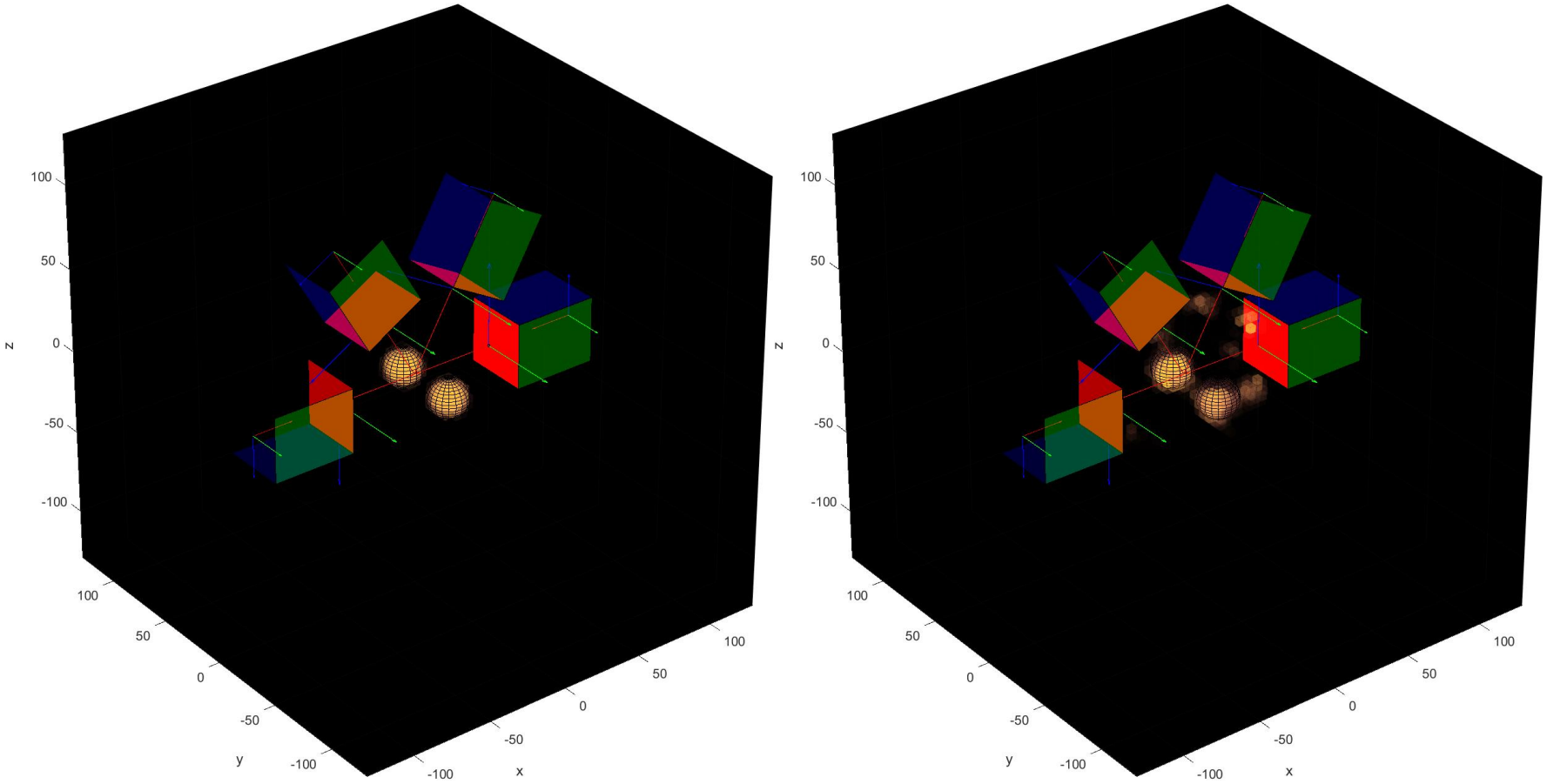




g_diff

Sanity check

Radioactivity field comparison



f_ref | **f_recons**

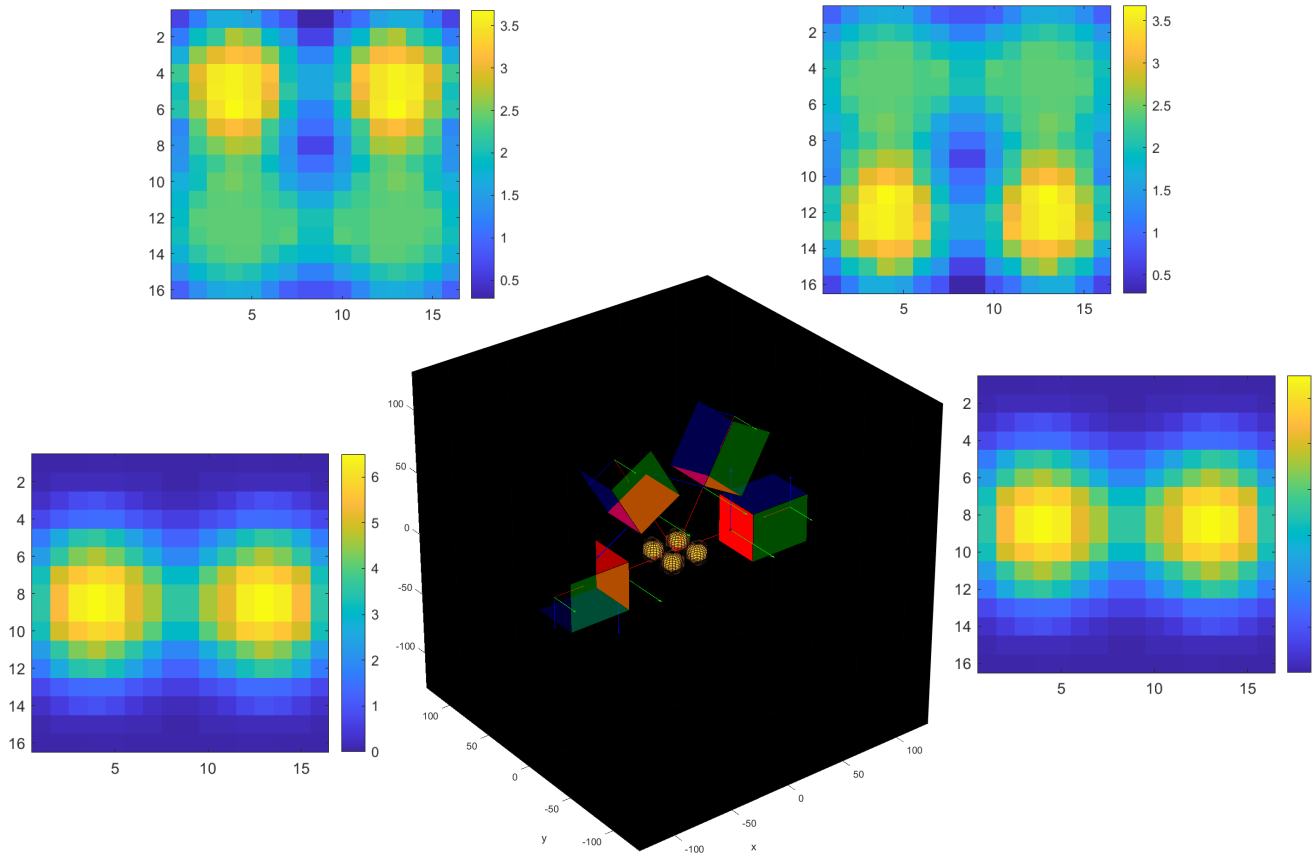
Sanity check

- Experiment workflow

1. Define the radioactivity field **f_ref**
   (known voxel grid as reference)
2. Build the system matrix **H**
   (fixed for the scene)
3. Forward projection to get camera acquisitions **g_ref** = **H** * **f_ref**
   (known cam acqusitions as reference)
4. Use the reference **g_ref** and the same system matrix **H**
   to do the reconstruction **f_recons**
   (solve for **g_ref** = **H** * **f_recons**)

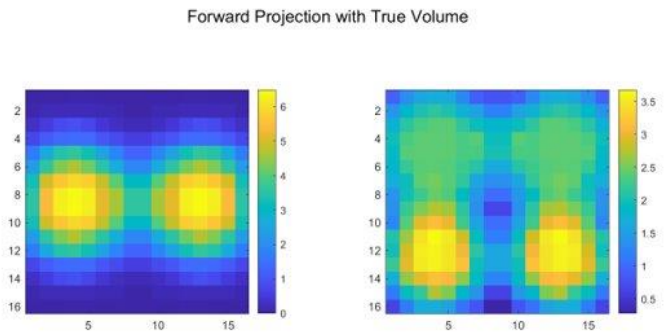
Forward projection on the reference radioactivity field **f_ref**

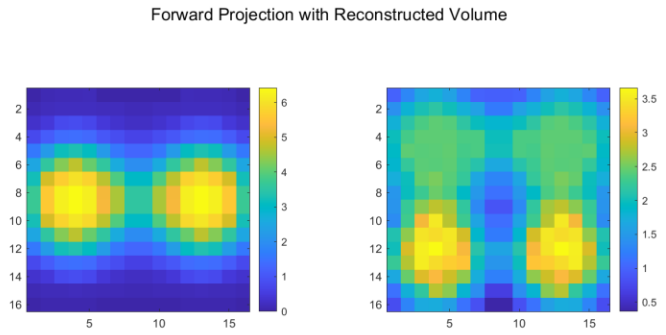
- Reference camera acquisitions: **g_ref**

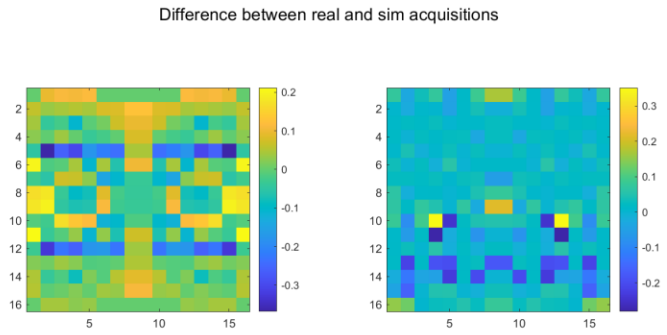


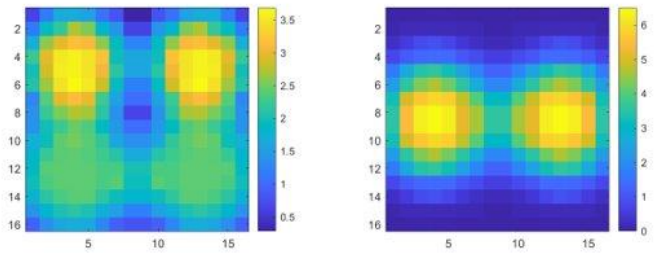
Sanity check

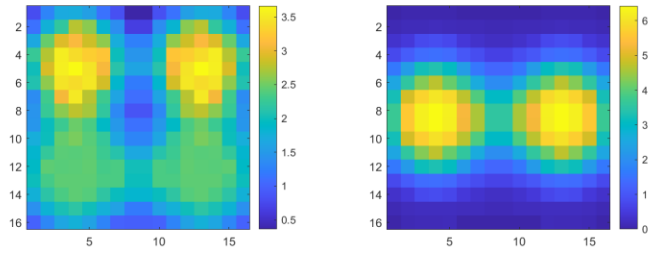
Forward projection comparison

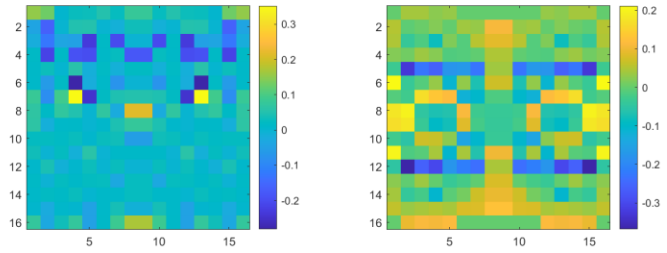








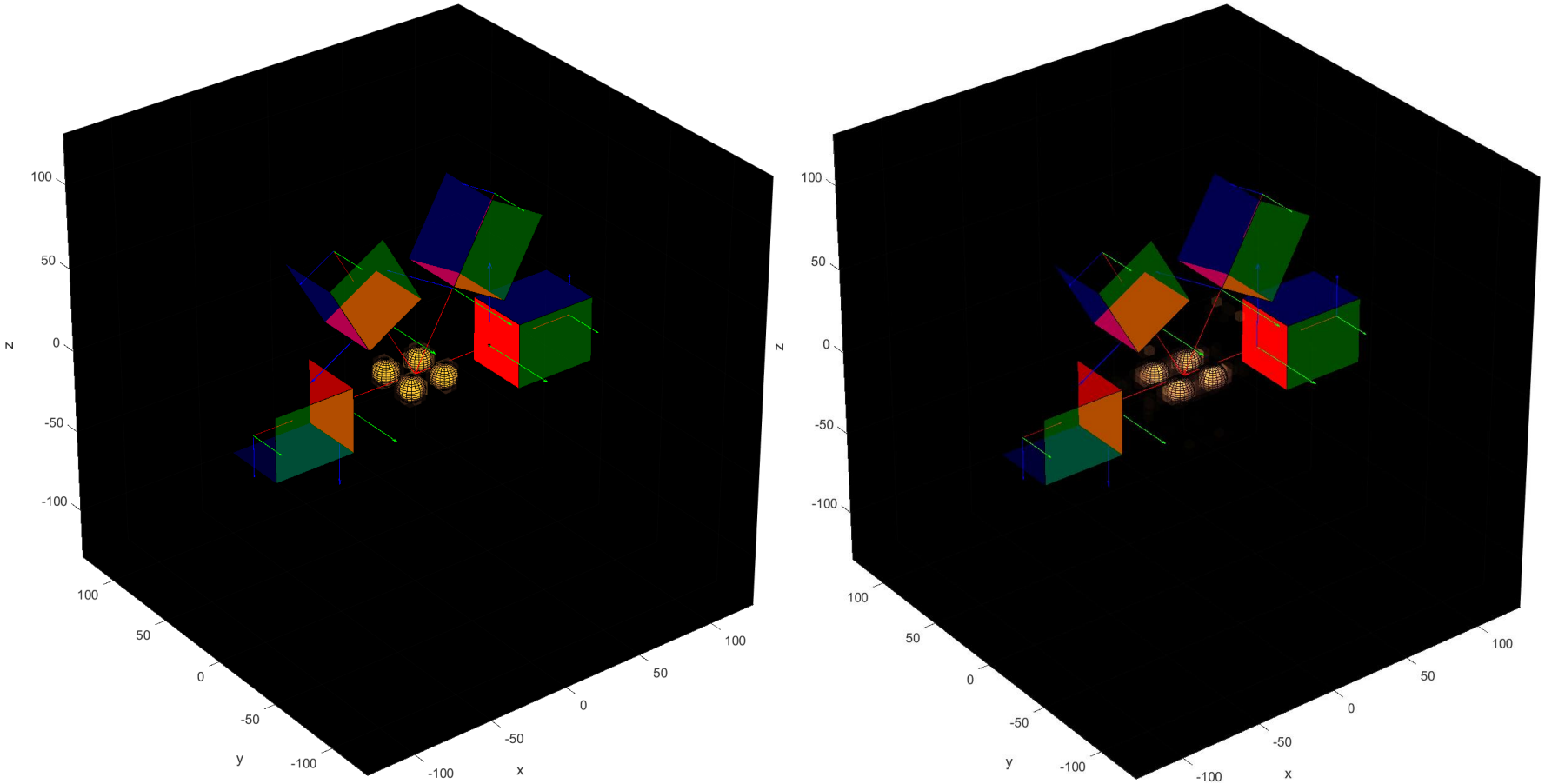




g_ref | **g_sim** | **g_diff**

Sanity check

Radioactivity field comparison



f_ref | **f_recons**

What's next?

- Sanity check works
- Reconstruction now runs slow due to larger reconstruction volume and more camera acquisitions... ~*O*(M * N^3)

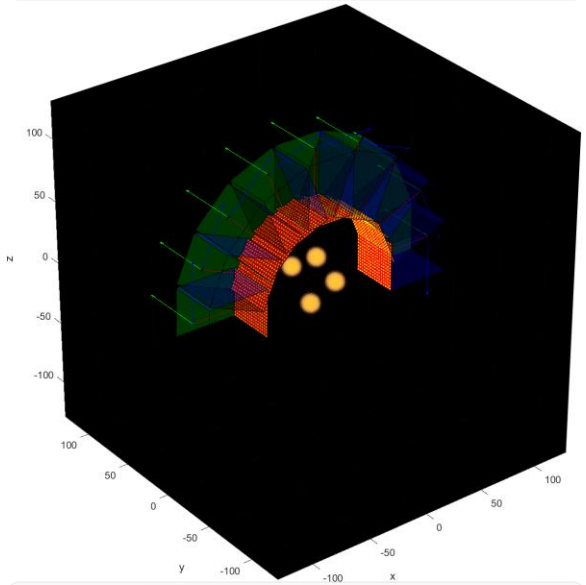
High Priority;
- Refactor Matlab code in a way that it runs on GPU
- Do reconstructions using Gate simulations; single point, 4 points …

Medium Priority;
- Tabulate the system matrix, using Gate simulations
- Do reconstructions using Gate simulations of thyroid with multiple cameras
- Gate simulations with the trajectory obtained from NDI tracking around neck

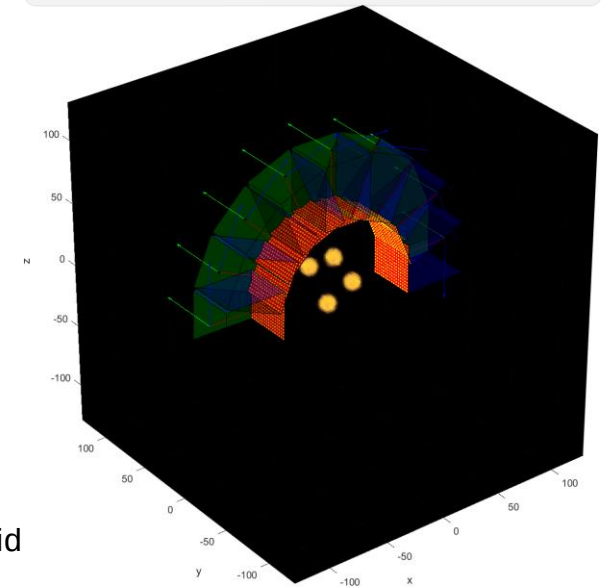
Low Priority;
- Try more complex system modellings
- Export reconstruction as .dicom file

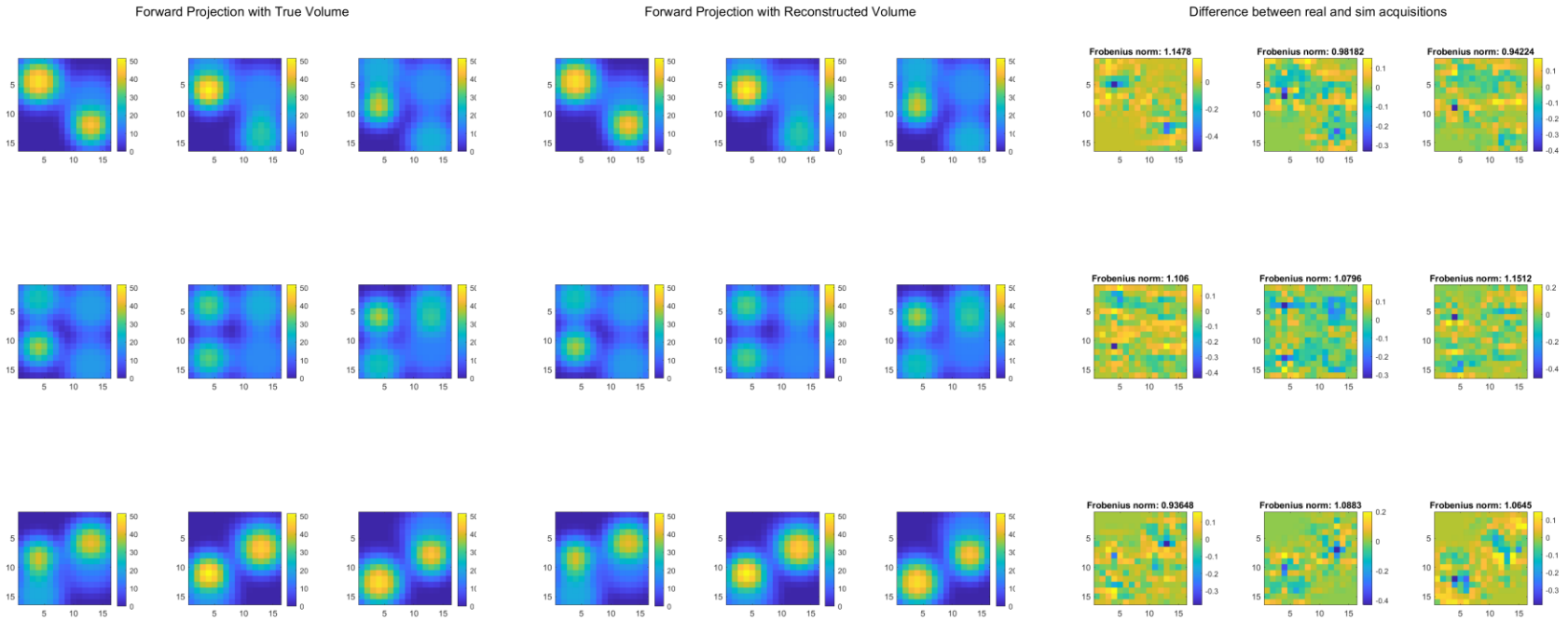
Sanity check.. but on GPU this time

f_ref



f_recons



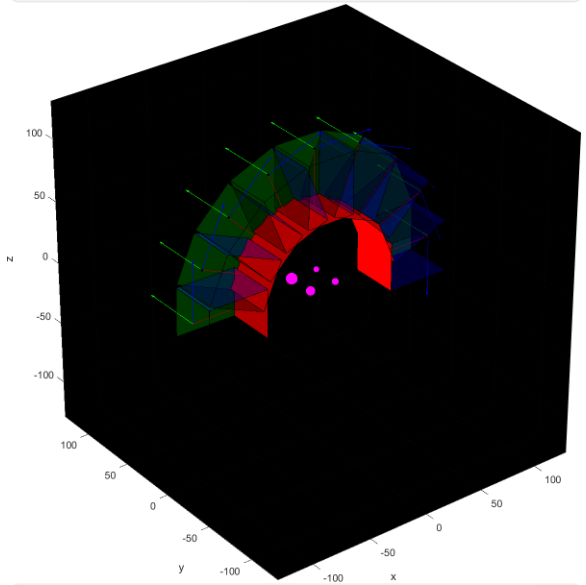


- Time elapsed for (9 cam acquisition, [32, 32, 32] voxel grid with 2mm voxel size);
  - Building the system matrix: **20 secs** (~**2.2 secs** per cam acquisition)
  - Solving the system : **35 secs** @ 2k iteration (~**0.02 secs** per iteration)

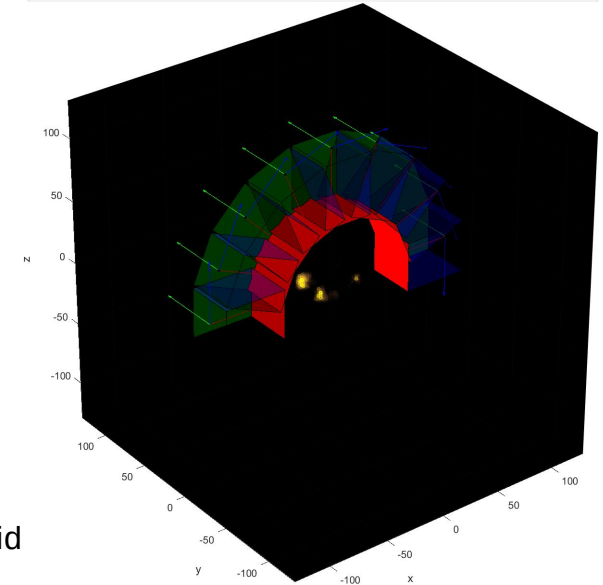
- Before it was..
  - Building the system matrix: **176 secs** (~**19.5 secs** per cam acquisition)
  - Solving the system : **550 secs** @ 2k iteration (~ **0.27 secs** per iteration)

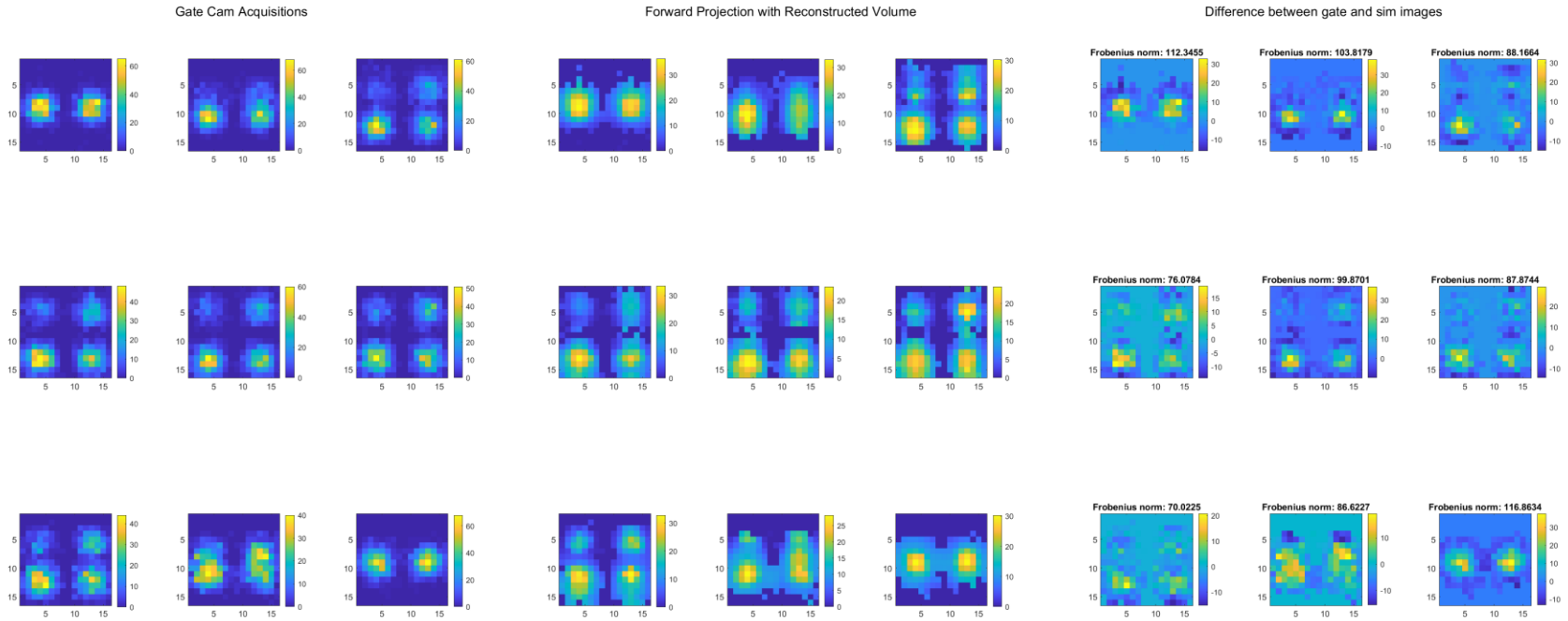
3D Reconstructions with Gate Simulations | 4 point-sources

scene



reconstructed volume





➢ **Gate dataset properties**

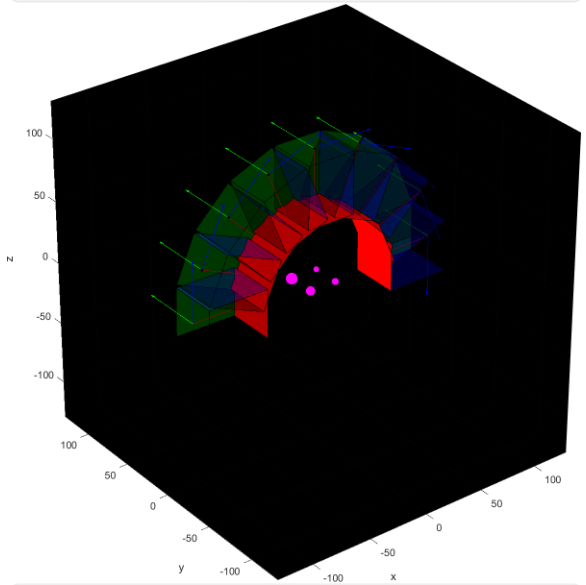
- 360 seconds of total acquisition
- 180 degrees of rotation (0.5 deg/s)
- Period of 0.25 secs for each slice (1440 slices in total)
- Point activities of {0.4, 0.3, 0.2, 0.1} MBq

➢ **Reconstruction scene properties**

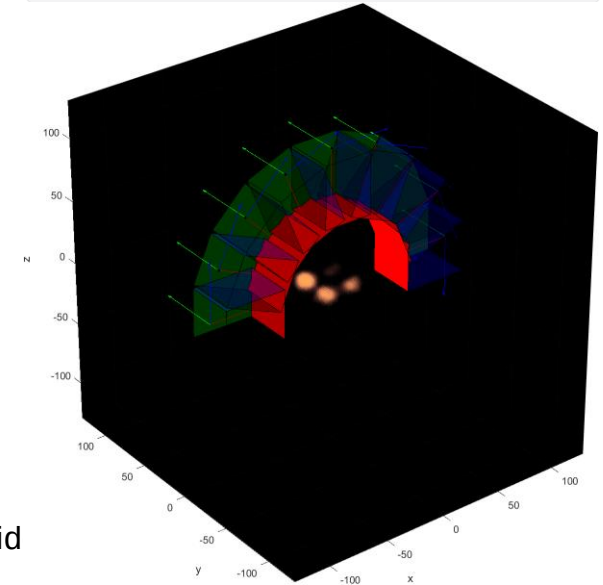
- [32, 32, 32] voxel grid with 2mm voxel size
- 9 camera acquisition
- Each camera acquisition;
  - 4 seconds of integration
  - (16 slice, +/- 1 degree)
  - No further post-processing

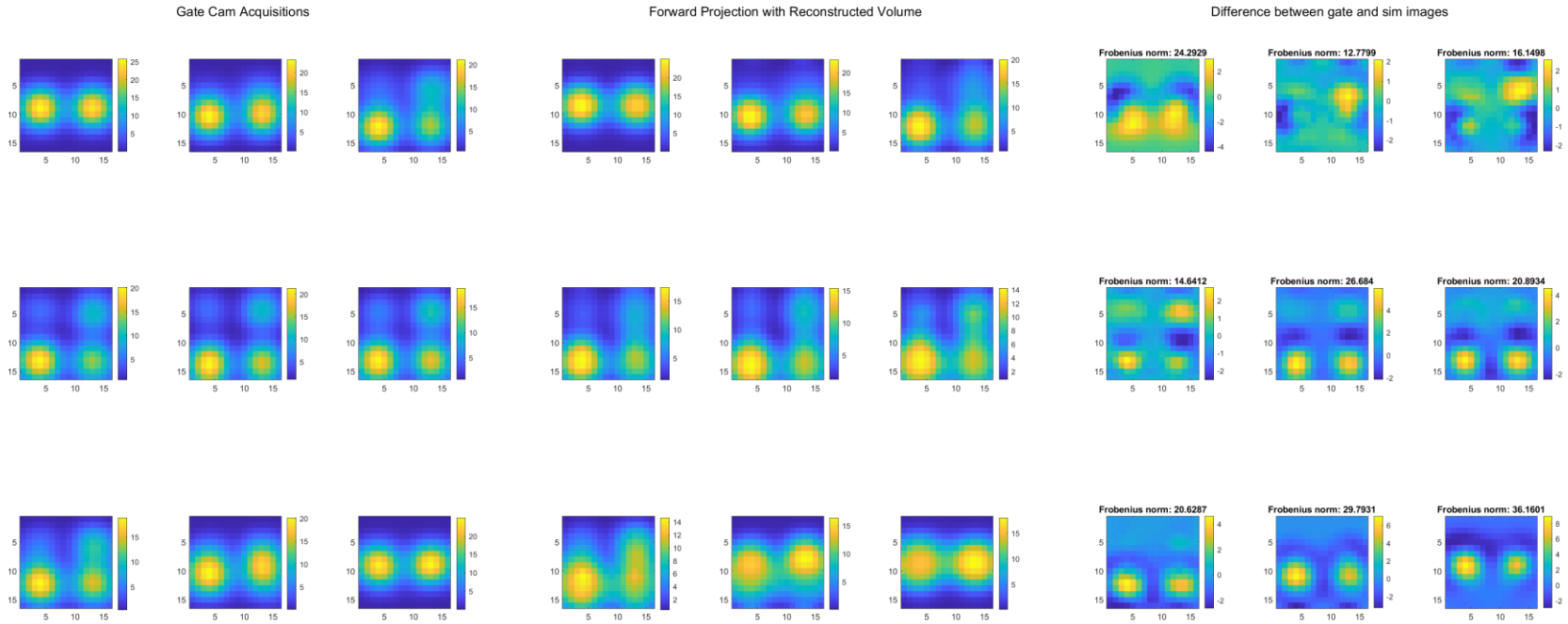
3D Reconstructions with Gate Simulations | 4 point-sources, smoothened

scene



reconstructed volume





➢ **Gate dataset properties**

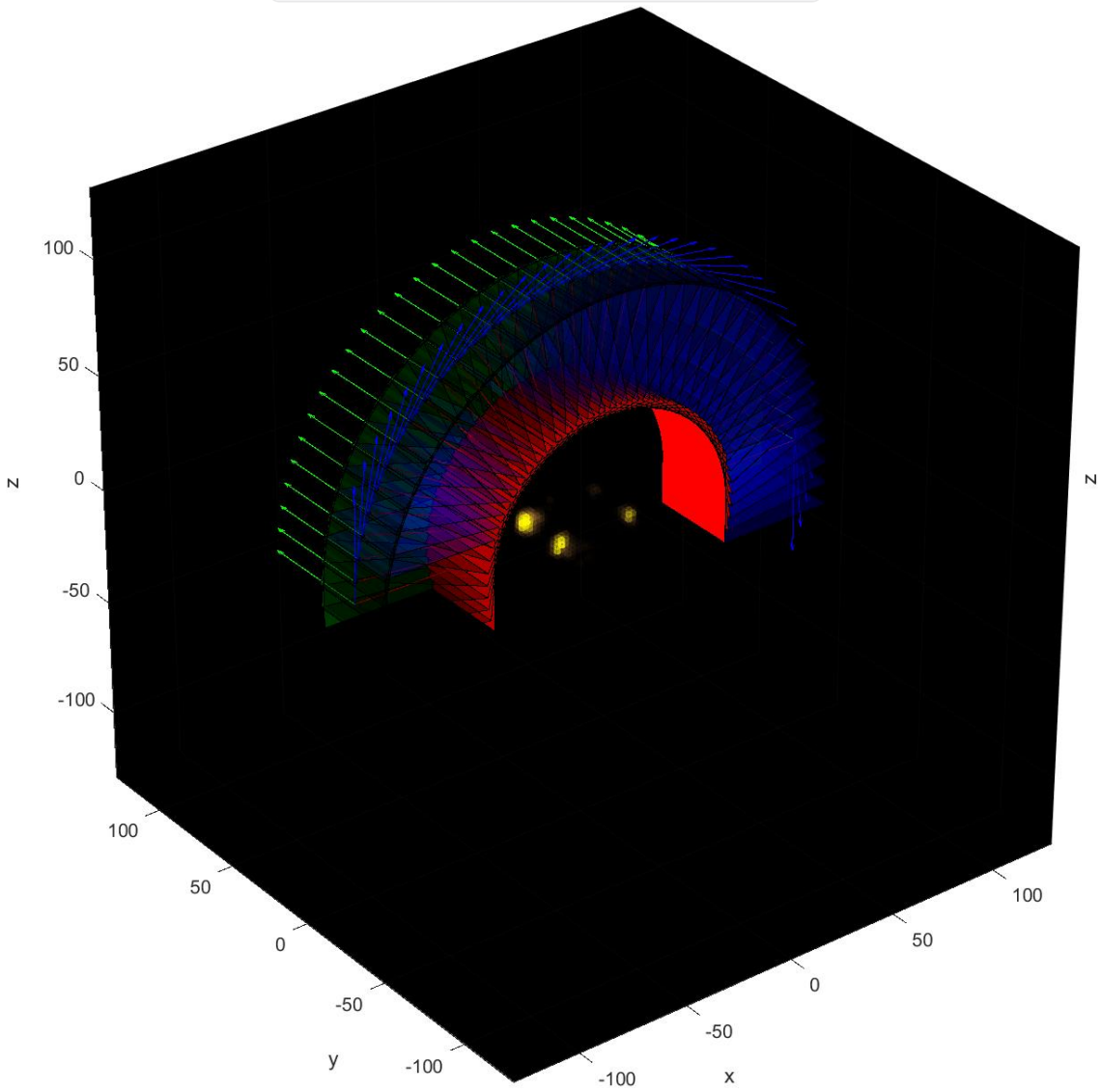
- 360 seconds of total acquisition
- 180 degrees of rotation (0.5 deg/s)
- Period of 0.25 secs for each slice (1440 slices in total)
- Point activities of {0.4, 0.3, 0.2, 0.1} MBq

➢ **Reconstruction scene properties**

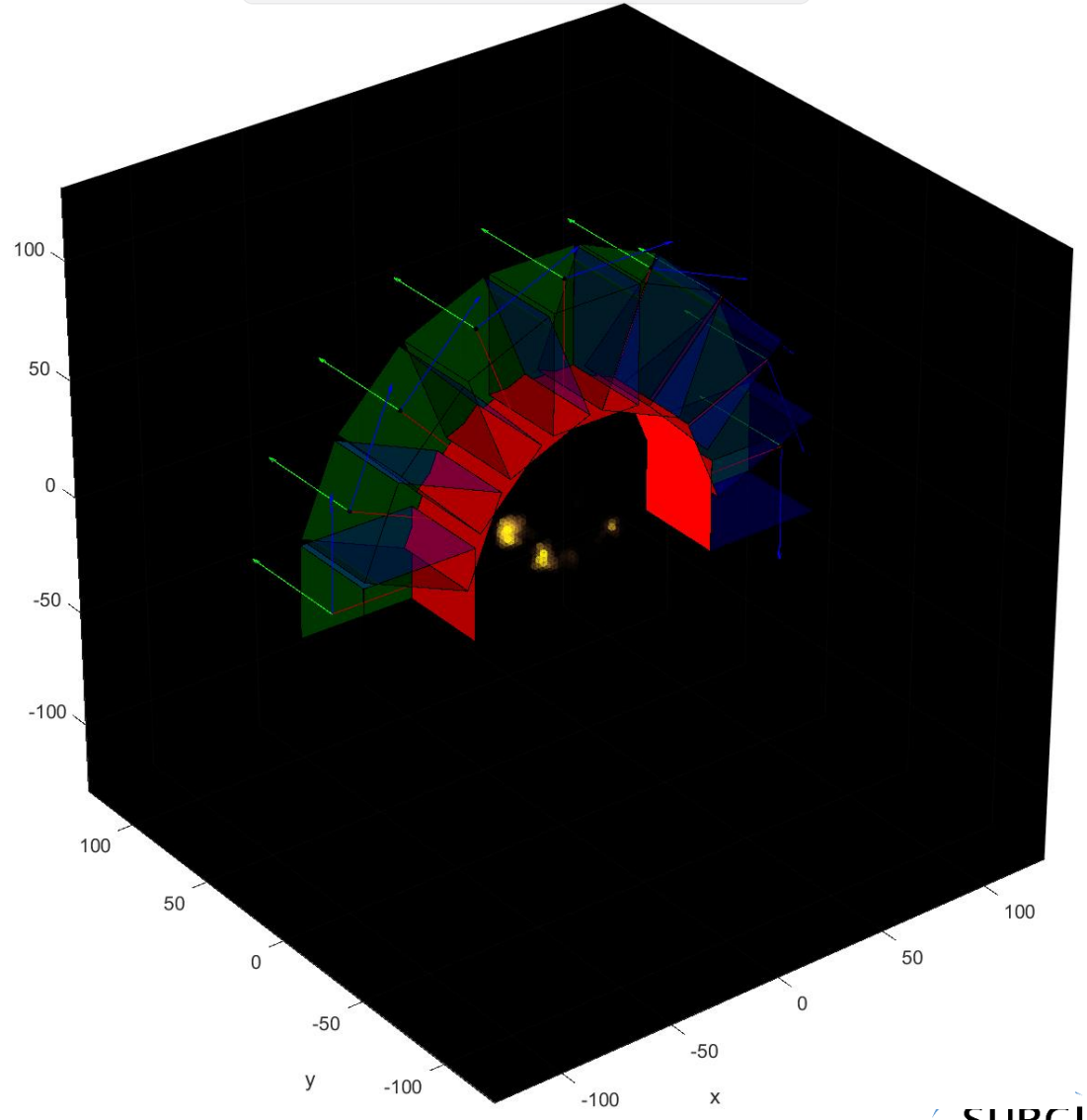
- [32, 32, 32] voxel grid with 2mm voxel size
- 9 camera acquisition
- Each camera acquisition;
  - 4 seconds of integration
  - (16 slice, +/- 1 degree)
  - Smoothened

3D Reconstructions with Gate Simulations | 4 point-sources, more cam acquisitions

36 cam acquisitions



9 cam acquisitions



What's next?

- Gate simulations are not smooth compared to artificial ones previously generated in Matlab
  - Do we need more acquisition time?
  - Do we need to apply filters?
  - Would multi-camera configuration with the same acquisition time help with better reconstruction?
- Code runs slow with higher number of camera acquisitions
  - The bottleneck is the data copy overhead (between RAM and GPU memory)
  - Matlab works with double precision by default... maybe use single precision (float instead of double)

High Priority;
- Do reconstructions using Gate simulations with single-/multi-camera settings
- Tabulate the system matrix, using Gate simulations

Medium Priority;
- Implement image filters (e.g., edge-preserving smoothing)
- Refactor Matlab code to change data types to float
- Try more complex system modellings; include attenuation, scattering

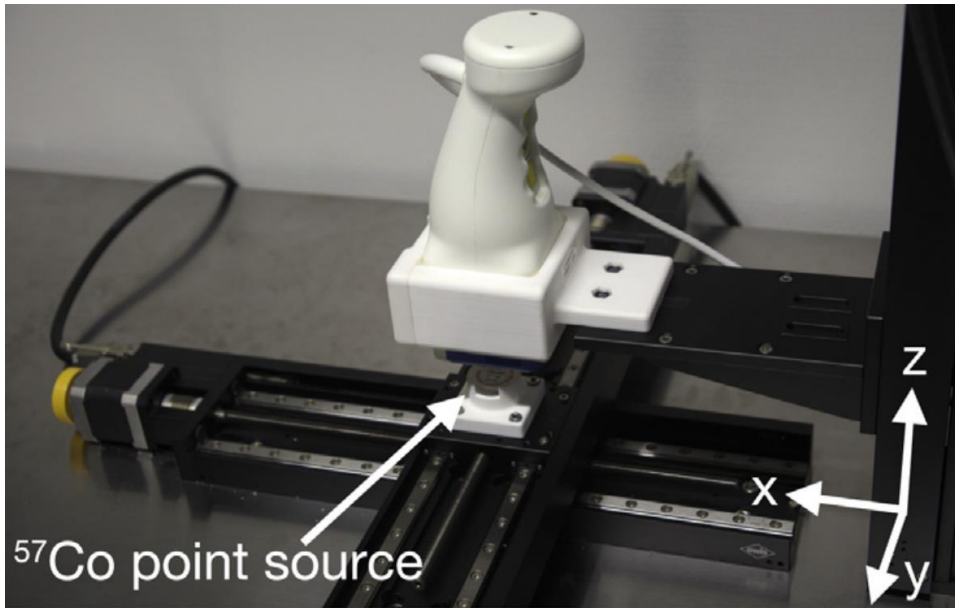
Low Priority;
- Gate simulations with the trajectory obtained from NDI tracking around neck
- Export reconstruction as .dicom file

LUT via Gate Simulations

➢ **Gate simulation dataset properties;**

- 1 mm resolution
- Relative positions of point sources to detector pixel;
  x, y = -10 : 1 : 10 [mm]
  z = 15 : 1 : 165 [mm]
- Number of Gate simulations = 66150 (21x21x150)
- Point activity = 0.1 MBq
- Simulation stopping criteria = 60 seconds of acquisition
- Energy window = 130-150 keV for Tc99m
- Collimator length = 11.15 mm

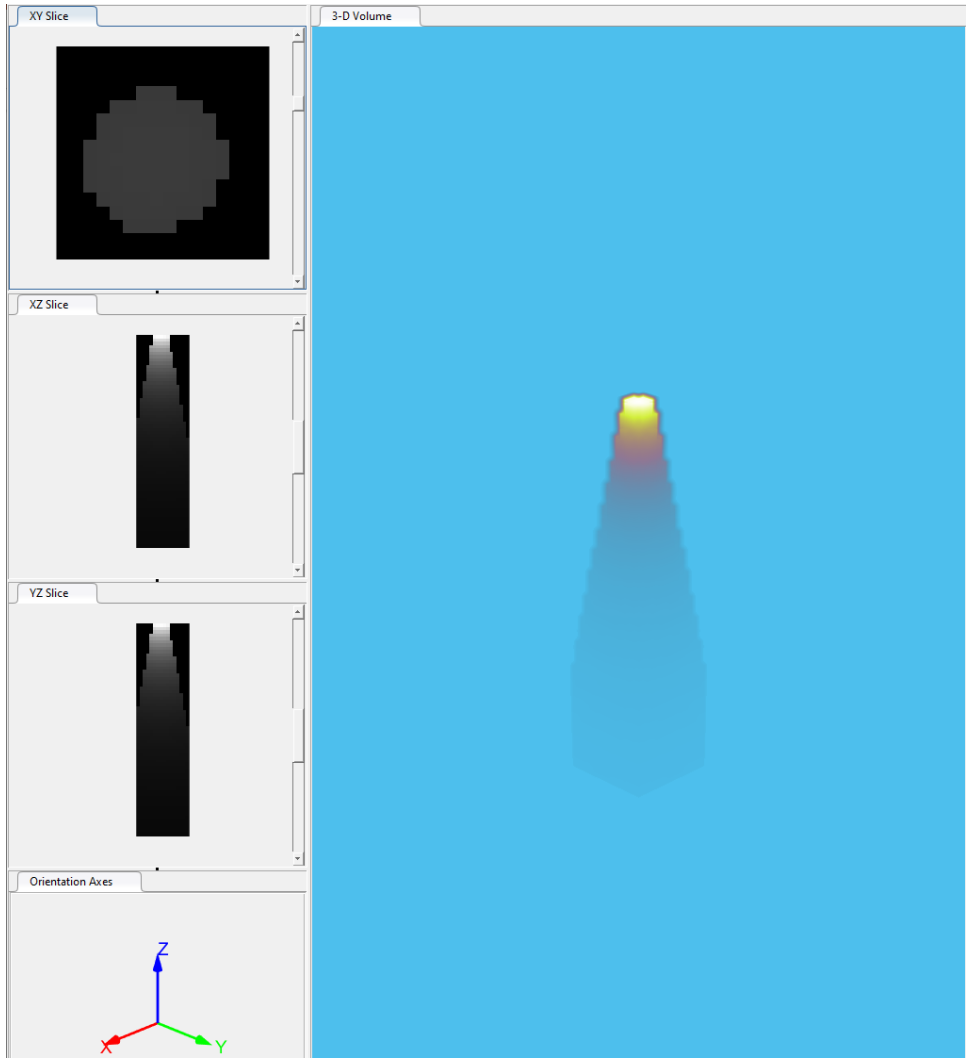
➢ **Real setup previously done in [1];**



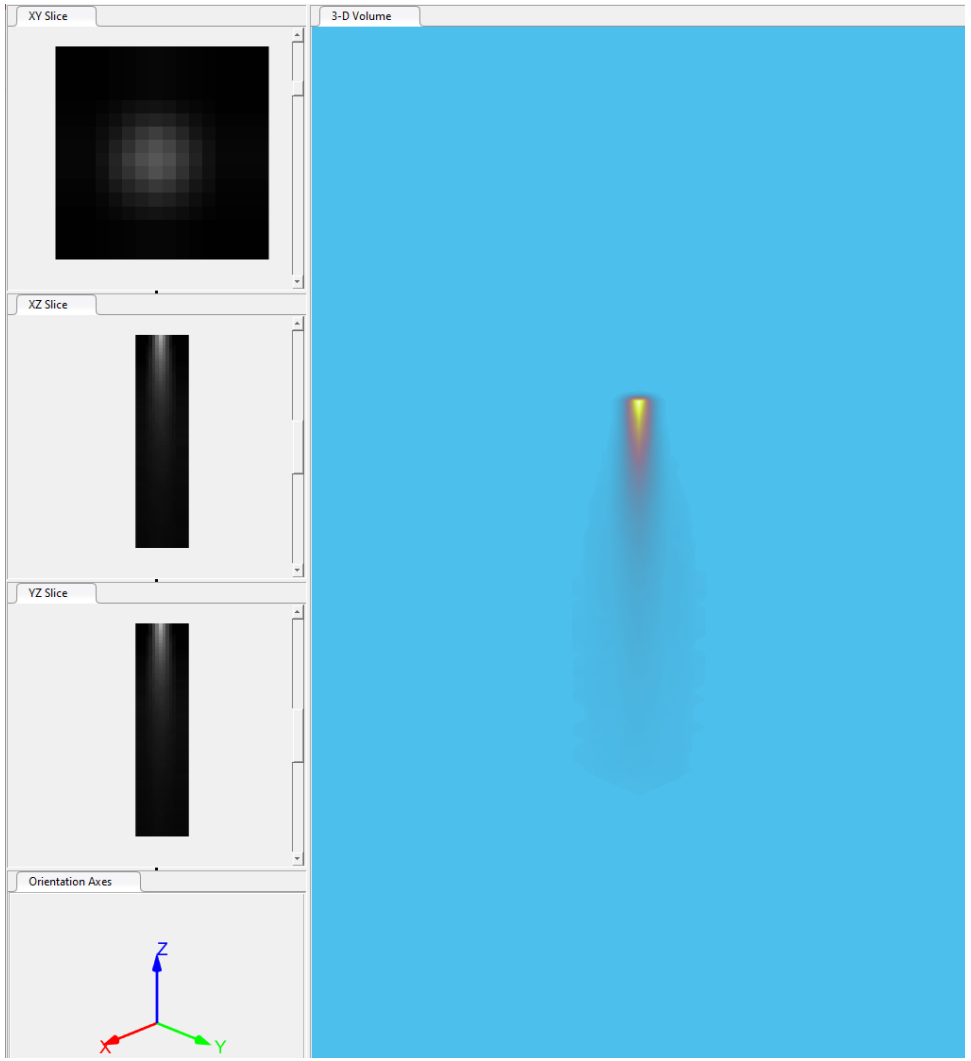
[1] Mini gamma cameras for intra-operative nuclear tomographic reconstruction. PhilippMatthies, José Gardiazabal, Aslı Okur, Jakob Vogela, Tobias Lasser, NassirNavab. https://doi.org/10.1016/j.media.2014.04.009

LUT via Gate Simulations

- **Simple Model in slide 5 (no attenuation/scattering correction);**



- **LUT via Gate Simulation;**



What's next?

- Basic code refactoring is done, converted to single precision
  - Less data copy overhead (between RAM and GPU memory), faster code
- We now have a workflow for generating LUT using Gate
  - Maybe we can verify quantitatively?
    - there is already a LUT obtained in real setup
    - we just need to re-run Gate simulations with the same parameters

High Priority;

- ➢ Do reconstructions using LUT instead of Simple-Model-Computation
- ➢ Tabulate the system matrix for drop-in probe using Gate simulations (Leiden Project)

Medium Priority;

- ➢ Implement image filters (e.g., edge-preserving smoothing)
- ➢ Try more complex system modellings; include attenuation, scattering

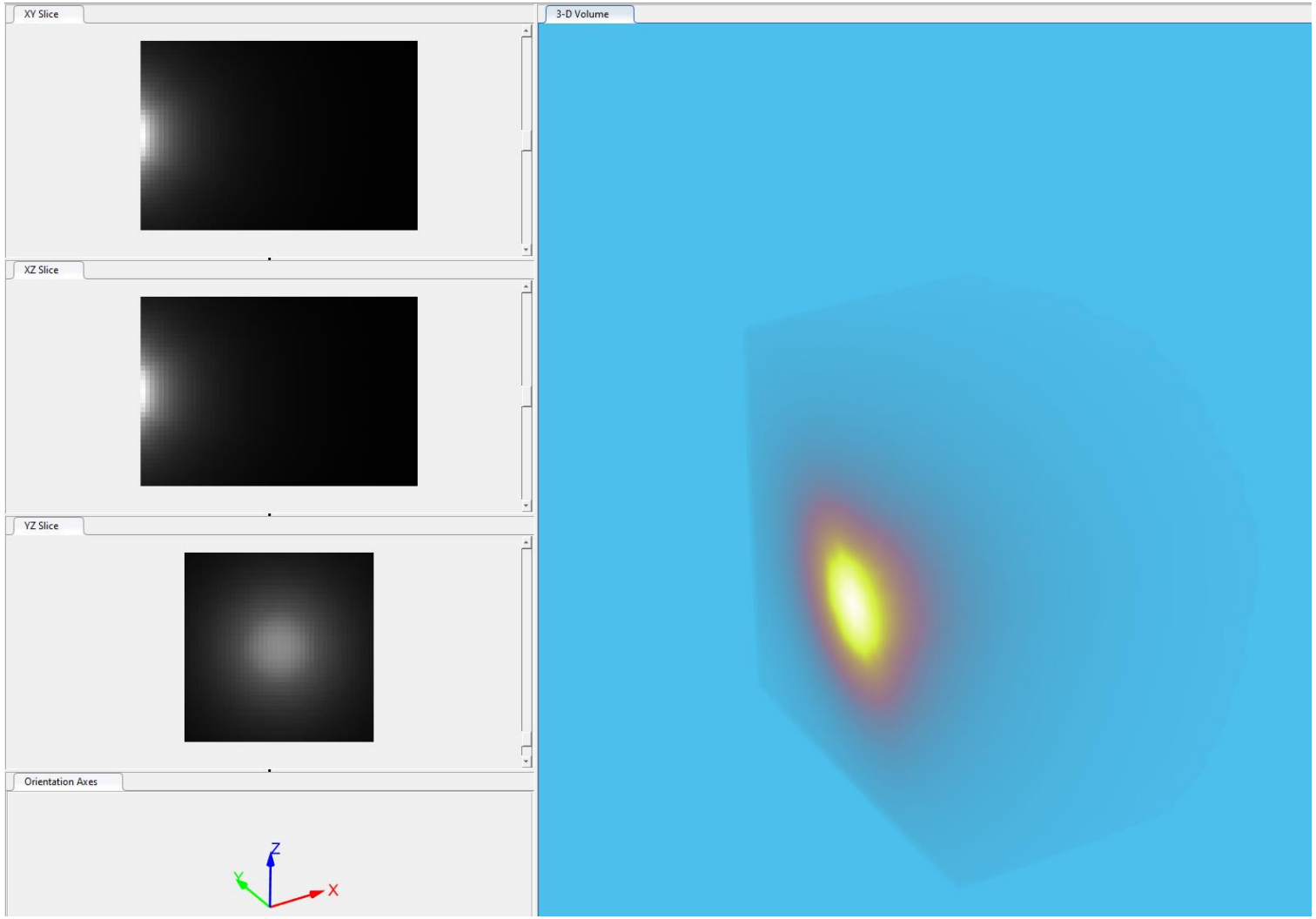
Low Priority;

- ➢ Gate simulations with the trajectory obtained from NDI tracking around neck
- ➢ Export reconstruction as .dicom file

LUT via Gate Simulations for the Drop-in Probe

➢ **Gate simulation dataset properties;**

- 2 mm resolution on x,y
  1 mm resolution on z
- Relative positions of point sources to detector;
  x, y = -20 : 2 : 20 [mm]
  z = 1 : 1 : 60 [mm]
- Number of Gate simulations = 26460 (21x21x60)
- Point activity = 0.2 MBq
- Simulation stopping criteria = 10000 counts
- Energy window = 130-150 keV for Tc99m



3D Reconstructions with LUT

➢ **How to build system matrix;**

- LUT contains count values,
  for every detector-source position pairs
- LUT data is stored in a structured grid
  (with 1 mm resolution)
- During a reconstruction;
  detector-source position pairs are arbitrary
- Need to map detector-source position pair to the LUT

➢ **First trial;**

- Perform knn search on LUT
- I realized that LUT data is not ordered,
  so MatLab was doing exhaustive search 👎

➢ **Second trial;**

- Sort LUT, by building a kd-tree out of it
- Now knn search is O(log(n)) 👍

➢ **declipse way?;**

- Floor the relative positions to integers,
  which would act like access indices
- Then search simply become O(1) 👍👍
- But the LUT data needs to be ordered first

Multi-Cam Test Scene with 4 Point Sources

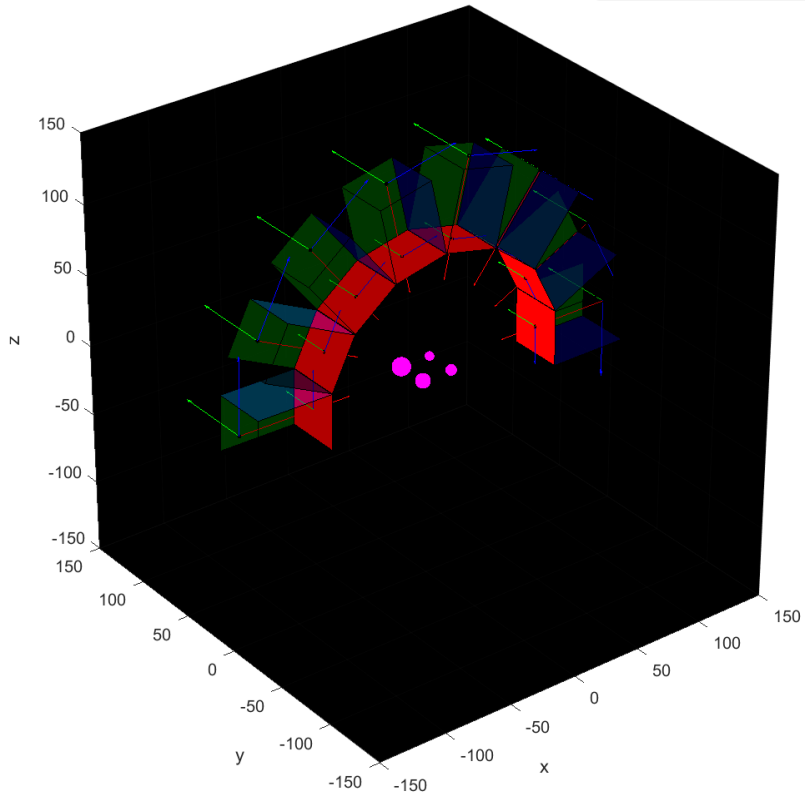
Gate Dataset Properties

➢ Single Camera

- Semi-circular motion
  (with a rotational speed of **0.5 deg/s**)
- **360 seconds** of acquisition
  (effective 360 seconds of total acquisitions)
- Period of each Gate frame is 0.25 s
  (**1440 frames** acquired frames)
- Point activities of {0.4, 0.3, 0.2, 0.1} MBq
- Collimator length 22.59 mm

➢ Double Camera

- Semi-circular motion
  (with a rotational speed of **1 deg/s**)
- **180 seconds** of acquisition **per camera**
  (effective 360 seconds of total acquisitions)
- Period of each Gate frame is 0.25 s
  (**720 frames** acquired frames **per camera**)
- Point activities of {0.4, 0.3, 0.2, 0.1} MBq
- Collimator length 22.59 mm
- Angle between cameras: 0..30 deg (every 2.5 deg)



Example Evaluation Scene

➢ Single Camera

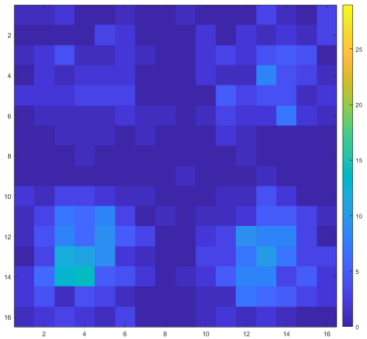
- 8 evenly distributed end effector positions
- 1 seconds of time integration for each position
  (4 frames x 8 positions = 32 frames in total)
- 32 x 0.25 = 8 seconds of effective acquisition time
  (duration that the end-user should experience)

➢ Double Camera

- 8 evenly distributed acquisition positions
- 0.5 seconds of time integration for each position
  (2 frames x 8 positions = 16 frames per camera)
- 16 x 0.25 = 4 seconds of acquisition time per camera
  (duration that the end user-should experience)

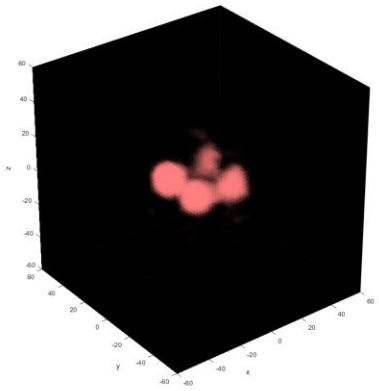
Aim is to have comparable reconstruction scenes

3D Reconstructions with LUT | Single Camera 4 point-sources

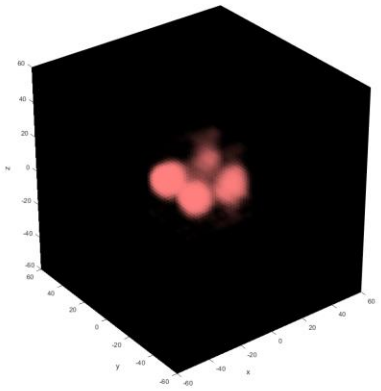


No filter

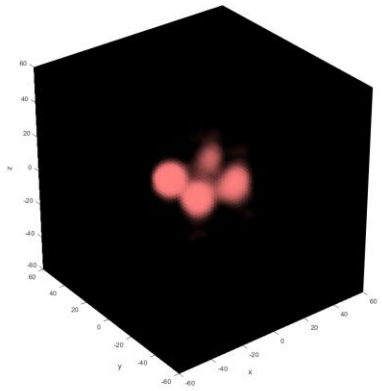
8 acq



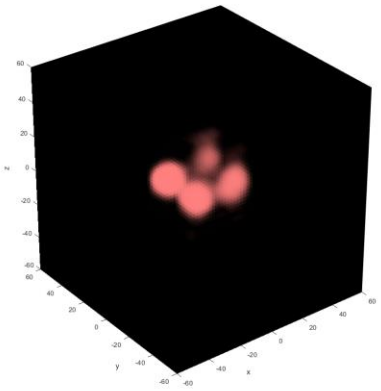
16 acq

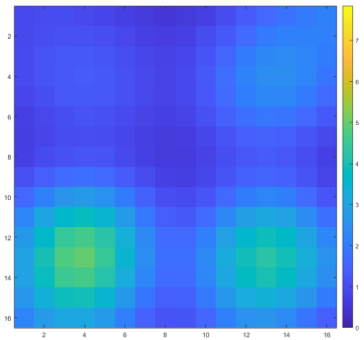


32 acq

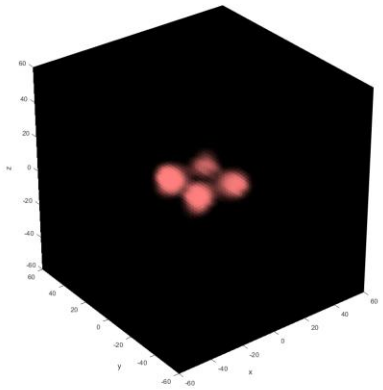


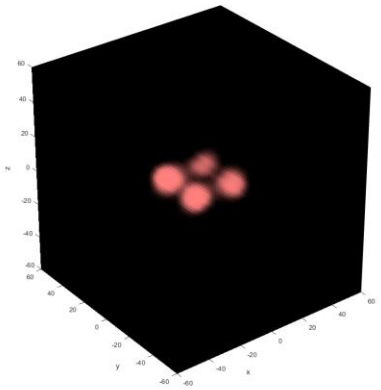
64 acq

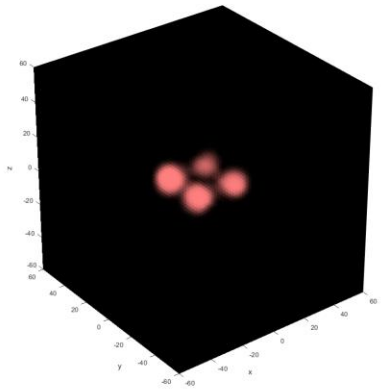


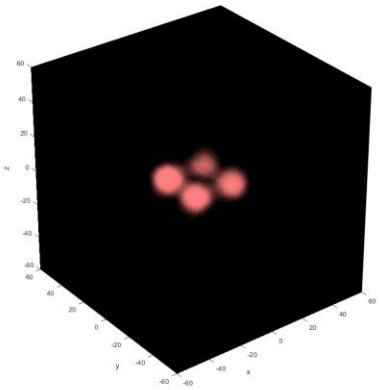


With Gaussian filter





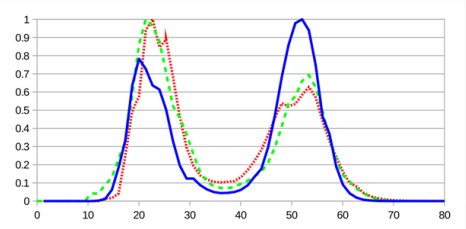




3D Reconstructions | Quantitative Evaluations

We should decide on evaluation procedure

- Separability via intensity profiles;



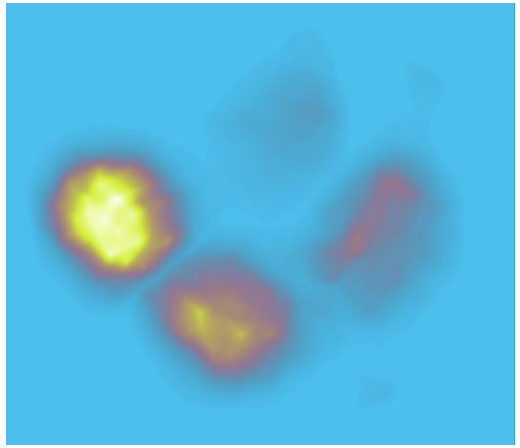
J. Gardiazabal, T. Reichl, A. Okur, T. Lasser, and N. Navab, "First flexible robotic intra-operative nuclear imaging for image-guided surgery"

- Fitting 3-dim Gaussian on each volume
  Fitting 2-dim Gaussian on slices

  --> Relate mean to source positions
  --> Relate std in each axis to asymmetricity

Eventually to evaluate the effect of;

- angle between double camera
- filtering on the acquisitions
- time integration
- trajectory around the source



What's next?

- Single camera 3D reconstruction via LUT is done
  - Need to modify the code so that we can do reconstructions with double camera
- By mean time Gate simulations are still running
  - LUT for CrystalCam with collimator length 22.59 mm
  - Double camera with different angles

High Priority;

- Decision on quantitative evaluation strategy for the reconstruction quality
- Do double camera reconstructions using LUT, and make the evaluations

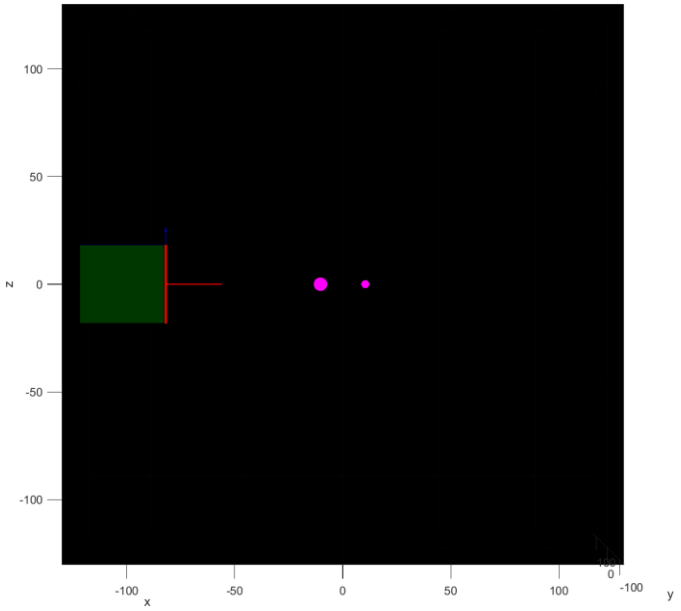
Medium Priority;

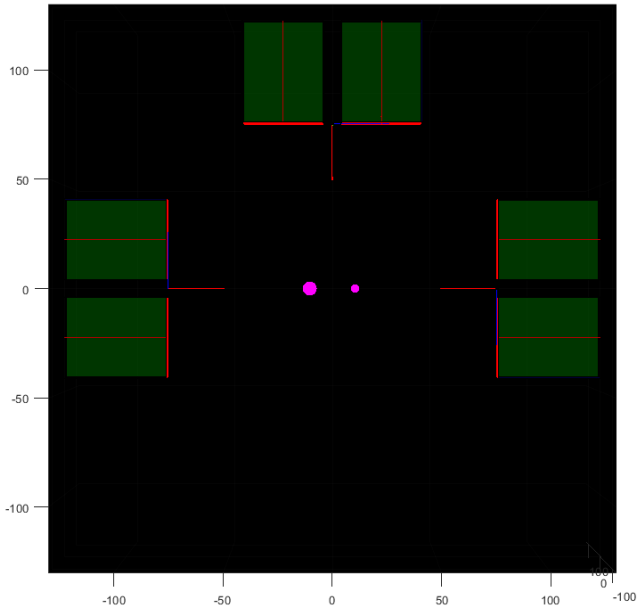
Low Priority;

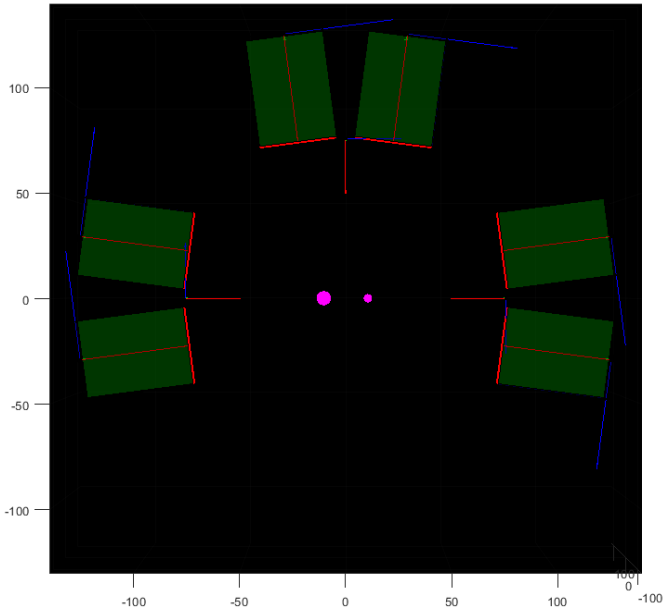
- Implement image filters (e.g., edge-preserving smoothing)
- Try more complex system modellings; include attenuation, scattering
- Gate simulations with the trajectory obtained from NDI tracking around neck
- Export reconstruction as .dicom file

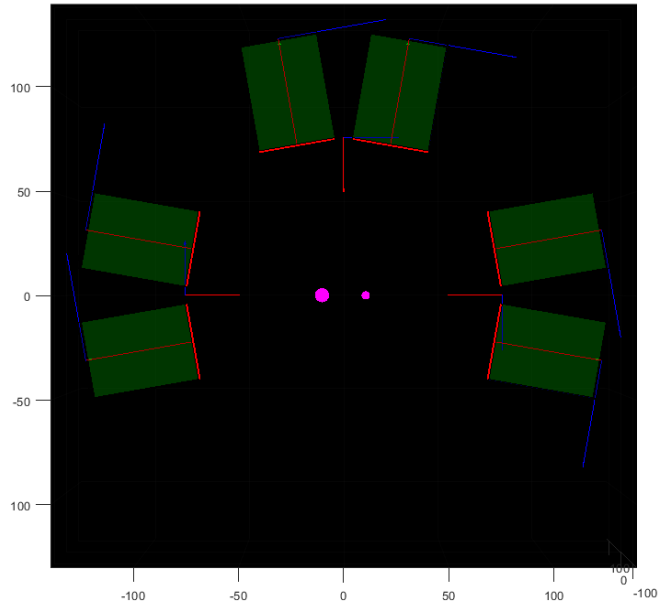
3D Reconstructions with LUT | Double Camera 4 point-sources

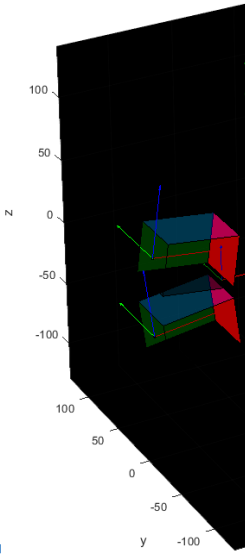
5254fca



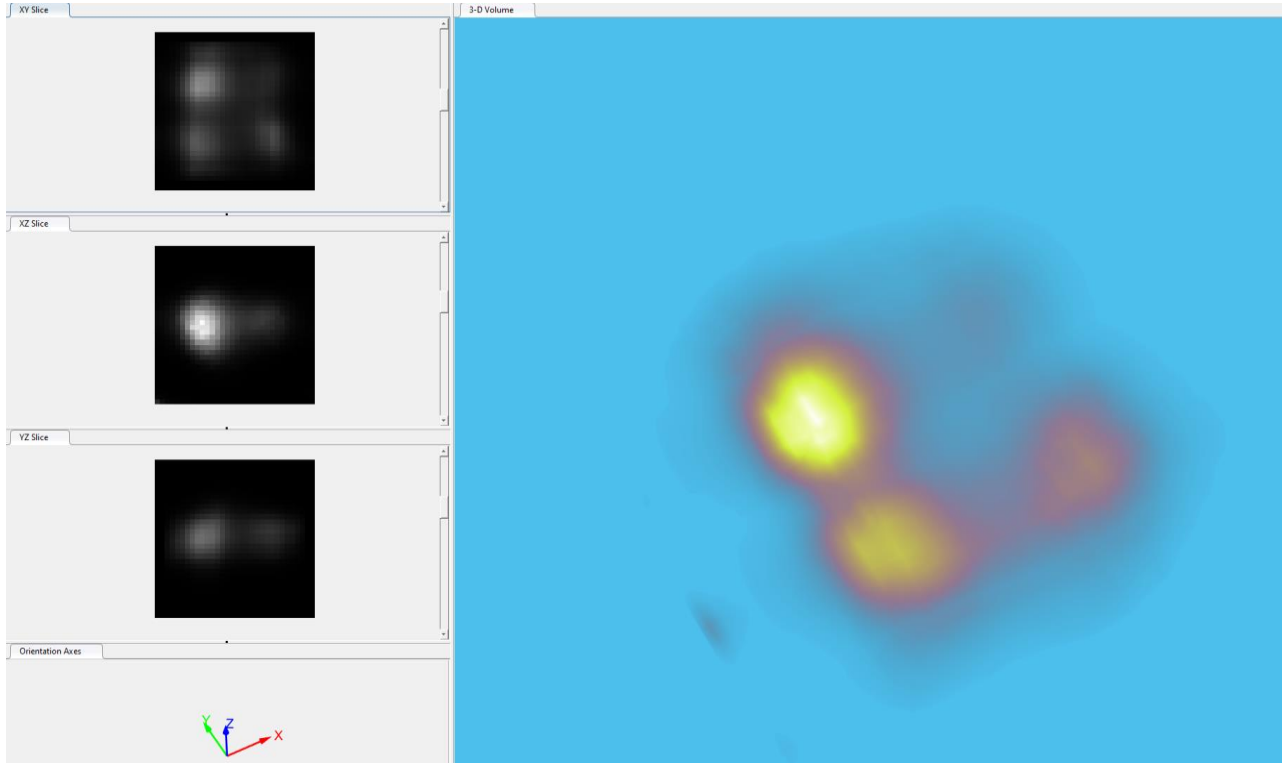








3D Reconstructions with LUT | Single Camera 4 point-sources



Just an example reconstruction scene

- 15 deg angle between double camera
- number of acquisitions = 8
  (16 CrystalCam acq.)
- Time integration = 0.5 seconds
- Simple Gaussian smoothening is applied

We should decide on evaluation strategy
then we can select best configuration

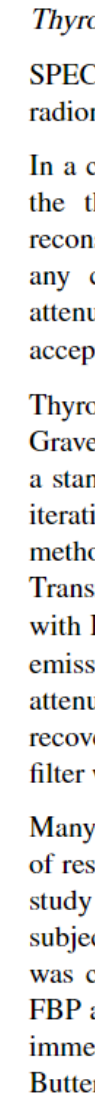
What's next?

- Double camera 3D reconstruction via LUT is done
  - Need to decide on the strategy for a quantitative evaluation
- LUT for CrystalCam with collimator length 22.59 mm is done
- By mean time Gate simulations are still running
  - Double camera with different angles (mostly done)

High Priority;
- ➢ Decision on quantitative evaluation strategy for the reconstruction quality
- ➢ Do double camera reconstructions using LUT, and make the evaluations

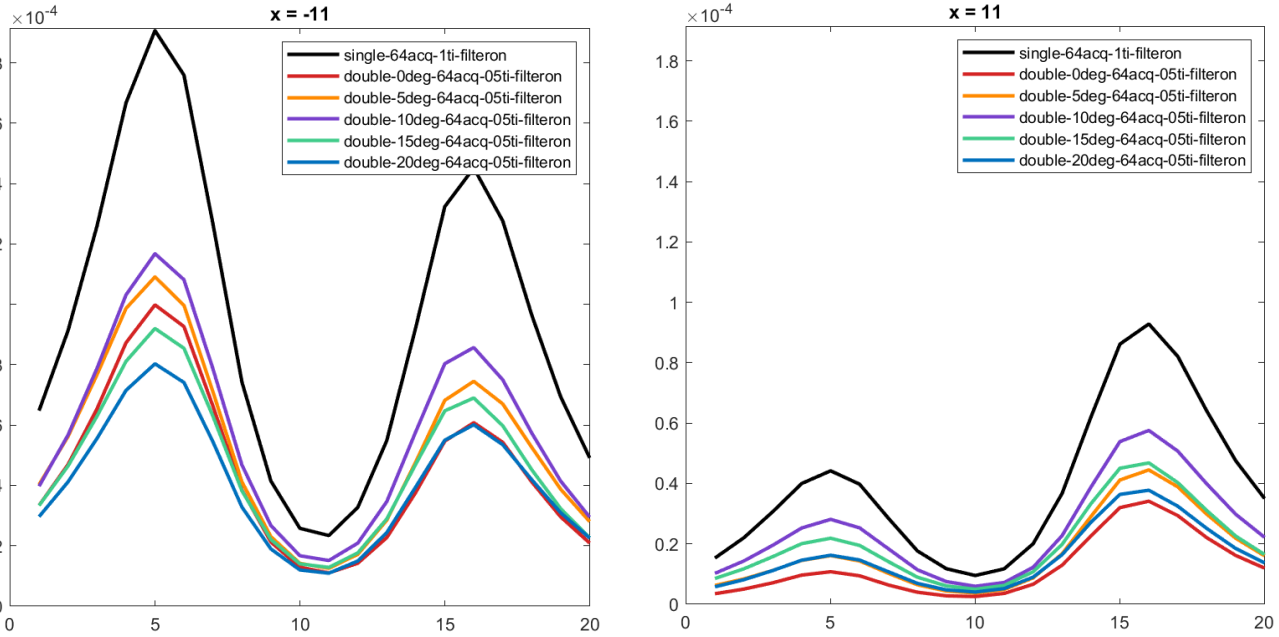
Medium Priority;
- ➢ Implement image filters (e.g., edge-preserving smoothing)

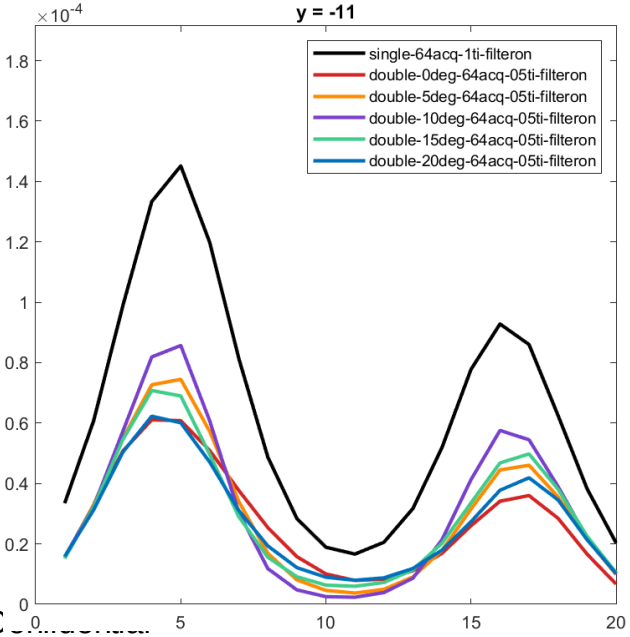
Low Priority;
- ➢ Try more complex system modellings; include attenuation, scattering
- ➢ Gate simulations with the trajectory obtained from NDI tracking around neck
- ➢ Export reconstruction as .dicom file

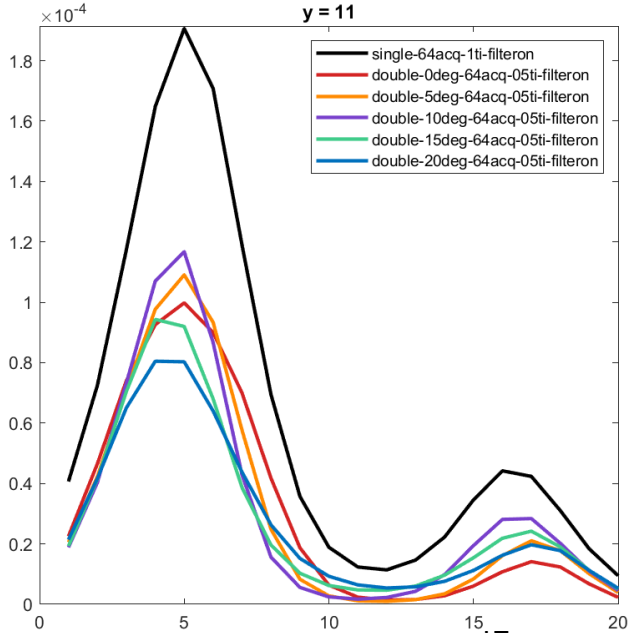

Thyro
SPEC
radio
In a c
the t
recon
any c
attenu
accep
Thyro
Grave
a stan
iterati
metho
Trans
with I
emiss
attenu
recov
filter
Many
of res
study
subje
was c
FBP a
imme
Butte

ae0b540

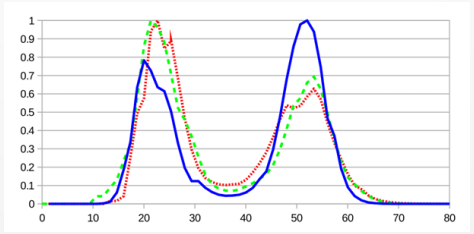
3D Reconstructions | Quantitative Evaluations



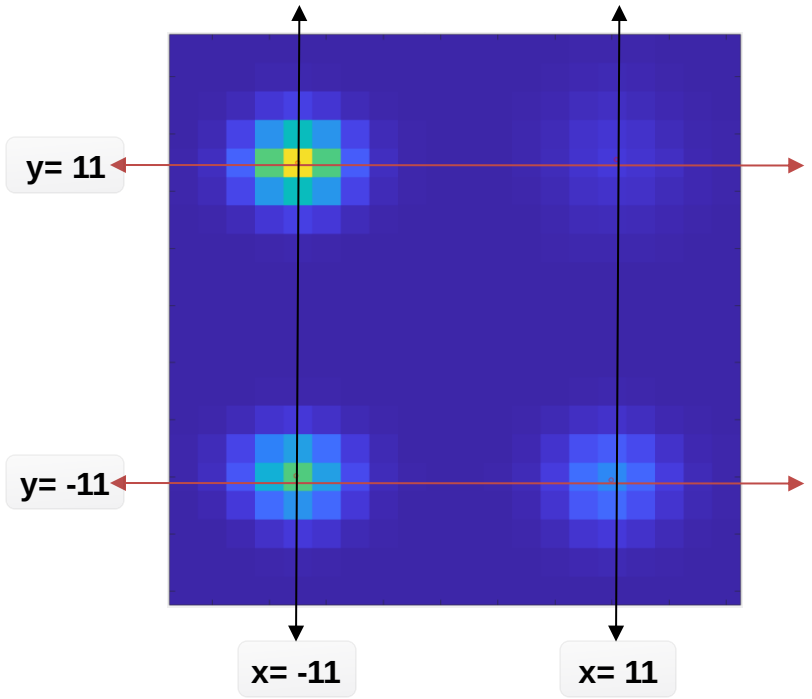




➢ **Separability via intensity profiles;**



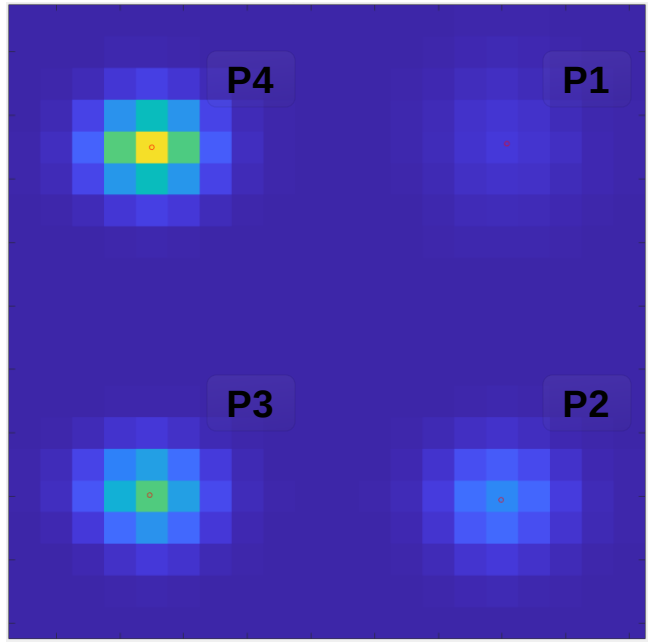
J. Gardiazabal, T. Reichl, A. Okur, T. Lasser, and N. Navab, "First flexible robotic intra-operative nuclear imaging for image-guided surgery"



3D Reconstructions | Quantitative Evaluations

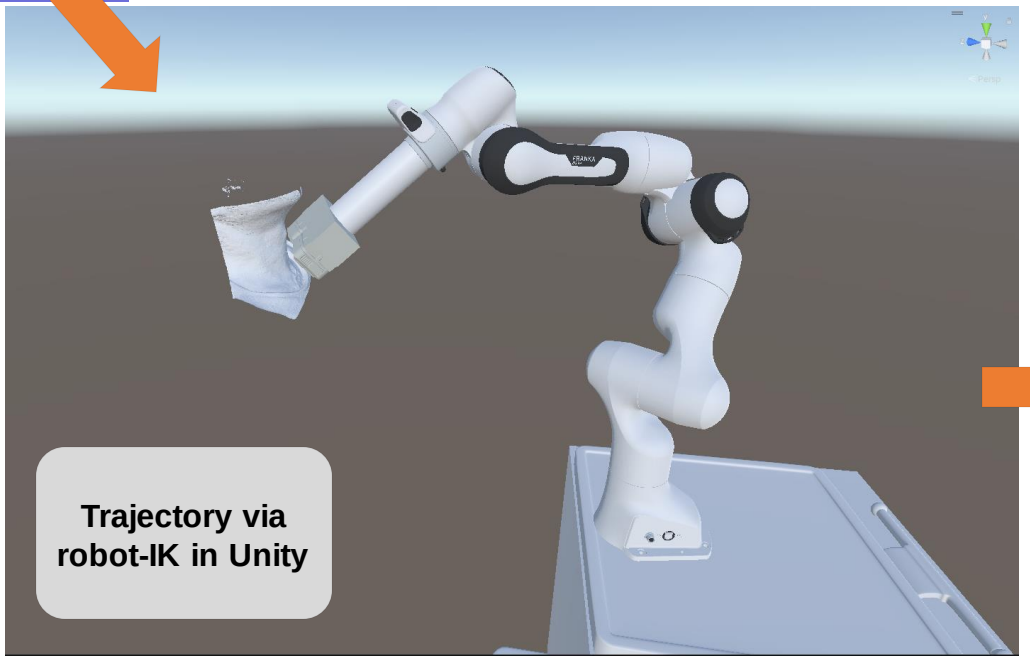
➢ **Volume centroids;**
- L2-norm between a centroid and its reference position
- [mm]

Detector Config	P1	P2	P3	P4	Mean L2-norm
single-64acq	0.55	0.41	0.15	0.10	0.30
double-0deg-64acq	0.60	0.68	0.54	0.28	0.52
double-5deg-64acq	0.57	0.22	0.16	0.33	0.32
double-10deg-64acq	0.64	0.64	0.76	0.73	0.69
double-15deg-64acq	0.24	0.40	0.47	0.59	0.43
double-20deg-64acq	0.55	0.07	0.13	0.10	0.21



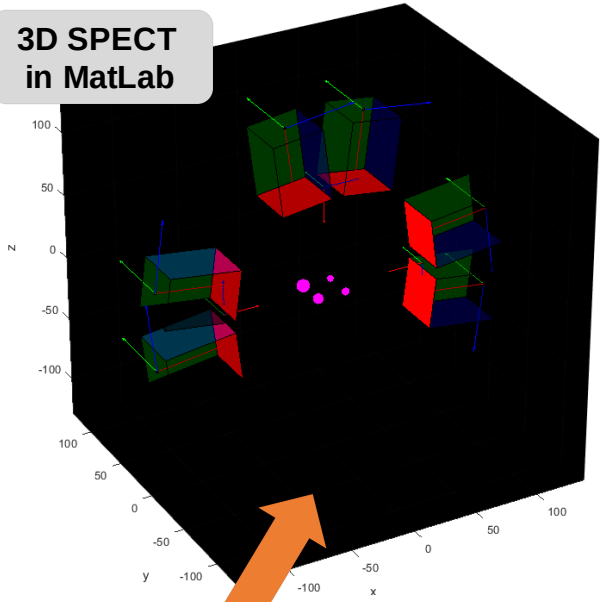
An idea for more realistic scene

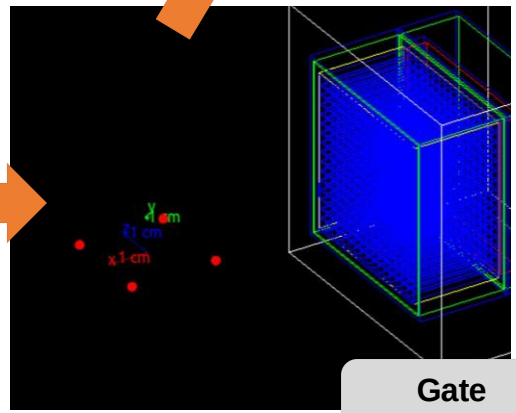
Neck reconstruction via rgbd camera



Trajectory via robot-IK in Unity

3D SPECT in MatLab





Gate simulation

What's next?

- An initial step towards quantitative evaluations is taken; intensity profiles and volume centroids
- Need to think about more realistic reconstruction scene

- Also need to improve code, maybe implementing List-Mode MLEM at this point is a good idea
  I cannot process more than 64 CrystalCam images on my laptop with the current implementation

High Priority;
- Design of a realistic scene
  - points at off-plane
- Implement List-Mode-MLEM

Medium Priority;
- Implement image filters (e.g., edge-preserving smoothing)
- Neck trajectory via robot-IK
- Uptake evaluations

Low Priority;
- Try more complex system modellings; include attenuation, scattering
- Gate simulations with the trajectory obtained from NDI tracking around neck
- Export reconstruction as .dicom file

ae0b540

Multi-Cam Test Scene with 5 Point Sources

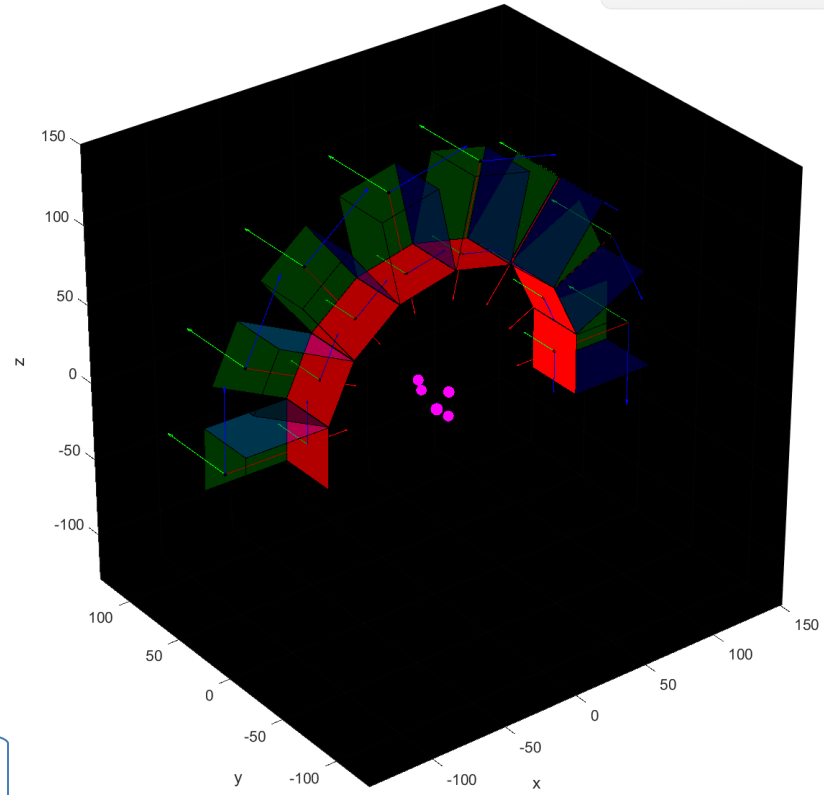
Gate Dataset Properties

➢ Single Camera

- Semi-circular motion
  (with a rotational speed of **0.5 deg/s**)
- **360 seconds** of acquisition
  (effective 360 seconds of total acquisitions)
- Period of each Gate frame is 0.25 s
  (**1440 frames** acquired frames)
- Point activities of {0.4, 0.3, 0.2, 0.1} MBq
- Collimator length 22.59 mm

➢ Double Camera

- Semi-circular motion
  (with a rotational speed of **1 deg/s**)
- **180 seconds** of acquisition **per camera**
  (effective 360 seconds of total acquisitions)
- Period of each Gate frame is 0.25 s
  (**720 frames** acquired frames **per camera**)
- Point activities of {0.4, 0.3, 0.2, 0.1} MBq
- Collimator length 22.59 mm
- Angle between cameras: 0..30 deg (every 2.5 deg)



Example Evaluation Scene

➢ Single Camera

- 64 evenly distributed end effector positions
- 1 seconds of time integration for each position
  (4 frames x 64 positions = 256 frames in total)
- 256 x 0.25 = 64 seconds of effective acquisition time
  (duration that the end-user should experience)

➢ Double Camera

- 64 evenly distributed acquisition positions
- 0.5 seconds of time integration for each position
  (2 frames x 64 positions = 128 frames per camera)
- 128 x 0.25 = 32 seconds of acquisition time per camera
  (duration that the end user-should experience)

Aim is to have comparable reconstruction scenes

List-Mode ML-EM | Data Structure

➢ Projection (Binned) - Mode Data

- On a 2D image plane
- Each pixel refers to
  - Integration of detected gamma hits
  - within a certain energy-window
  - within a certain time-window
  - unit being cps

➢ List-Mode Data

- List of raw data for individual photon events
- Each event may have the following information;
  - Time of arrival
  - Energy
  - Pose
  - Detector identifier (for multi cam systems)
  - Detector timing (for multi cam systems, dead time, detector syncing)
  - Scatter / attenuation corrections (e.g. pixels with capacitive sensors)
  - Quality metrics (such as signal-to-noise ratio, event classification, confidence level)
  - Motion tracking (patient motion tracking, respiratory signals)

So that we can do;

- ➢ Event filtering based on energy range, region of interest, patient motion etc. (reduce dimensionality)
- ➢ Event clustering based on time or position (track changes in the radioactivity, Ordered-Subset-MLEM)
- ➢ Motion correction
- ➢ Statistical analysis about noise characteristics, event statistics (image quality)

List-Mode ML-EM | Data Acquisition

➢ **Gate simulation output format**

- https://opengate.readthedocs.io/en/latest/data_output_management.html#hits-file-gatehits-dat-bin

➢ With this output format we can get the list of events with;

- Time of the event in [s]
- Energy in [MeV]
- Position of detected hit [XYZ, mm]
- Detector identifier

➢ We should be able to have wrapper software to include;

- Orientation
- Pixel id of each hit
- Attenuation / scattering (implicitly)

➢ **CrystalCam software output format**

- Each detector event with a pixel indexing is parsed
- Each event raw data is then processed into bundle of pixel-energy-time
- The unit of the energy in the pairs is arbitrary! (needs calibration per pixel to convert into keV)

➢ With the current software we should be able to store list of events with;

- Time of the event
- Detector identifier
- Detector pixel index
- Energy in arbitrary unit
- Calibration param per pixel for energy unit conversion to keV

➢ We should be able to have wrapper software to include;

- Pose
- Scatter / attenuation corrections (e.g. pixels with capacitive sensors)
- Quality metrics (such as signal-to-noise ratio, event classification, confidence level)
- Motion tracking (patient motion tracking, respiratory signals)

Plain MLEM Recap

- $\mathbf{y} = \mathbf{A}\mathbf{x}$ or $y_j = \sum_i A_{ij} x_i$
- Inverse problem: Know y_i, want x_i
- Could do $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$
- But: $\mathbf{A}$ is huge & cannot be inverted
- Solve inverse problem iteratively

- Measurements y_i: independent random variables (**Poisson**)
- Expectation value: $\mu_i = \mathbf{A}_i \cdot \mathbf{x} = \sum_j A_{ij} x_j$
- Probability to measure k given λ: $p(k|\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$

Likelihood:

$$L(\mathbf{x}) = p(\mathbf{y}|\mathbf{x}) = \prod_i p(y_i|\mathbf{x}) = \prod_i \frac{e^{-\mathbf{A}_i \cdot \mathbf{x}} (\mathbf{A}_i \cdot \mathbf{x})^{y_i}}{y_i!}$$

- Among all possible images $\mathbf{x}$ we choose the one that maximises the probability of producing the data (find most likely image)
- Maximise the log-likelihood, i.e. maximise $\log(p(\mathbf{y}|\mathbf{x}))$

Maximum Likelihood - Expectation Maximisation (ML-EM)

- Objective function to maximise: log-Likelihood (ML)
- Maximisation algorithm: Expectation Maximisation (EM)

- Initial guess for the image (uniform) $x_i^{(0)}$
- Simulate measurements from estimate (forward proj.) $y_j^{\text{simu}} = \sum_k A_{kj} x_k^{(0)}$
- Compare this with actual measurements Ratio $R_j = \frac{y_j}{y_j^{\text{simu}}}$
- Improve image estimate (backward projection) $x_i^{(1)} = x_i^{(0)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} R_j$
- Repeat until convergence

ML-EM

$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

SURGICeye®

Plain MLEM vs List-Mode-MLEM

$$x_i^{(n+1)} = x_i^{(n)} \cdot \frac{1}{\sum_j A_{ij}} \cdot \sum_j A_{ij} \frac{y_j}{\sum_k A_{kj} x_k^{(n)}}$$

- The sum is over the projection-data (for every binned detector pixel reading in cps)
- The y_j represents sum of the counts for each detector pixel

$$\lambda_j^{(k+1)} = \frac{\lambda_j^{(k)}}{\sum_{i \in \text{LORs}} A_{ij}} \sum_{m \in \text{event-list}} A_{i_m j} \frac{1}{\sum_j A_{i_m j} \lambda_j^{(k)}}$$

[1] [2] [3]

- The sum is now only over the list of available events (each event is a single count, not pixel-binned cps)
- Yet, the normalization term has the sum over all possible LORs ("all possible LORS" means all the detector pixels even if some pixels have no hits acquired?)
- In the ratio term, instead of (pixel binned cps) / (forw. Proj.), now (1) / (forw. proj.)

What's next?

- Gate simulations with points sources off the plane are done
  - Need to make the evaluations of different camera configurations
- Implementation of List-Mode MLEM is started, but not finished yet
- Data acquisition from Gate in List-Mode format can be doable,
  - Static scene with a single point source is done, next step is to get a simple simulation with moving camera

High Priority;
- Evaluations of camera configurations with the 5points-scene
- Implement List-Mode-MLEM

Medium Priority;
- Neck trajectory via robot-IK
- Uptake evaluations
- List-Mode data acquisition from a simple Gate scene with a moving camera

Low Priority;
- Try more complex system modellings; include attenuation, scattering
- Implement image filters (e.g., edge-preserving smoothing)
- Gate simulations with the trajectory obtained from NDI tracking around neck
- Export reconstruction as .dicom file

we implement
interventional guidance through imaging

we define
new standards in tumor treatment

we care
about the **cure** of cancer

SURGICeye®

SurgicEye GmbH
Munich, Germany

www.surgiceye.com

info@surgiceye.com
+49 89 54 99 890 01