Invers Radon Transform?

Radon Transform

$$Rf(L) = \int_L f(x)|dx|$$

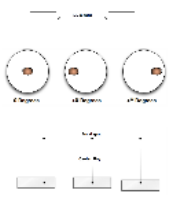
How hard can it be...

CT SCANNING

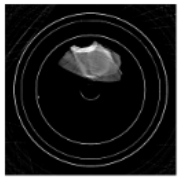
Industrial CT
No time for pretty pictures....

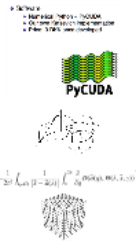
238 GFLOP for a reconstruction

X-Ray Tube



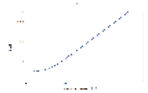
Detector panel



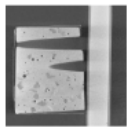


PyCUDA

Toolbox



More of the same...



Invers Radon Transform?

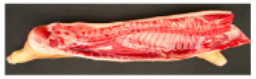
Radon Transform

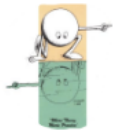
$$Rf(L) = \int_L f(x)|dx|$$

How hard can it be...

CT SCANNING

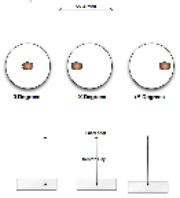
Industrial CT
No time for pretty pictures....



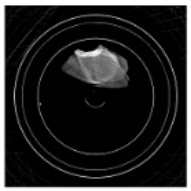


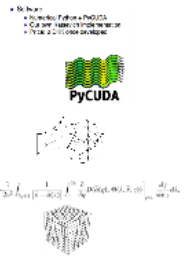
338 GFLOP for a reconstruction

X-Ray Tube



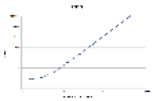
Detector panel



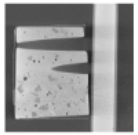


PyCUDA

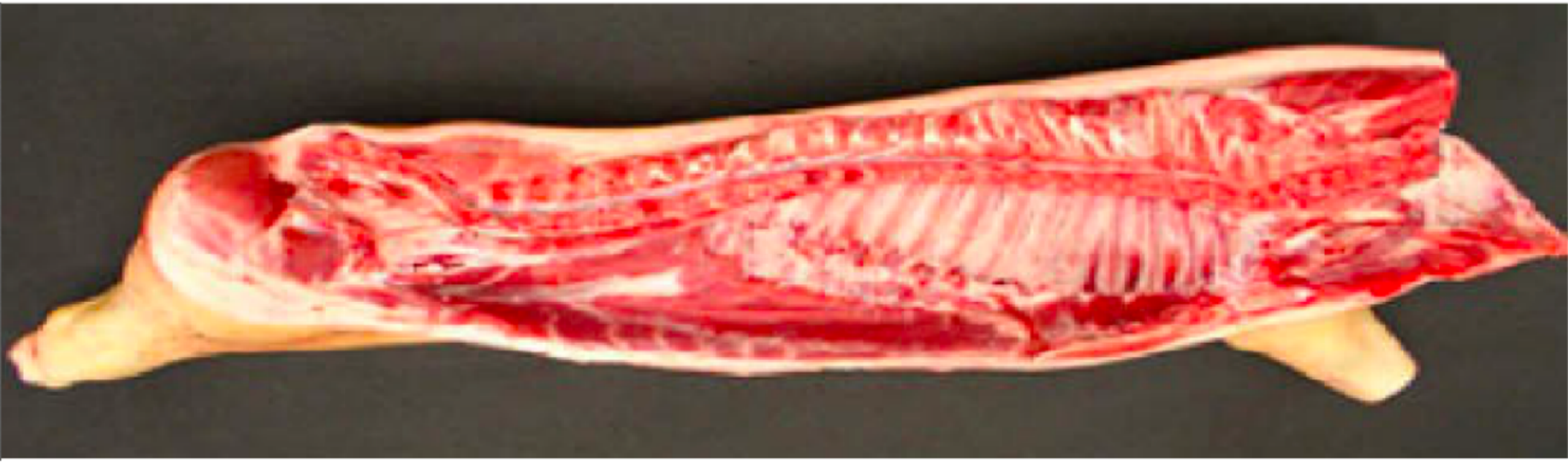
Toolbox



More of the same...

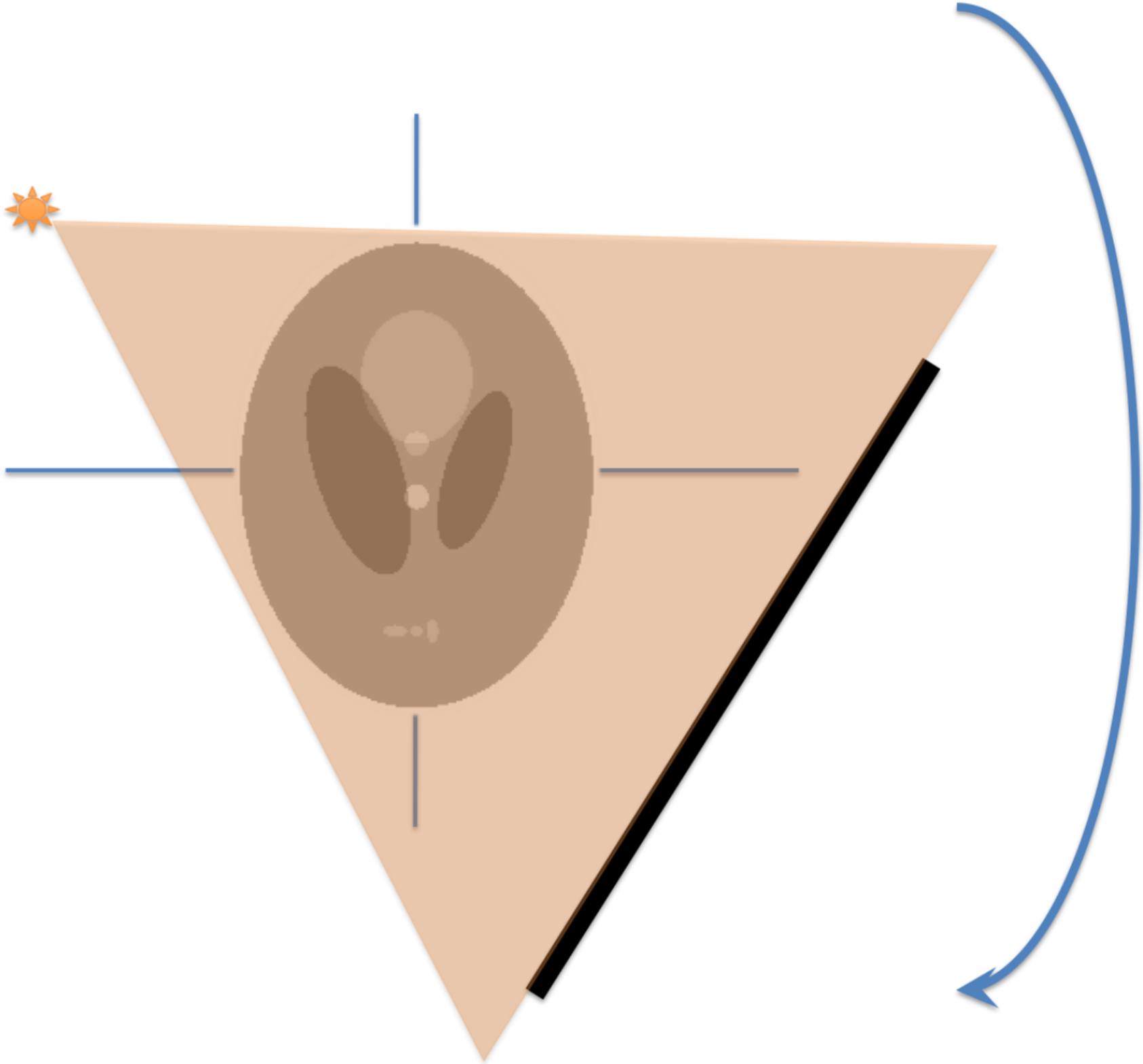


Industrial CT
No time for pretty pictures….



How hard can it be...

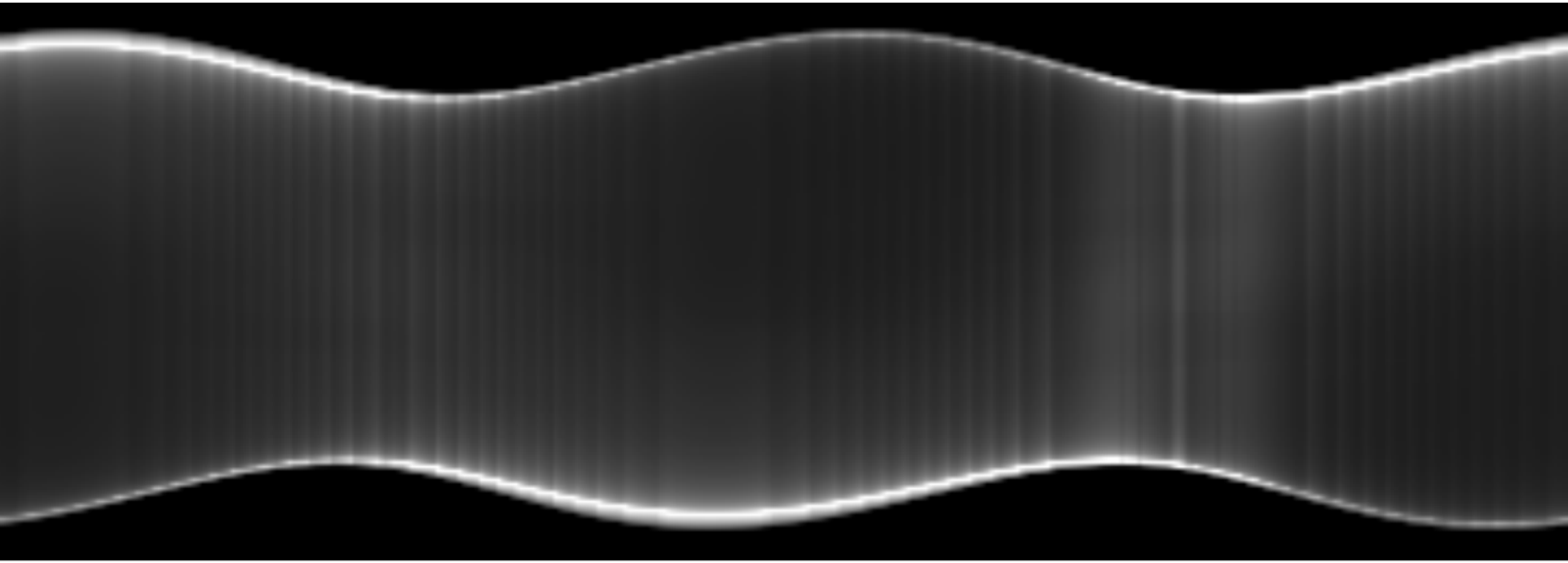
CT SCANNING

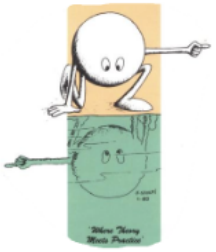


Radon Transform

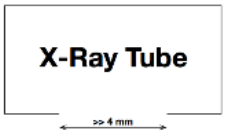
$$Rf(L) = \int_L f(x)\left|dx\right|$$

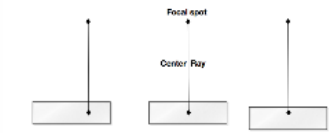
Invers Radon Transform?

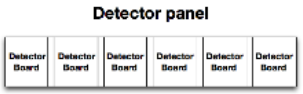




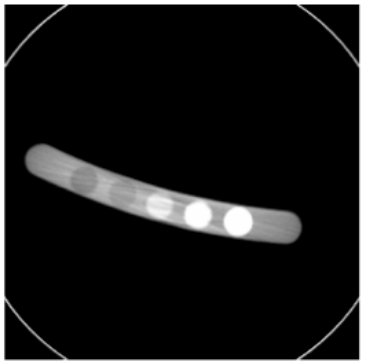
238 GFLOP for a reconstruction

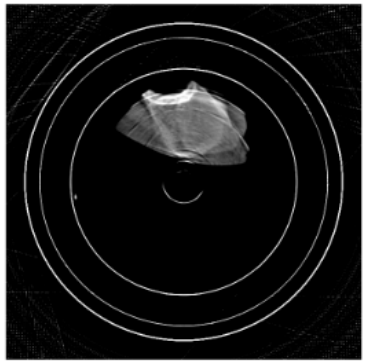


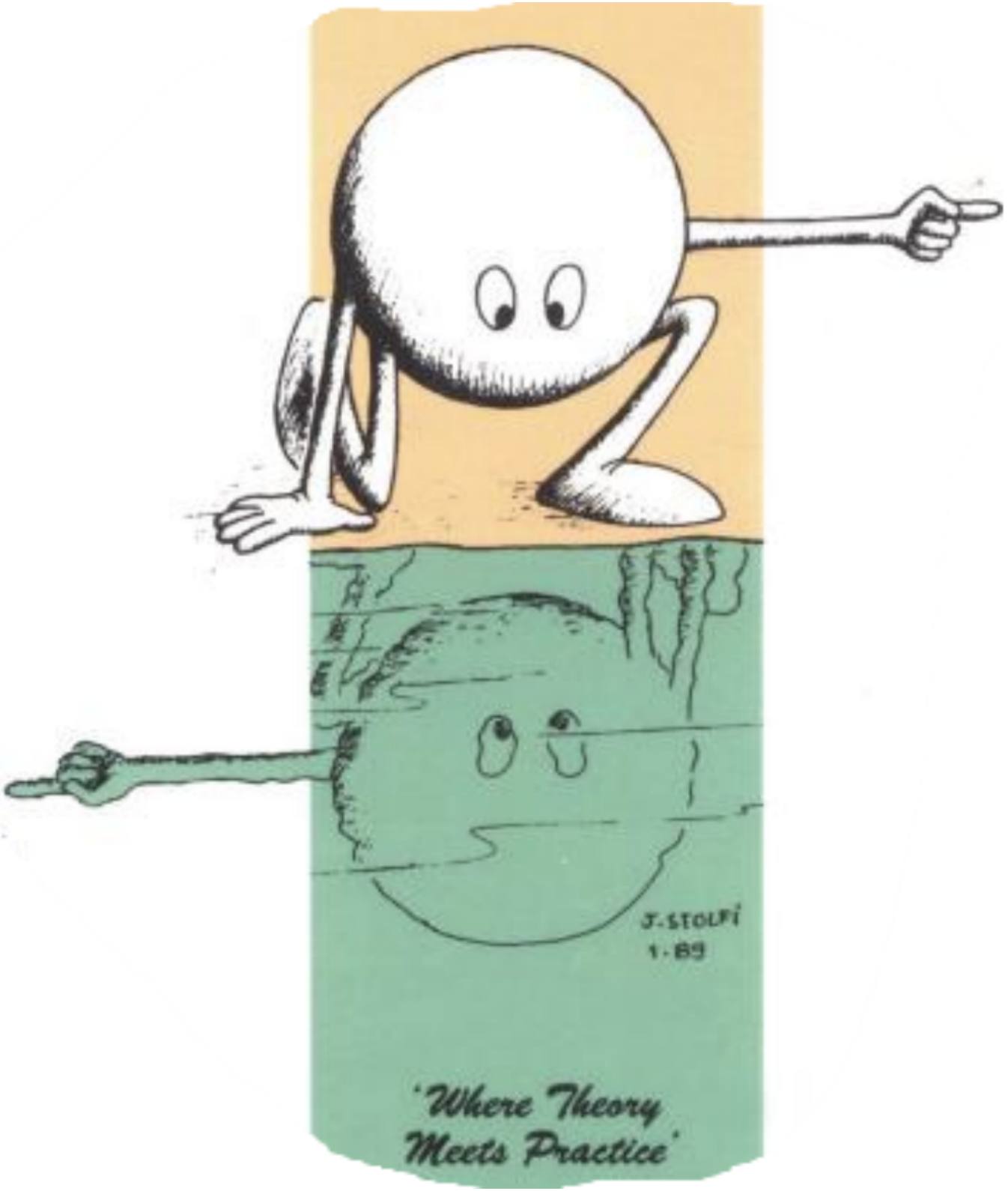


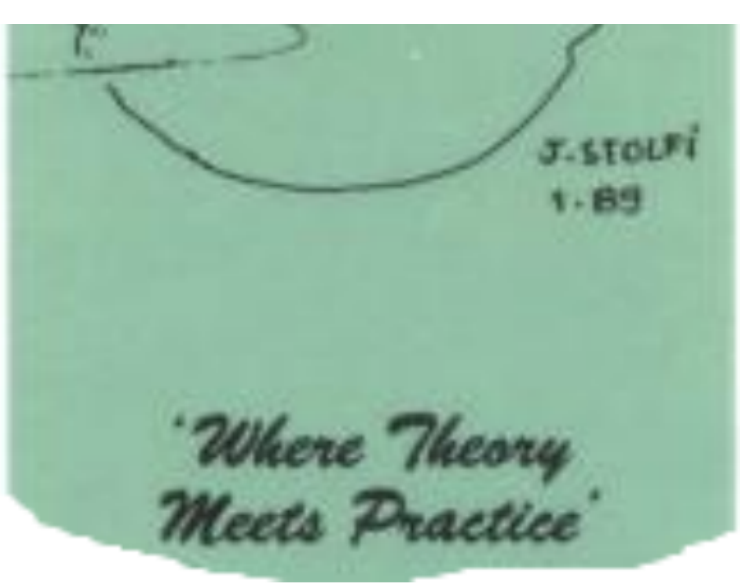


- Rotation angle alignment
- Detector sample rate
- Detector dynamic range
- Detector scintillator



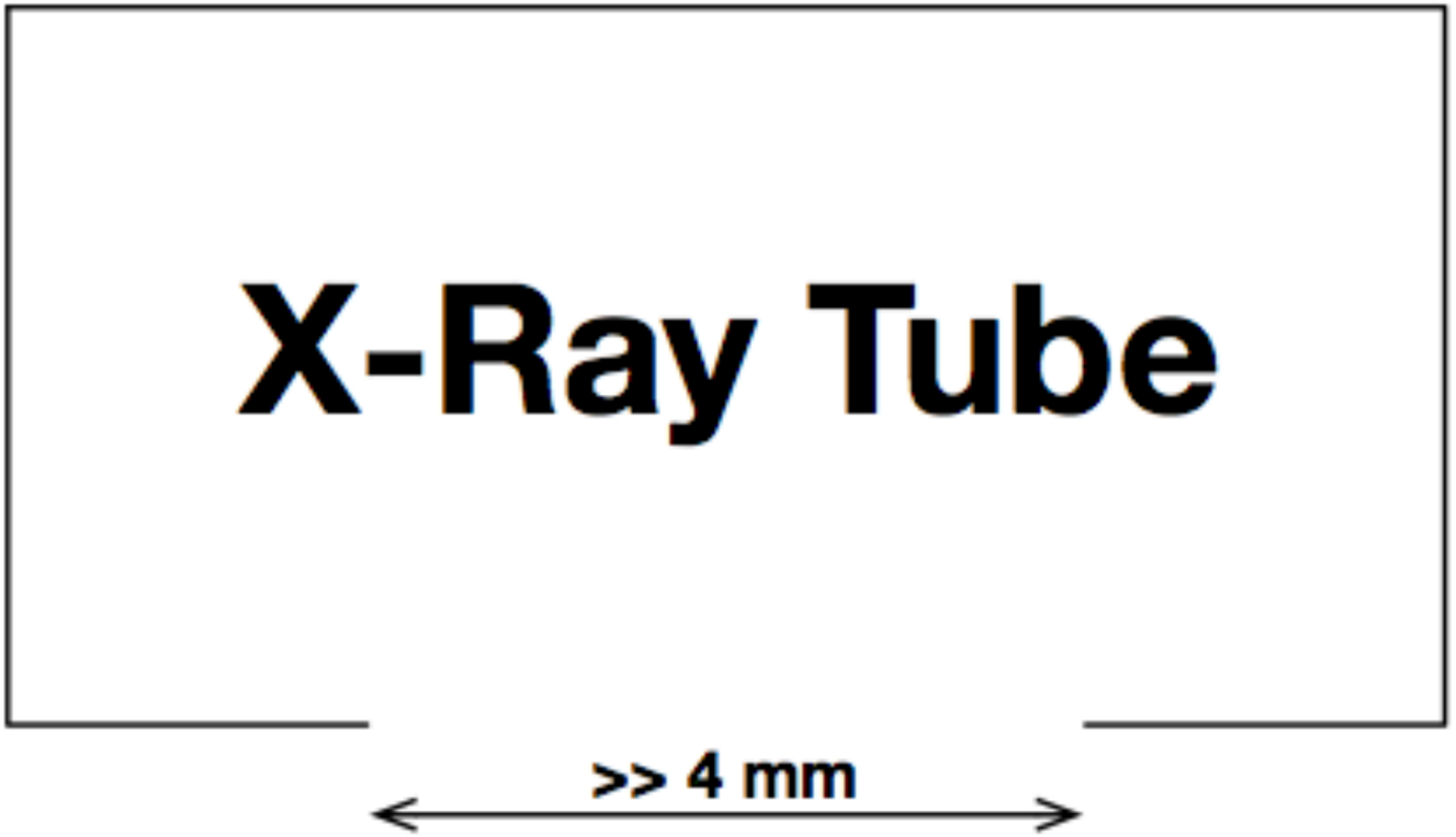


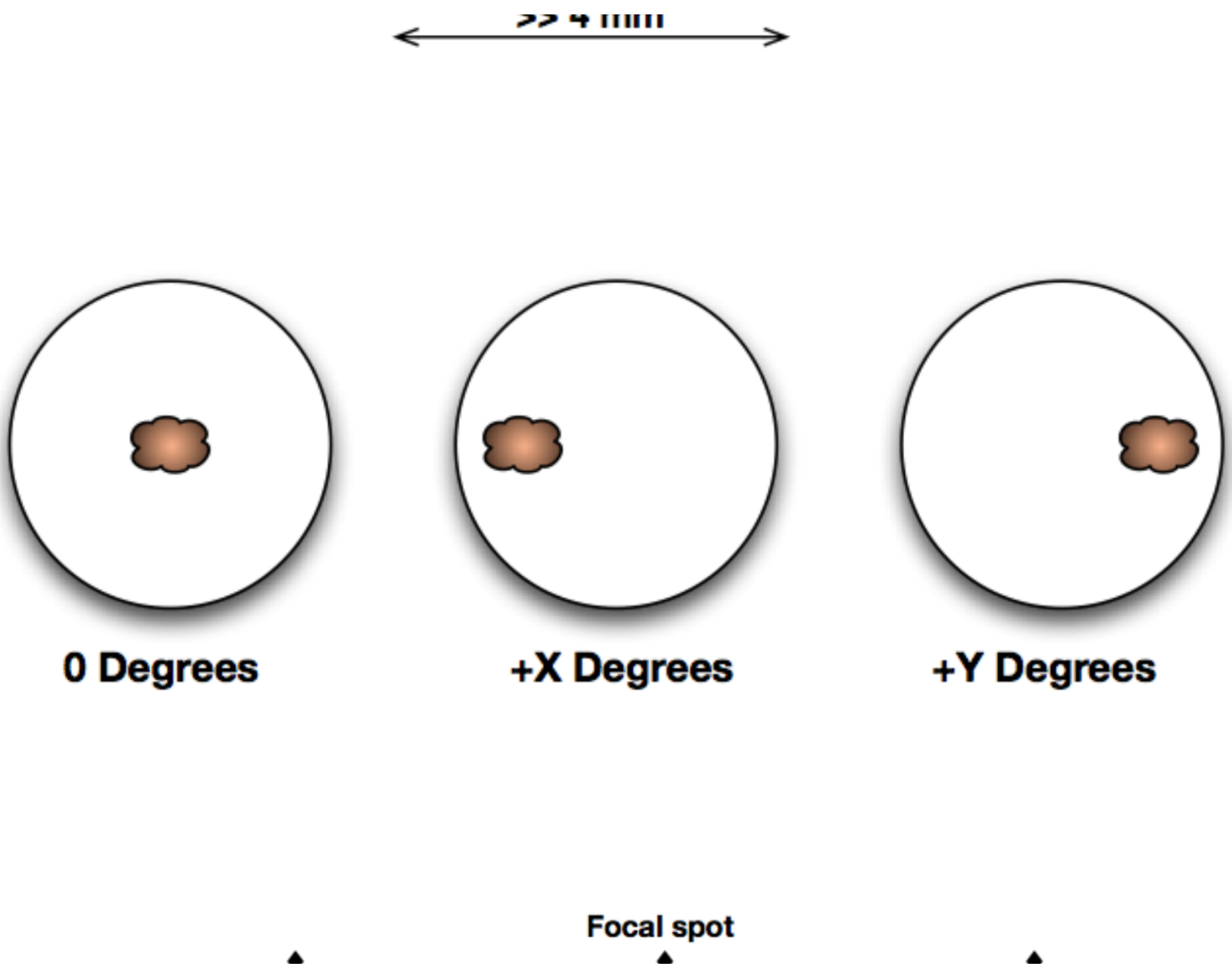

J.STOUFI
1-89
'Where Theory
Meets Practice'



238 GFLOP for a reconstruction

X-Ray Tube

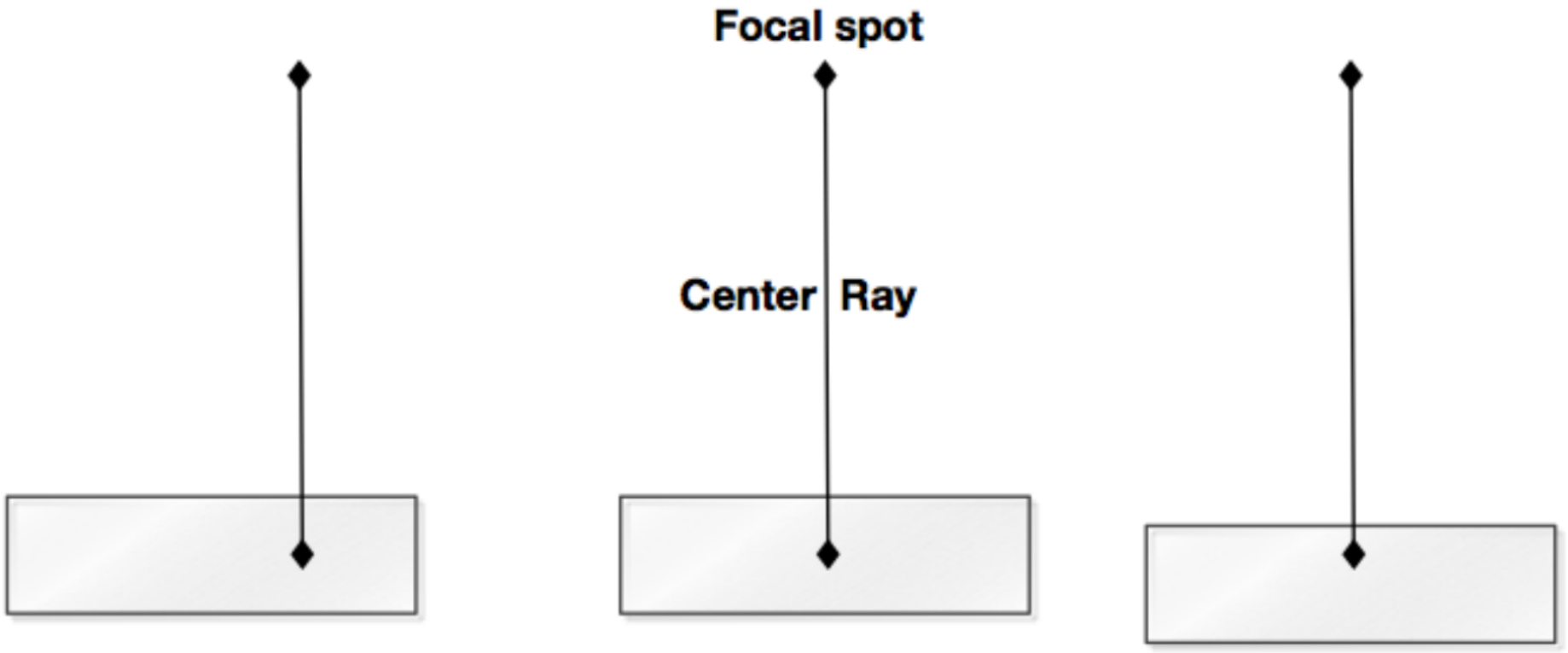

X-Ray Tube
>> 4 mm


0 Degrees
+X Degrees
+Y Degrees
Focal spot

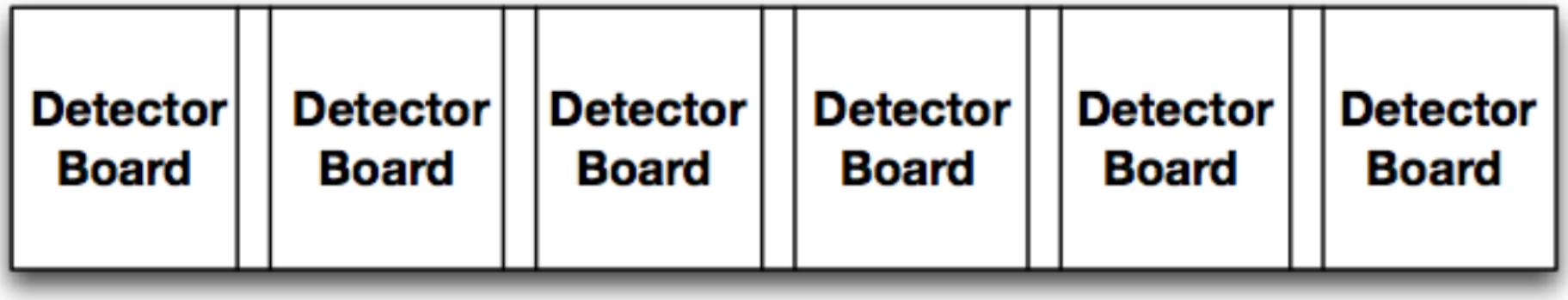
0 Degrees

+X Degrees

+Y Degrees



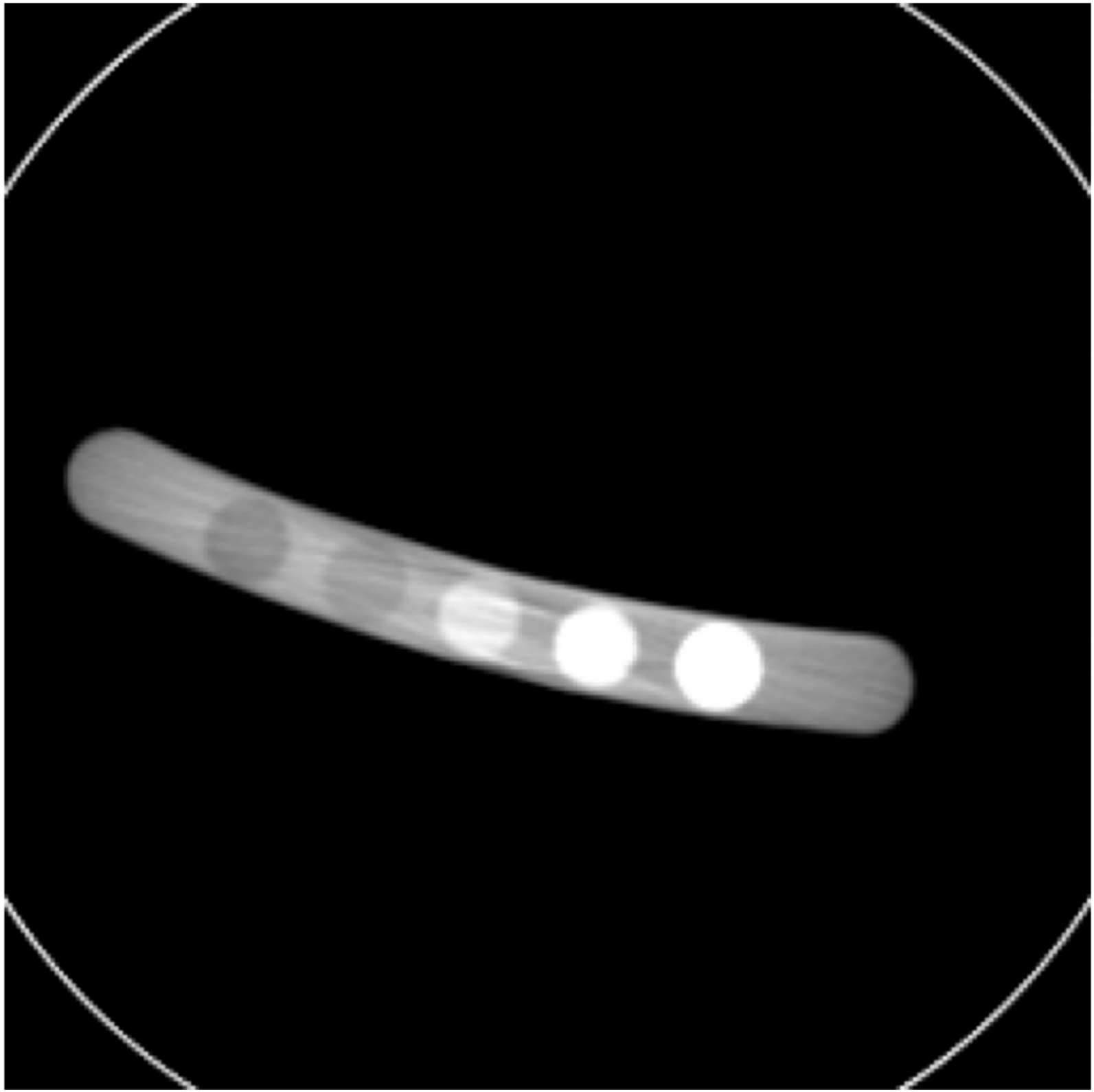
Detector panel

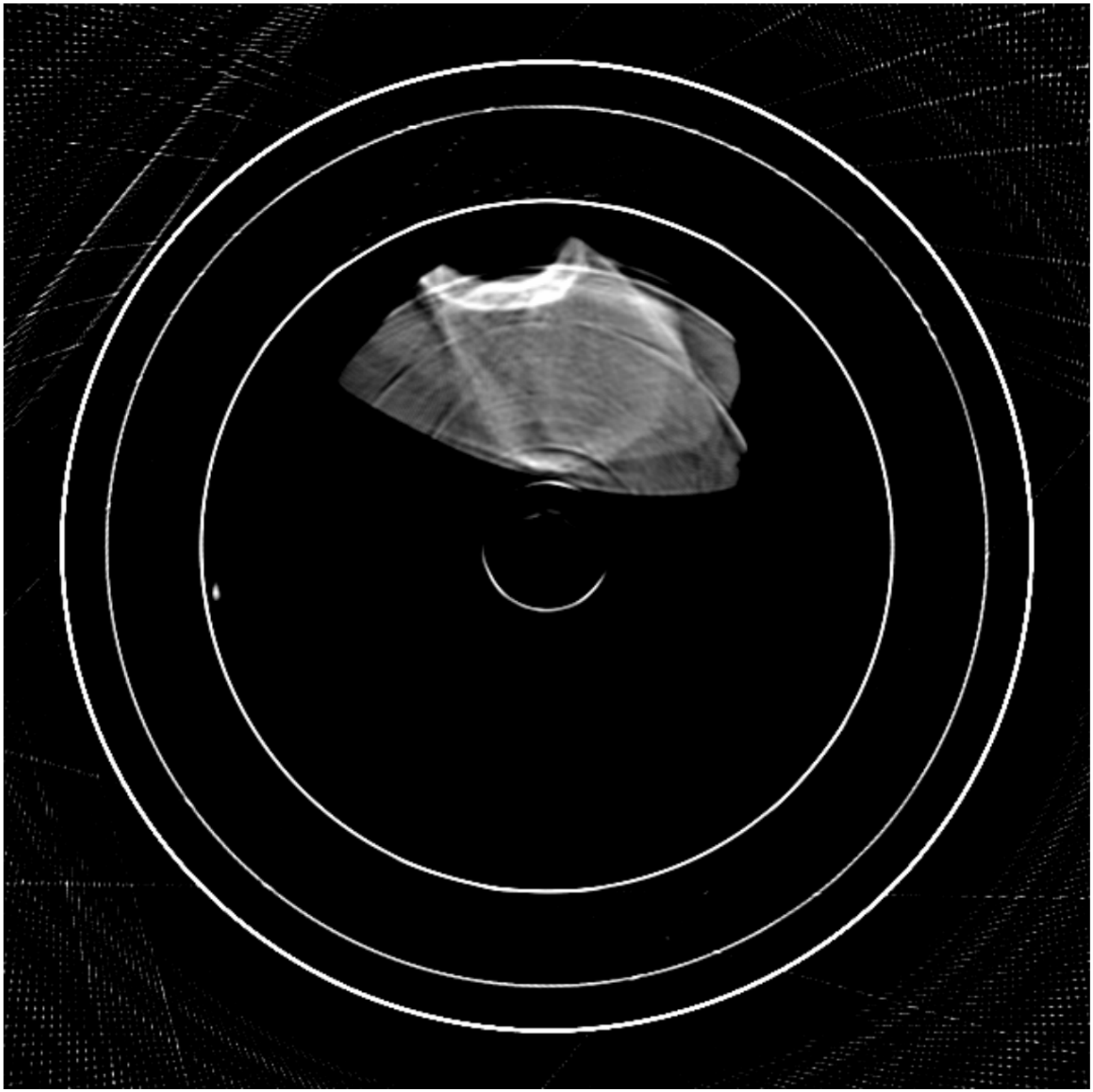


- Rotation angle alignment

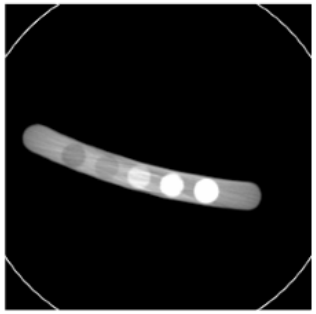
- Rotation angle alignment
- Detector sample rate
- Detector dynamic range
- Detector scintillator


12
1
2
3
4
5
6
7
8
9
10
11





More of the same



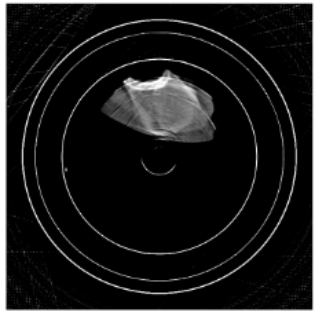
- Software
  - Numerical Python + PyCUDA
  - Our own Katsevich implementation
  - Price: 0 DKK once developed

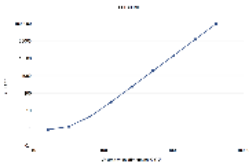
PyCUDA

$$f(\vec{x}) = -\frac{1}{2\pi^2}\int_{I_{PI}(\vec{x})}\frac{1}{|\vec{x}-\vec{a}(\lambda)|}\left[\int_0^{2\pi}\frac{\partial}{\partial q}D(\vec{a}(q),\Theta(\lambda,\vec{x},\gamma))\right]_{q=\lambda}\frac{d\gamma}{\sin\gamma}\,d\lambda,$$

Toolbox

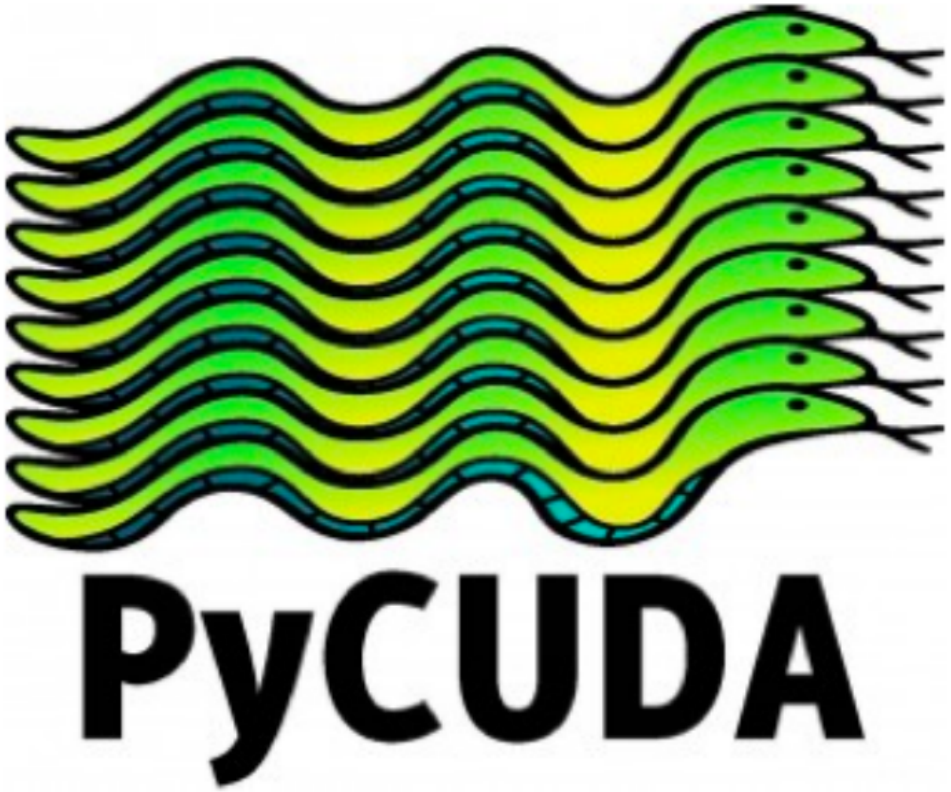
- CT reconstruction software toolbox
  - Fan beam Setup and Shoot (flat/curved)
  - Cone beam Setup and Shoot (flat/curved)
  - Cone beam helix (flat/curved)
- Including both educational and highly optimized code
- User manuals for CT reconstruction
- CT Reconstruction examples
- To be released this fall under GPL V2 license

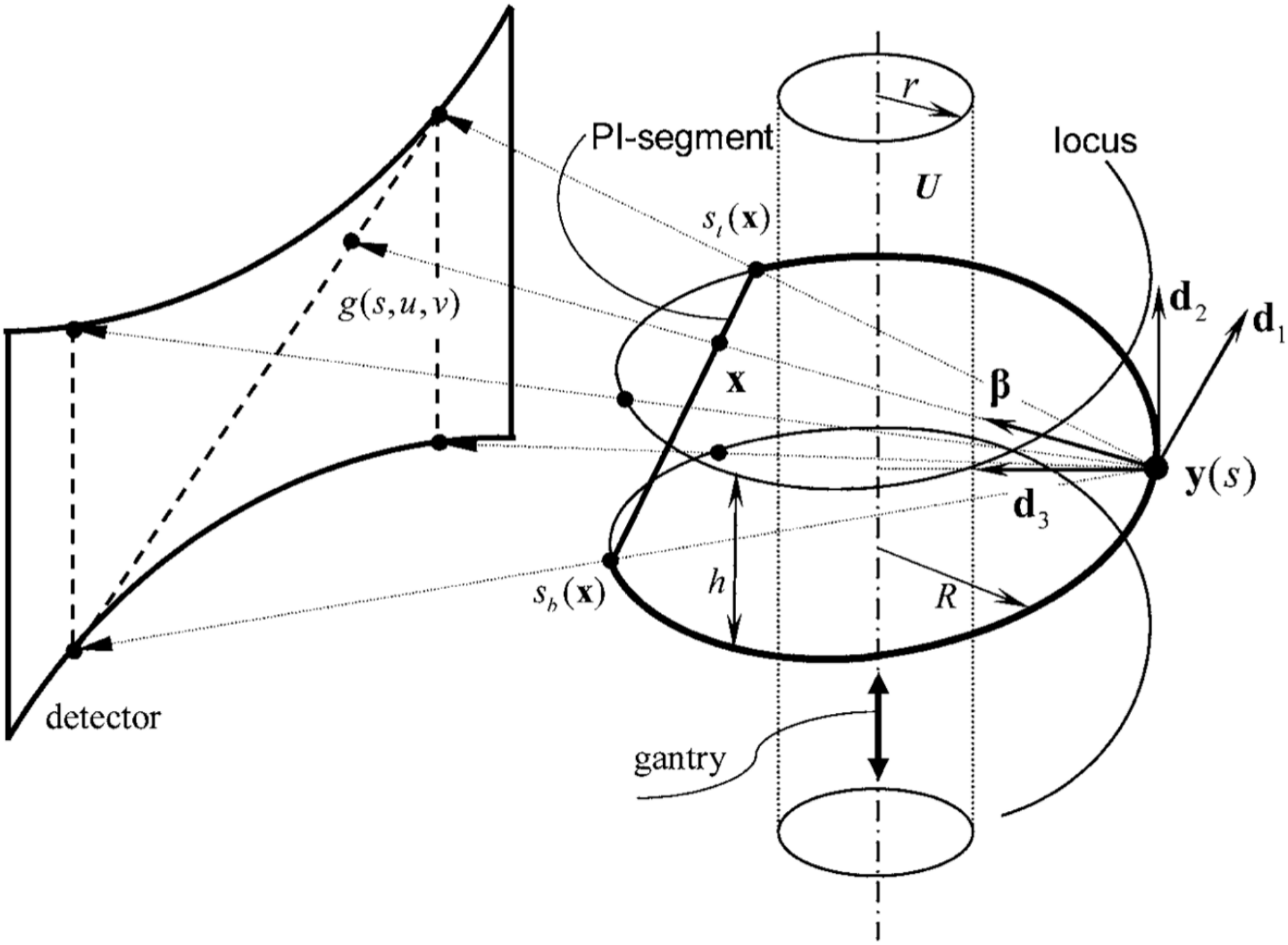


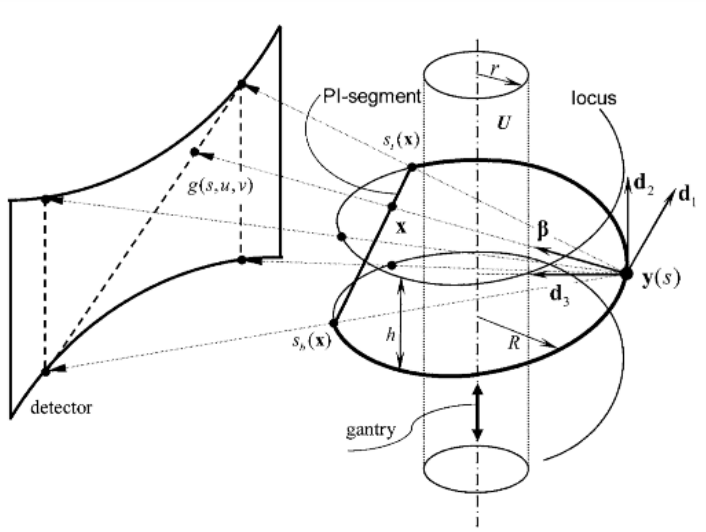



GTX 46
NVIDIA

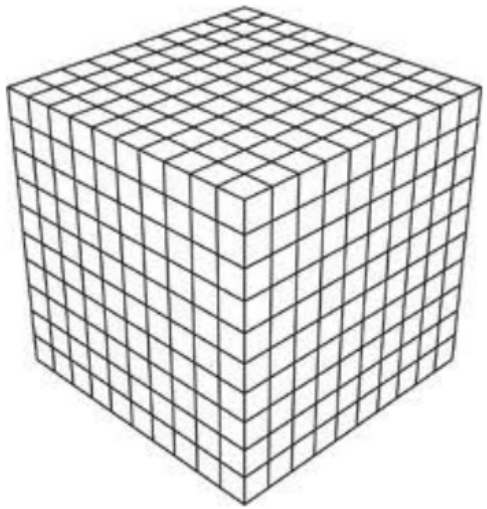
- Software
  - Numerical Python + PyCUDA
  - Our own Katsevich implementation
  - Price: 0 DKK once developed

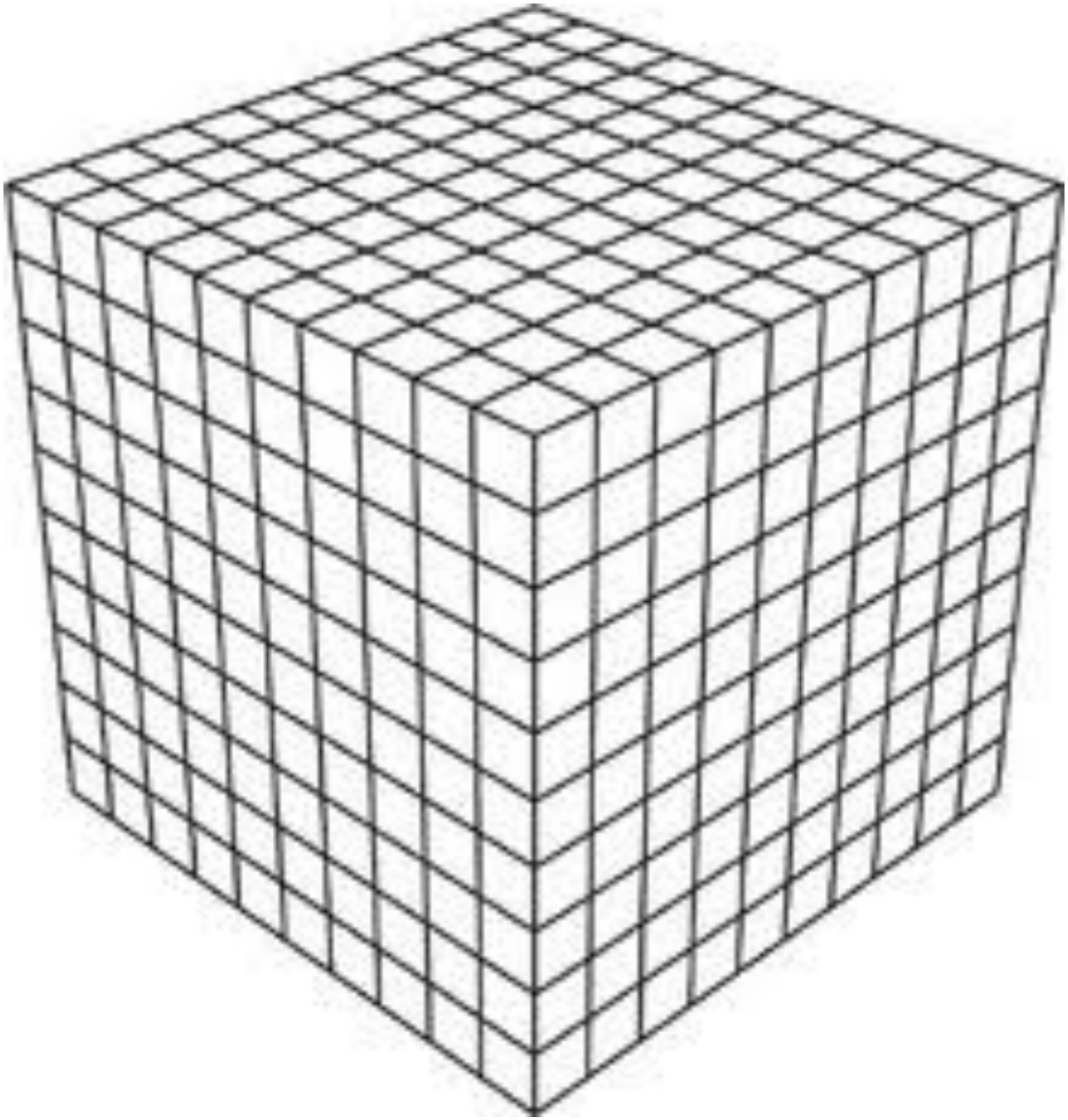


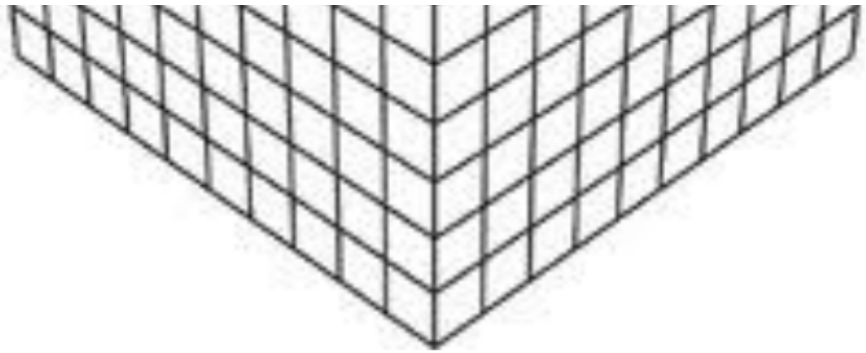

r
PI-segment
locus
U
$s_t(\mathbf{x})$
$g(s,u,v)$
$\mathbf{d}_2$
$\mathbf{d}_1$
$\mathbf{x}$
β
$\mathbf{y}(s)$
$\mathbf{d}_3$
h
R
$s_b(\mathbf{x})$
detector
gantry



$$f(\vec{x}) = -\frac{1}{2\pi^2} \int_{I_{PI}(\vec{x})} \frac{1}{\left|\vec{x} - \vec{a}(\lambda)\right|} \int_0^{2\pi} \frac{\partial}{\partial q} D(\vec{a}(q), \Theta(\lambda, \vec{x}, \gamma))\Big|_{q=\lambda} \frac{d\gamma}{\sin\gamma} d\lambda,$$







Toolbox

- CT reconstruction software toolbox
  - Fan beam Step and Shoot (flat/curved)

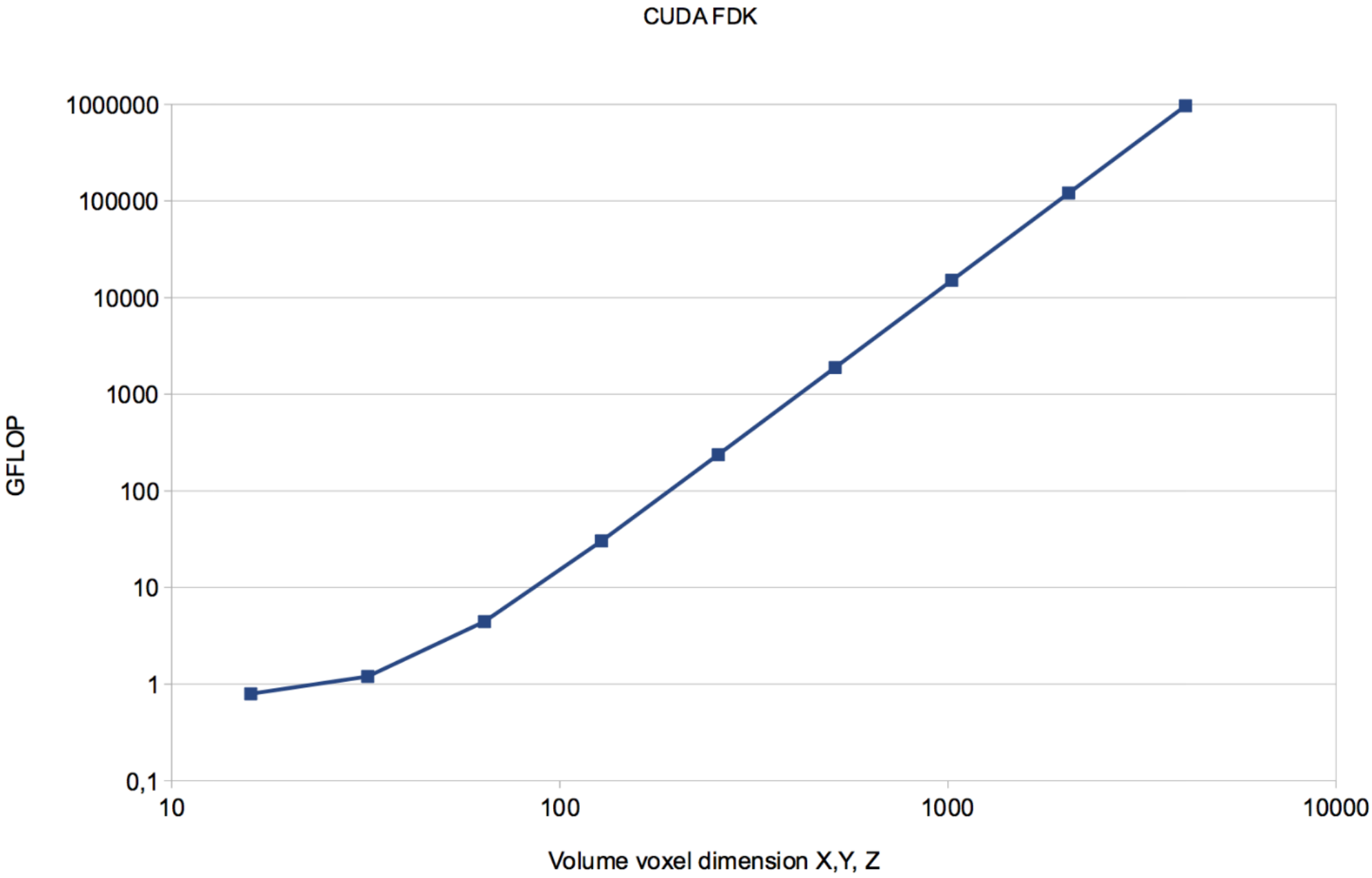
- CT reconstruction software toolbox
  - Fan beam Step and Shoot (flat/curved)
  - Cone beam Step and Shoot (flat/curved)
  - Cone beam helix (flat/curved)
- Including both educational and highly optimized code
- User manuals for CT reconstruction
- CT Reconstruction examples
- To be released this fall under GPL V2 license

skull-base.cfg

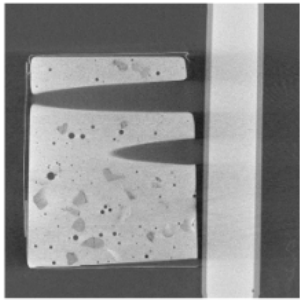
```
# For FDKnpy verification
# This is the base conf for all runs and each run then additionally
# provides the specific settings and resulting paths for use:
# fdk.py skull-base.cfg skull-XxXxX.cfg

log_level=info
proj_filter=hamming
working_directory=/Users/rehr/tmp/CT-Toolbox/demo
projs_per_turn=320
total_turns=1

detector_shape=flat
detector_columns=256
detector_rows=192
detector_pixel_width=0.1552
detector_pixel_height=0.1552
source_distance=115.0
detector_distance=40.0
detector_column_offset=0.5647
detector_row_offset=0.5

precision = float32
```

skull-256x256x256.cfg

```
# FDK npy verification
# This provides the specific reconstruction settings and resulting paths for use:
# fdk.py skull-base.cfg skull-XxXxX.cfg

chunk_size=256
x_voxels=256
y_voxels=256
z_voxels=256
x_min=-15
x_max=15
y_min=-15
y_max=15
z_min=-15
z_max=15

input_precision=uint16
npy_load_input=loadscene#scene_path=input/skull.csv
npy_preprocess_input=flux2proj#zero_norm=0#air_norm=65535
npy_postprocess_output=saveslices#dim=x#save_path=results/skull/slices/x/png/slice.x.%%.
4d.png:saveslices#dim=y#save_path=results/skull/slices/y/png/slice.y.%%.
4d.png:saveslices#dim=z#save_path=results/skull/slices/z/png/slice.z.%%.4d.png
npy_save_output=savevolume#save_path=results/skull/raw/data/recon_volume.bin
```

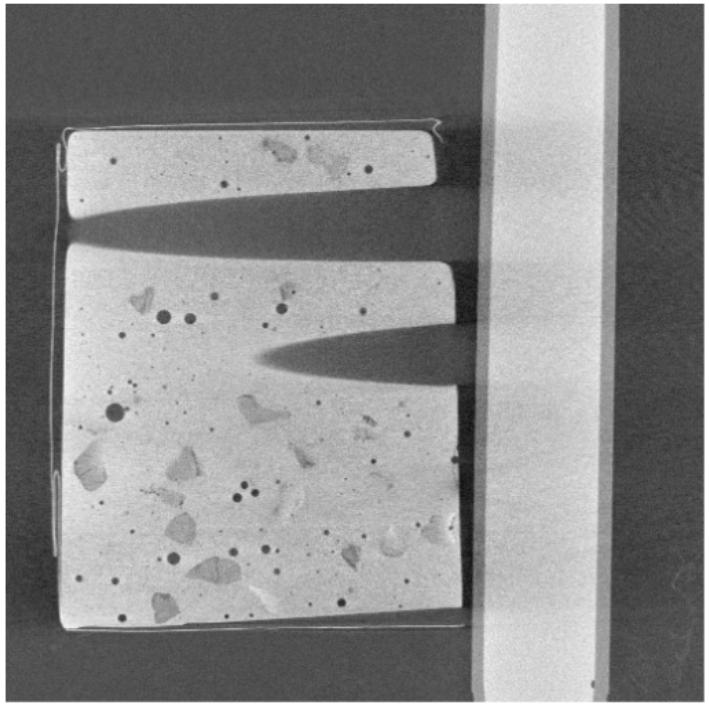

CUDA FDK
1000000
100000
10000
1000
100
10
1
0,1
GFLOP
10
100
1000
10000
Volume voxel dimension X,Y, Z

More of the same...



More of the same…


GULD
BARRE



Invers Radon Transform?

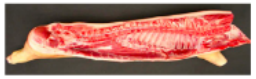
Radon Transform

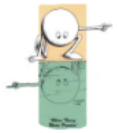
$$Rf(L) = \int_L f(x)|dx|$$

How hard can it be...

CT SCANNING

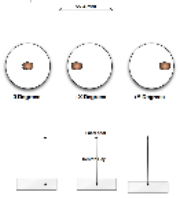
Industrial CT
No time for pretty pictures....



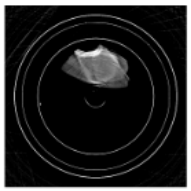


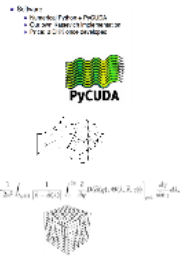
338 GFLOP for a reconstruction

X-Ray Tube



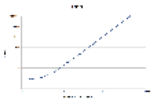
Detector panel





PyCUDA

Toolbox



More of the same...