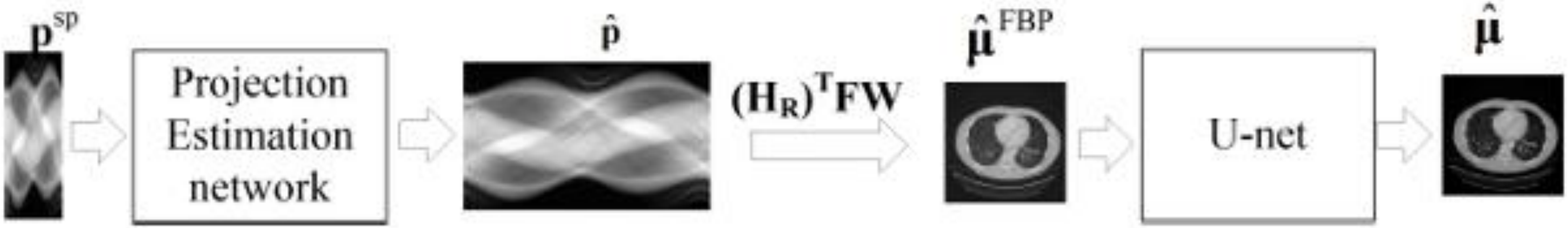
Derenzo Phantom Learned Reconstructions

Bill Worstell

PicoRad->MGH

1/23/2024

DiffDRR



Liang, K., Yang, H. and Xing, Y., 2018. Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction. *arXiv preprint arXiv:1804.04289*.

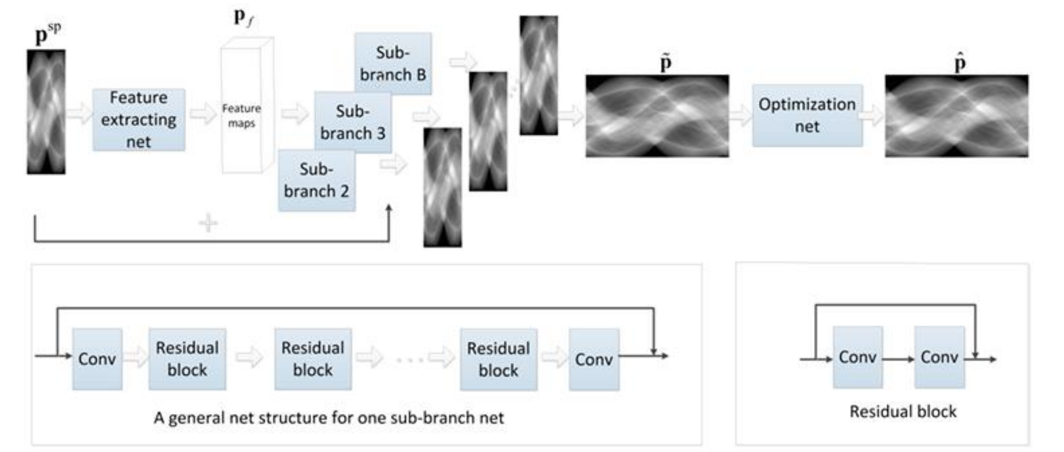
… compare three types of deep learning network architectures:

1) Estimate missing projections by deep learning and reconstruction by FBP afterwards.

2) Reconstruction with FBP using sparse-view data and using deep learning network to reduce artefacts;

3) Network estimating missing projections +FBP +image domain network.

The architectures of these networks are shown in Fig.1, 2, and 3 respectively.



The main structure of this network is consist of residual blocks [22]. All convolutional layers are followed with leaky-relu activation and batch-norm layers[24], except for the output layers of branches and last output layer. The output of this network is reconstructed by FBP to get reconstruction images.

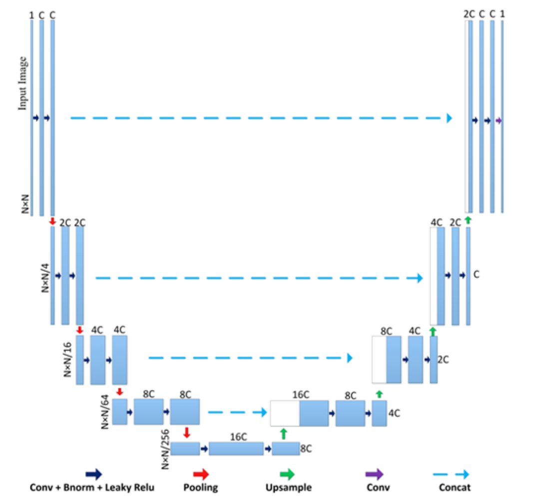


Figure 2: The architecture of U-net in image domain for sparse view CT reconstruction.

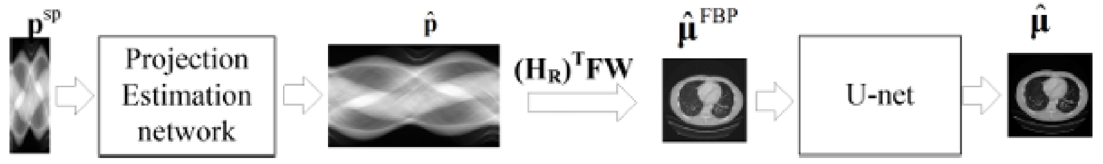
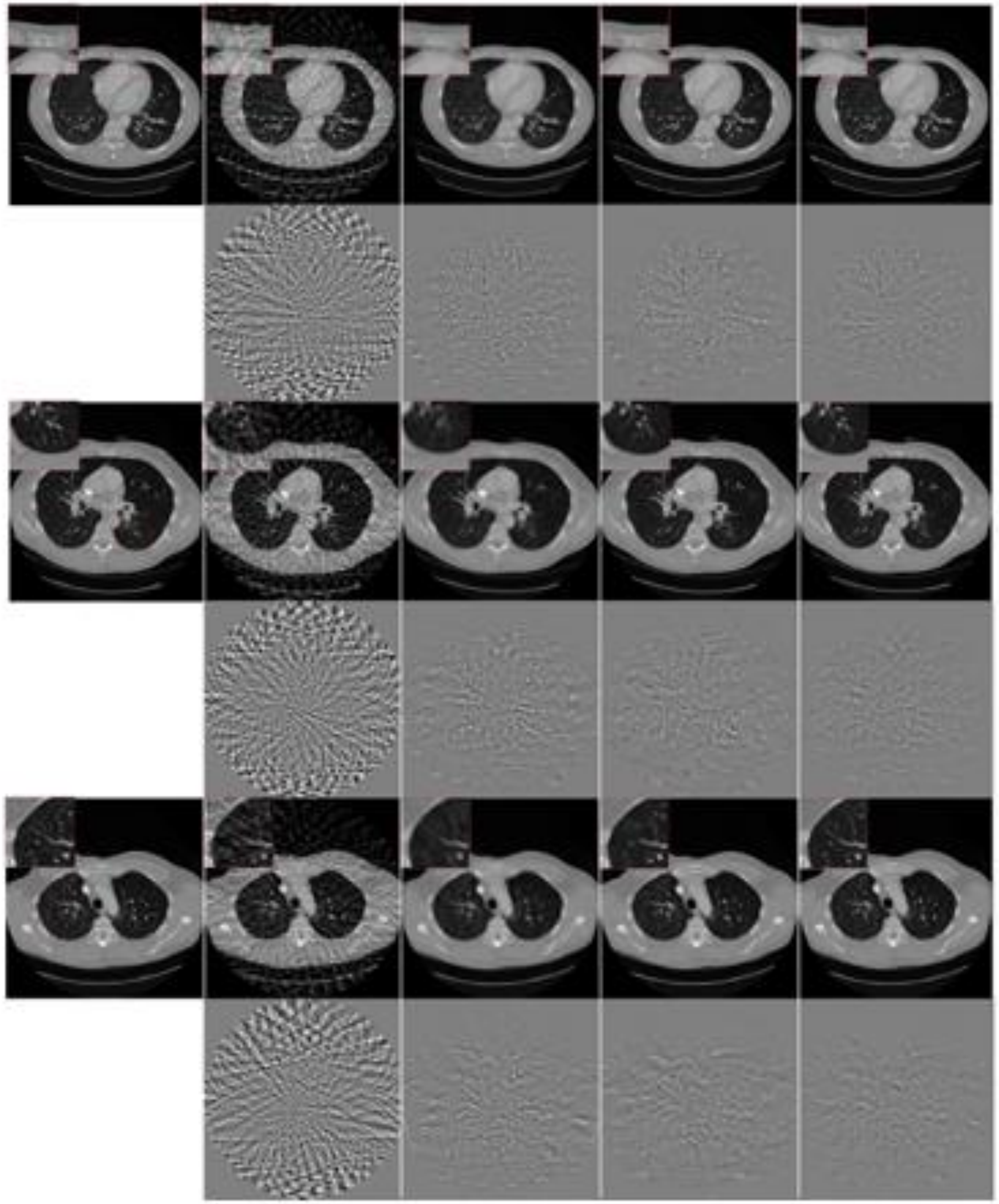


Figure 3: The architecture of comprehensive network combining projection estimation and U-net in image domain for sparse view CT reconstruction.

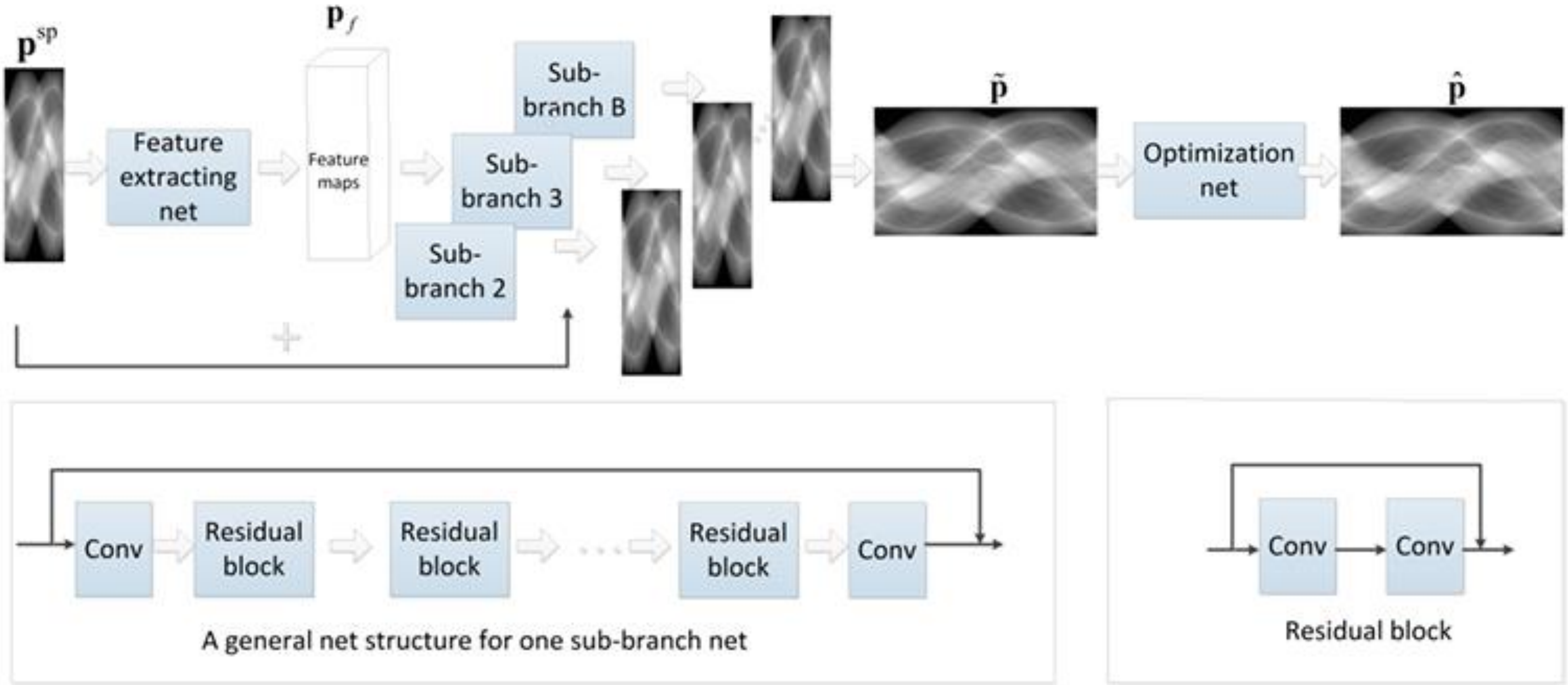


Liang, K., Yang, H. and Xing, Y., 2018. [Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction.] *arXiv preprint arXiv:1804.04289*.

Figure5: Three cases of reconstructions. From left to right: phantoms, 45 view FBP, 45-view projection estimation reconstruction, 45-viewU-Net, and the proposed network reconstructions. In each case, details in the red boxes are displayed on upper left corner. The differences of the four reconstructions of 45-viewdata from the phantom are displayed below the corresponding reconstructions. Images in a same category are in same gray scale

Sparse-angle FBP reconstruction has hivh spatial frequency artifacts that network can learn to correct for

Liang, K., Yang, H. and Xing, Y., 2018. Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction. *arXiv preprint arXiv:1804.04289*.



Liang, K., Yang, H. and Xing, Y., 2018. Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction. *arXiv preprint arXiv:1804.04289*.

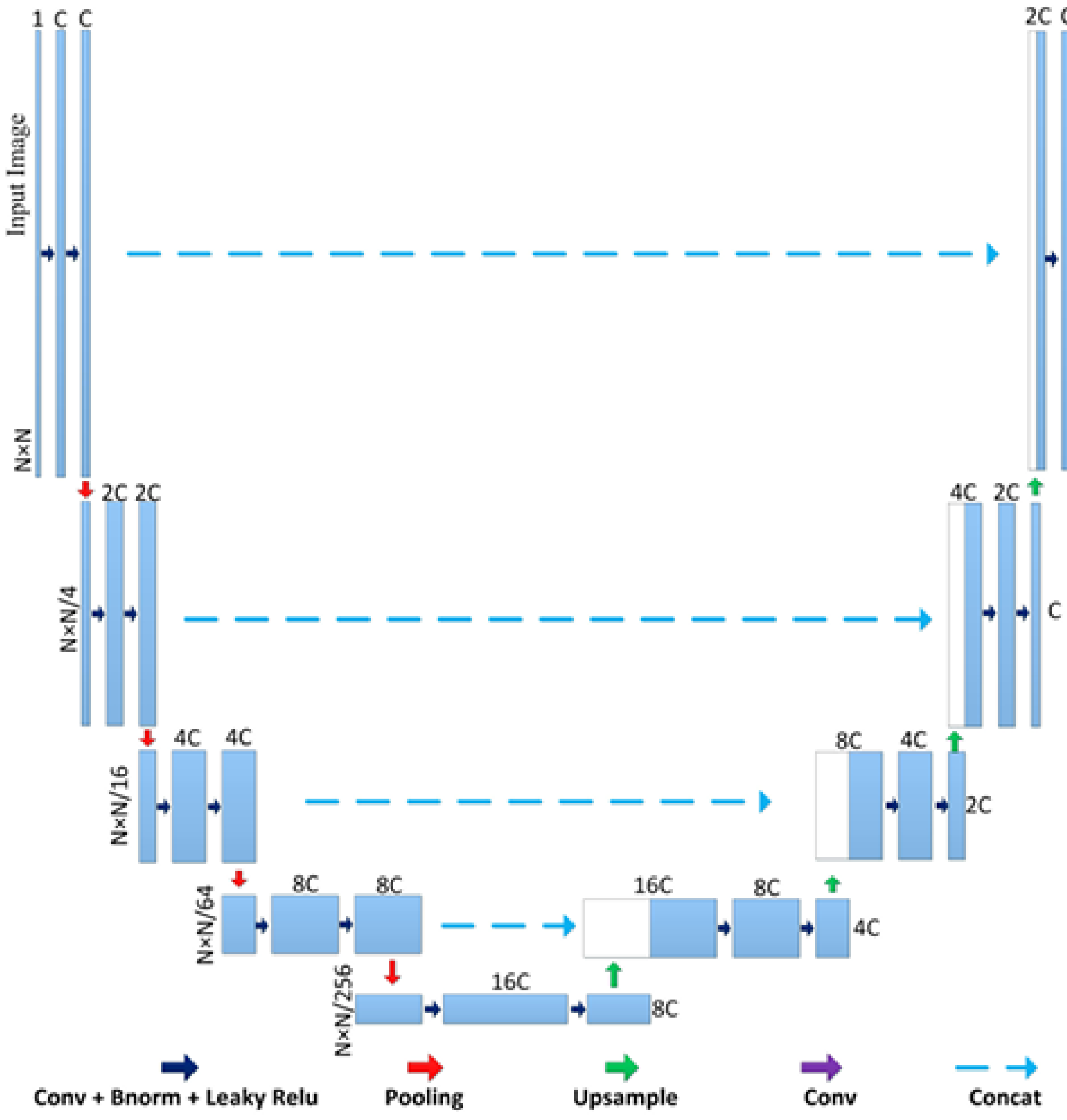
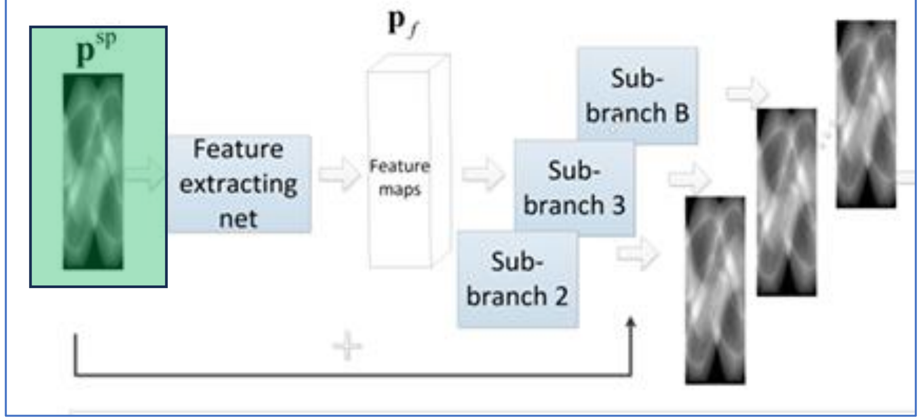
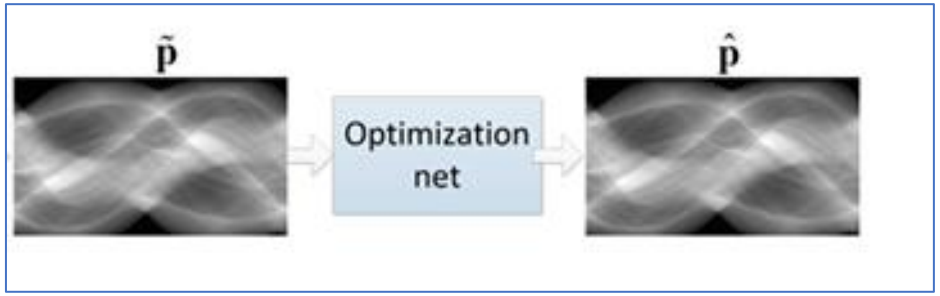


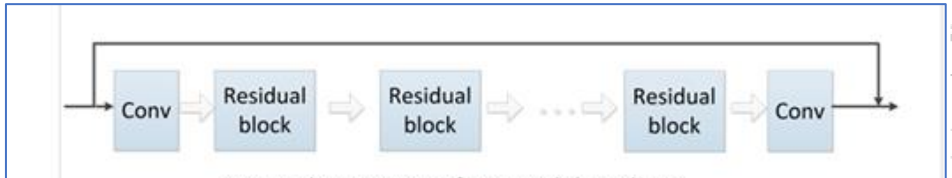
Figure 2: The architecture of U-net in image domain for sparse view CT reconstruction.

Applying U-net in image domain for sparse-view CT is very straightforward. The U-net structure used in this study is as shown in Fig. 2. This network takes FBP reconstruction of sparse-view data as input.

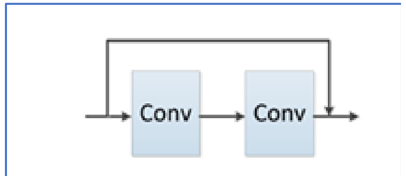
Substitute our DRRs at the input







A general net structure for one sub-branch net



Residual block

Map depth-distorted MGH mphSPECT DRRs onto conventional SPECT projections/sinograms

The projection estimation network is to estimate missing projection data from sparse view projections $\mathbf{p}^{sp}$ to gain complete data [23]. As Fig. 1 shows, the projection estimation network is composed of three functional blocks. The first block of net is feature extracting net. It takes $\mathbf{p}^{sp}$ as input and extracts input projection map's features with convolutional layers. Defining the function of feature extracting block as ϕ_f, the extracted feature maps of three dimension can be denoted as: $\mathbf{p}_f = \phi_f(\mathbf{p}^{sp})$. The second block is consist of projection estimation sub-branches. According to the CT scanning geometry, missing views are grouped into subsets according to their neighborhood relationship,e.g., B subsets in case of sparsity factor being B. Considering different projection subsets have different angular distance from the acquired projection $\mathbf{p}^{sp}$, $B-1$ sub-branch nets are designed to estimate the $B-1$ subsets of projections separately [1]. These nets take feature maps as inputs and finally generate $\Delta\mathbf{p}_1, \Delta\mathbf{p}_2, ..., \Delta\mathbf{p}_B$ to estimate $\widetilde{\mathbf{p}}_1, \widetilde{\mathbf{p}}_2, \ldots, \widetilde{\mathbf{p}}_B$. Defining the estimation operator as φ_b , then $\widetilde{\mathbf{p}}_b = \Delta\mathbf{p}_b + \mathbf{p}^{sp} = \varphi_b(\mathbf{p}_f) + \mathbf{p}^{sp}$. At the end of prj-estimating sub-branches, the estimated projections together with input sparse-view projection are concatenated in the order of view angles to form full-view projections $\widetilde{\mathbf{p}} = \mathrm{concat}(\widetilde{\mathbf{p}}_1, \widetilde{\mathbf{p}}_2, \ldots, \widetilde{\mathbf{p}}_B)$. A third block is added to unify the projections coming from different sub-branches so that the full-view projection are further tuned to be consistent as a complete dataset. The final output of full-view projection estimated is denoted as $\widehat{\mathbf{p}}$. The main structure of this network is consist of residual blocks [22]. All convolutional layers are followed with leaky-relu activation and batch-norm layers [24], except for the output layers of branches and last output layer. The output of this network is reconstructed by FBP to get reconstruction images.

FBP_SPECT.ipynb

File Edit View Insert Runtime Tools Help All changes saved

See https://pytomography.readthedocs.io/en/latest/usage.html

and in particular https://pytomography.readthedocs.io/en/latest/notebooks/t_fbp.html

Modified to run from Google Colab: https://colab.research.google.com/

Data stored on Google Drive in path = '/content/drive/MyDrive/pytomography_tutorial_data/simind_tutorial/'

May require restart after first run

Tested with V100 GPU on Colab

https://pytomography.readthedocs.io/en/latest/notebooks/t_fbp.html

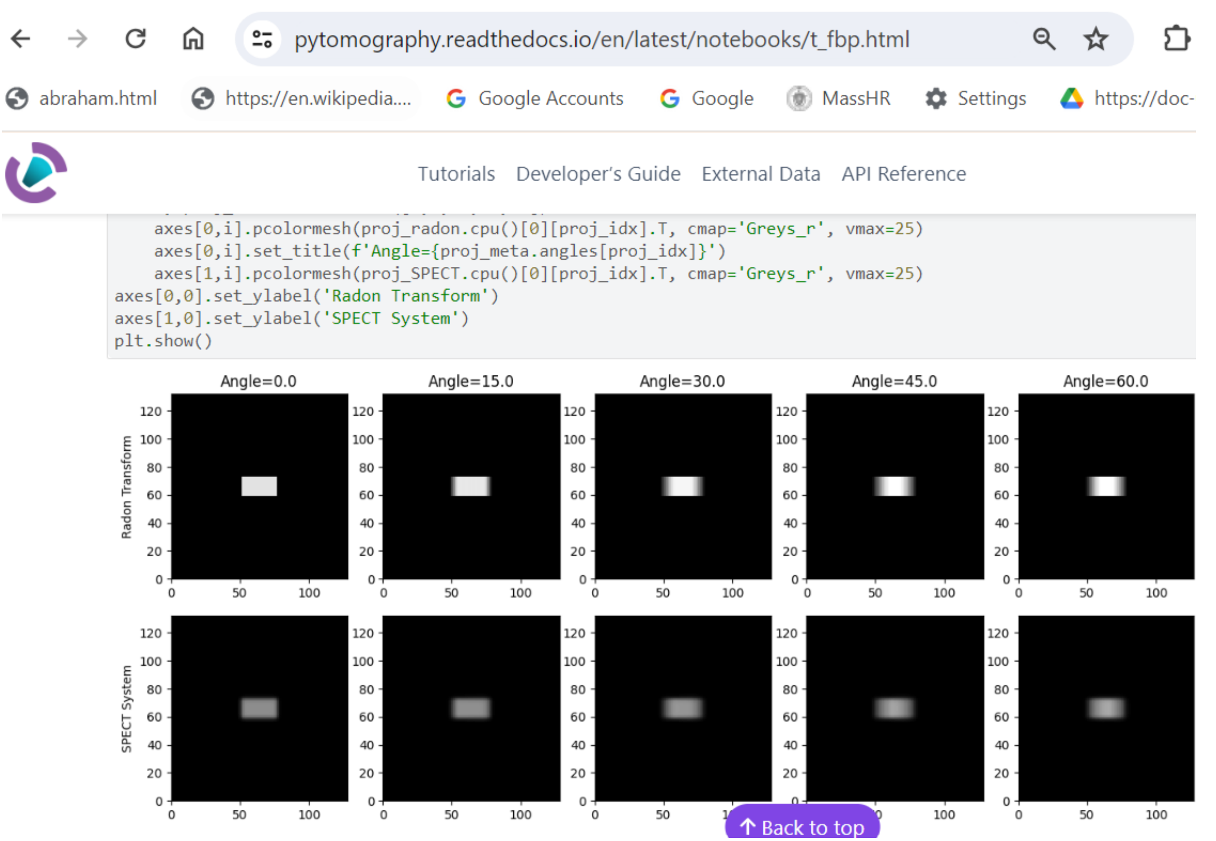
Tutorials Developer's Guide External Data API Reference

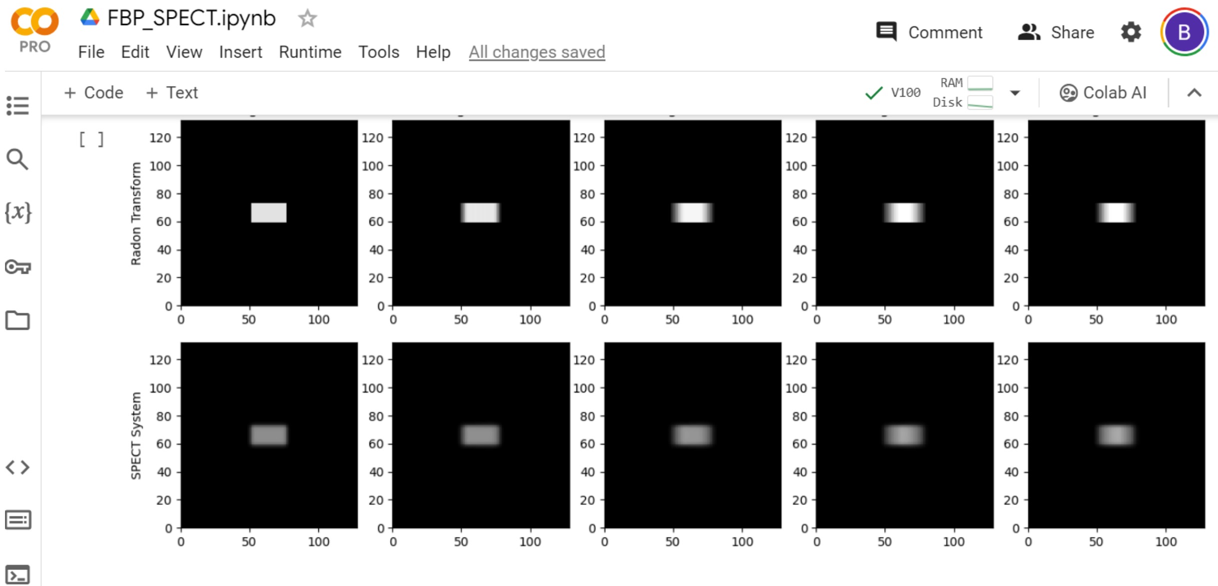
Filtered...

Filtered Back Projection

We'll use the classes of PyTomography to implement filtered back projection in SPECT.

Interface with pytomography supports radon forward projection and FBP reconstruction, as well as modeling a conventional SPECT system with the same source and arbitrary angular sampling





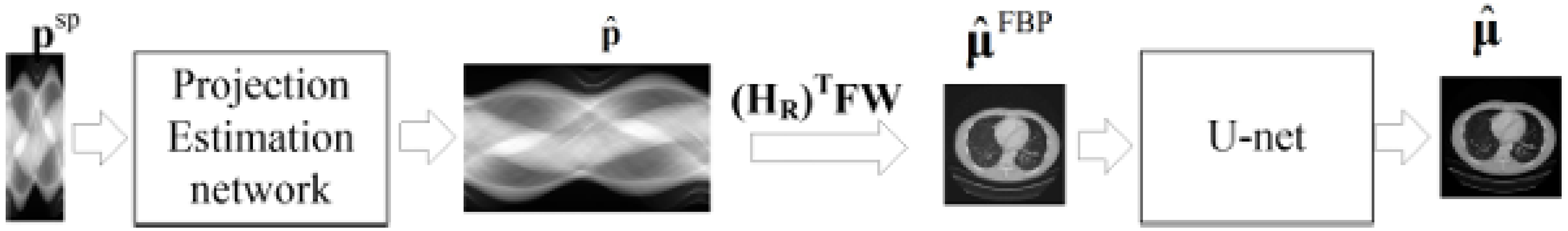


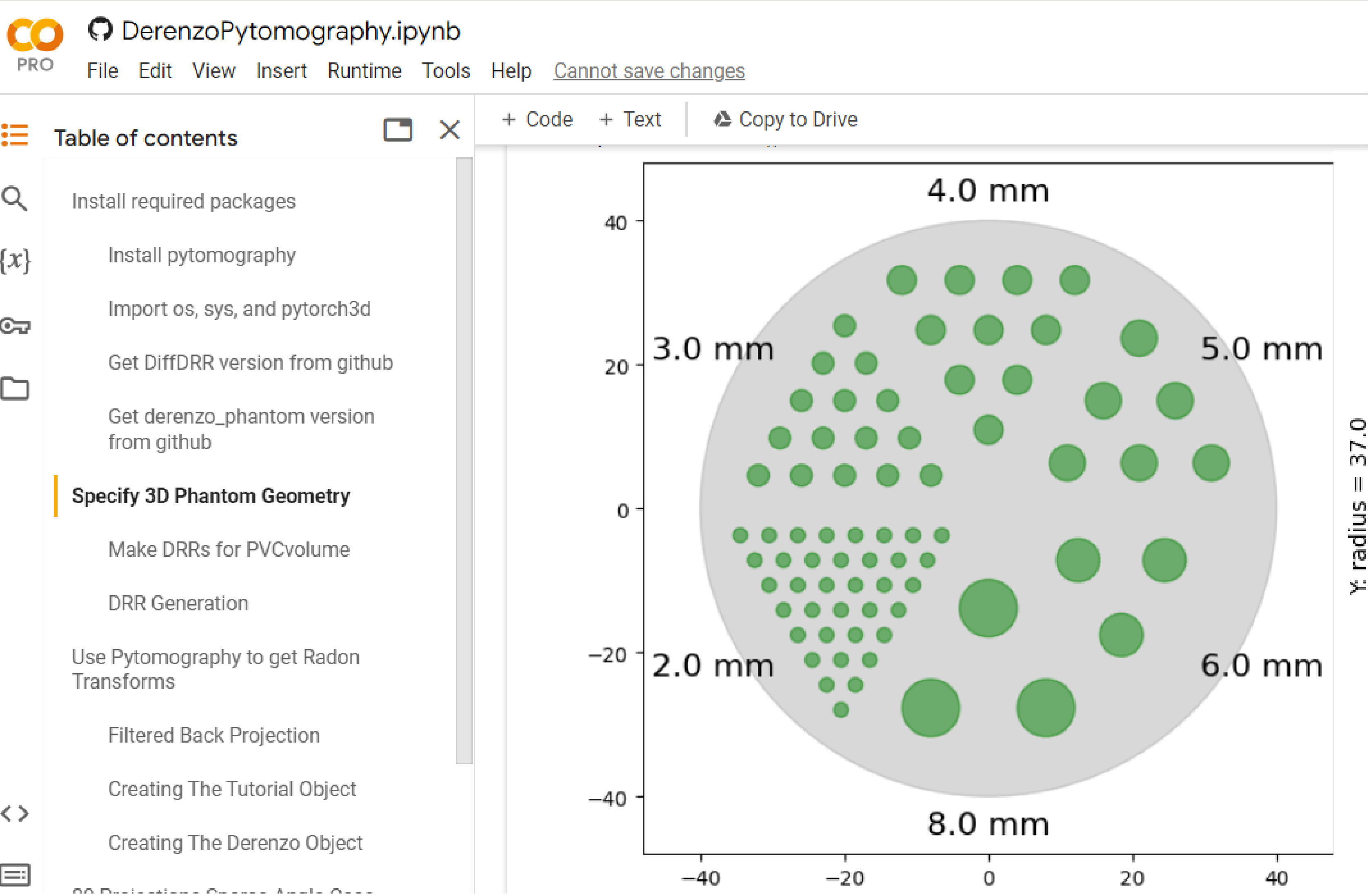
Figure 3: The architecture of comprehensive network combining projection estimation and U-net in image domain for sparse view CT reconstruction.

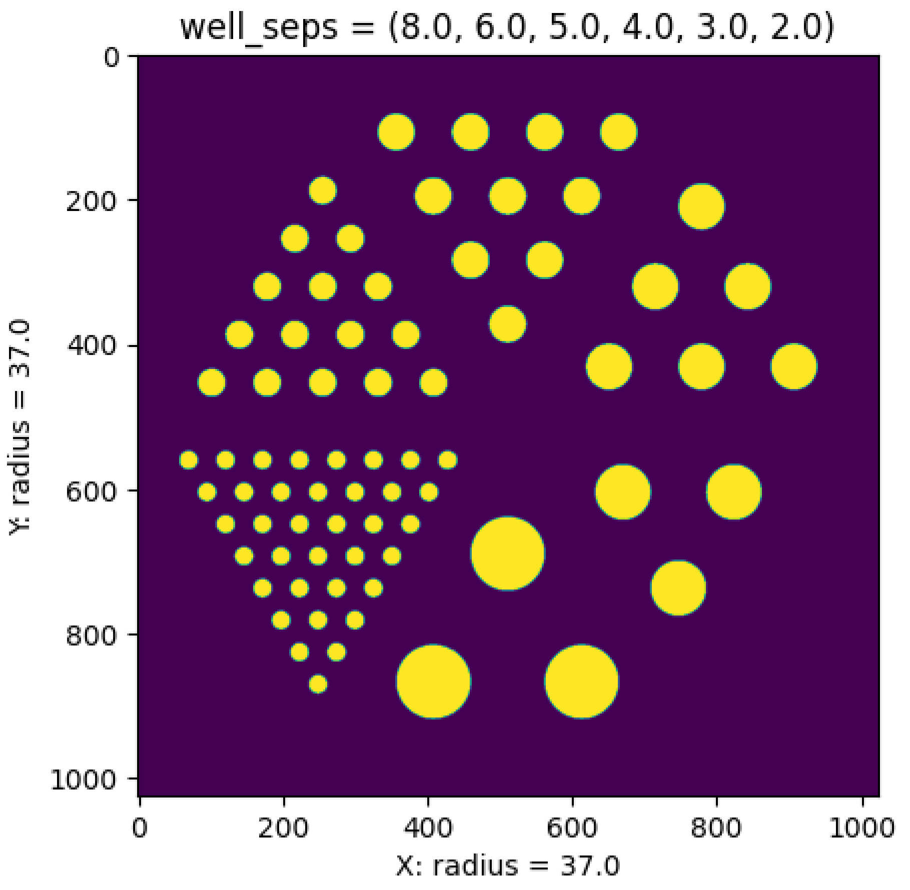
Liang, K., Yang, H. and Xing, Y., 2018. Comparison of projection domain, image domain, and comprehensive deep learning for sparse-view X-ray CT image reconstruction. *arXiv preprint arXiv:1804.04289*.

We can adapt this network to take input from MGH mphSPECT data tensors and to estimate csonventional SPECT projections, using supervised learning and 2D or 3D U-nets in pytorch

https://colab.research.google.com/github/BillWorstell/derenzo_phantom

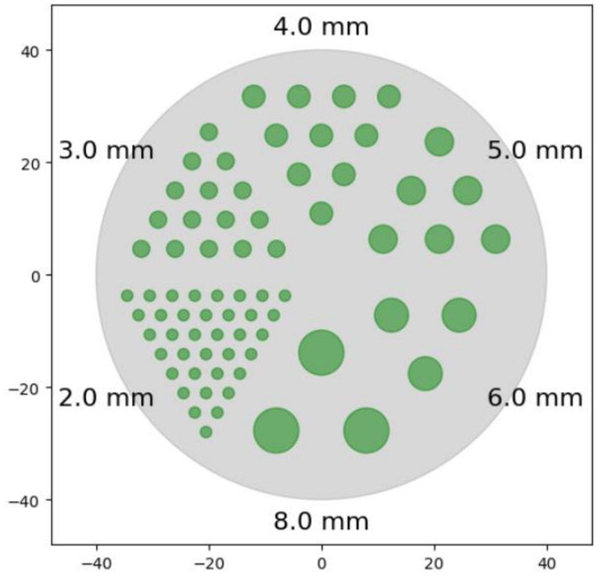
Makes a resolution phantom object with arbitrary geometry

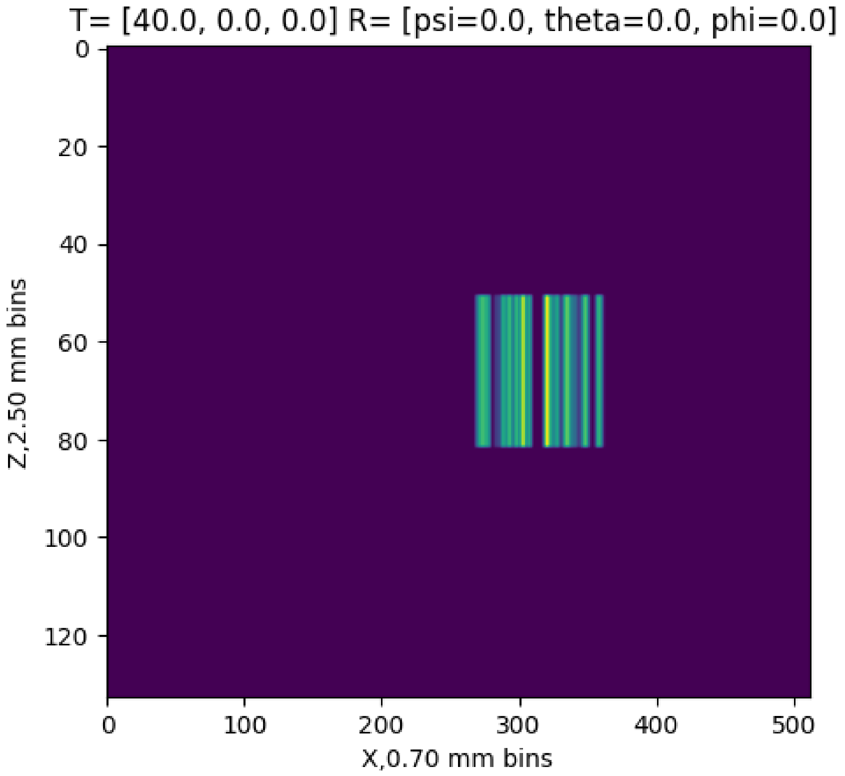


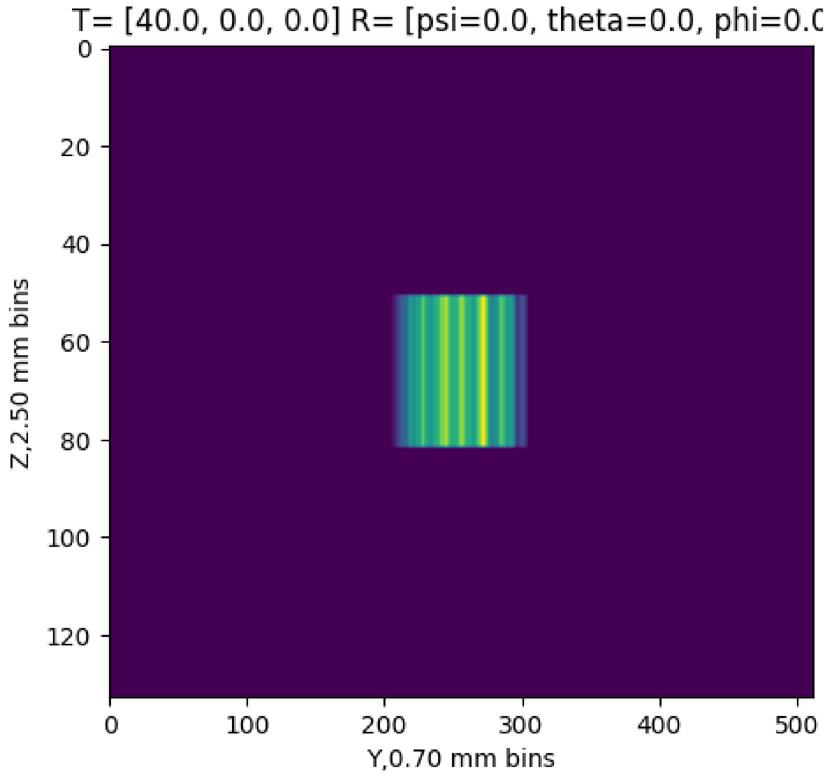


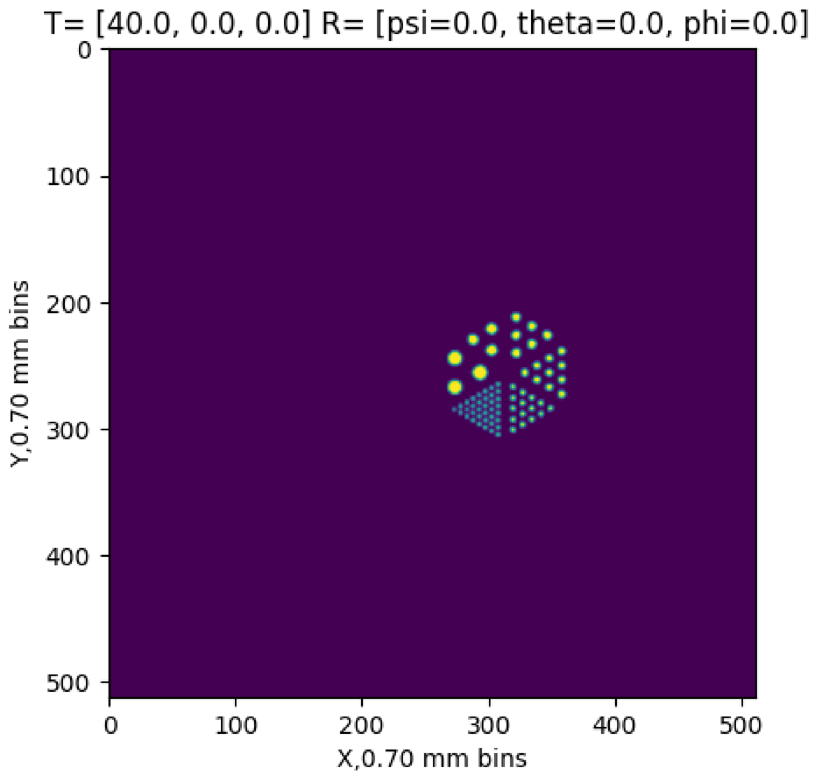
https://colab.research.google.com/github/BillWorstell/derenzo_phantom

Independently control phantom position, orientation, and size within volume





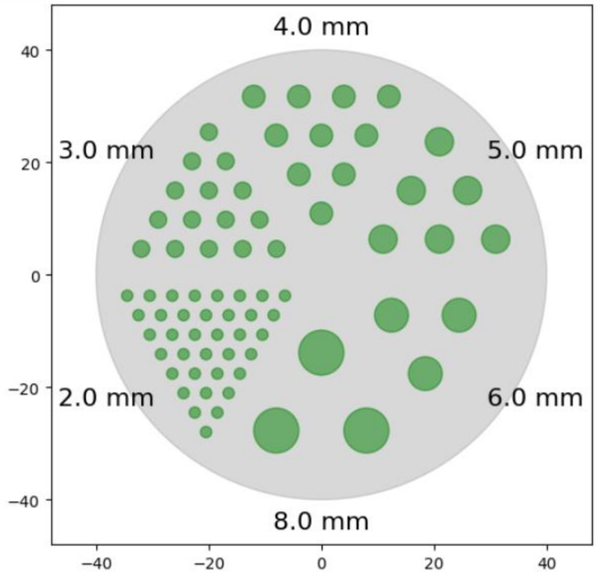


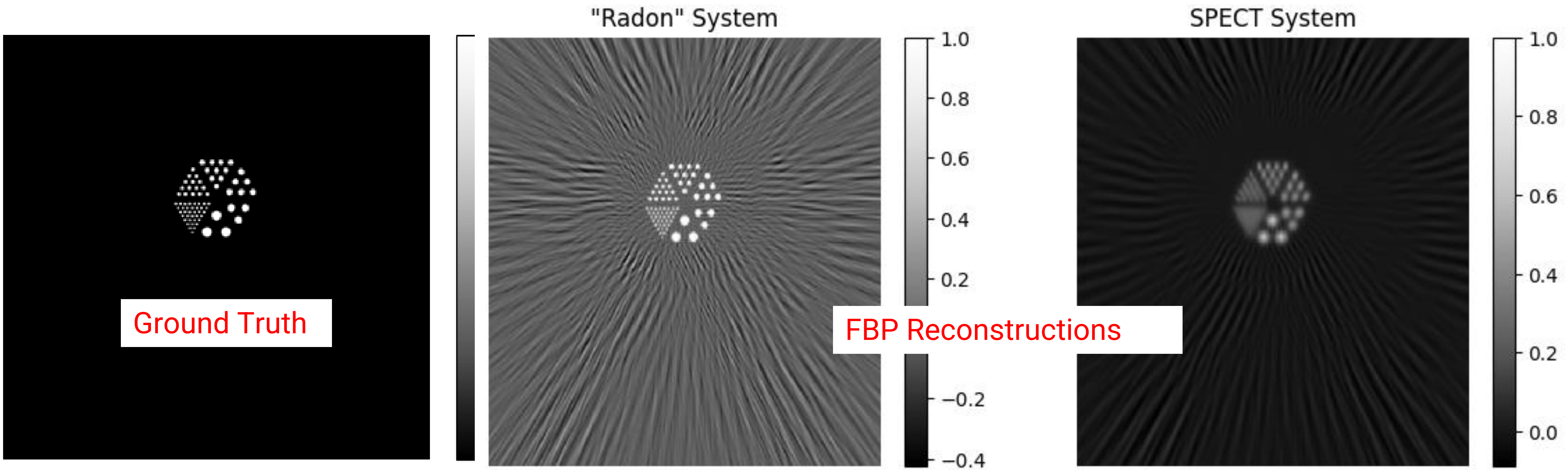


https://colab.research.google.com/github/BillWorstell/derenzo_phantom

Interface with pytomography supports radon forward projection and FBP reconstruction, as well as modeling a conventional SPECT system with the same source and arbitrary angular sampling

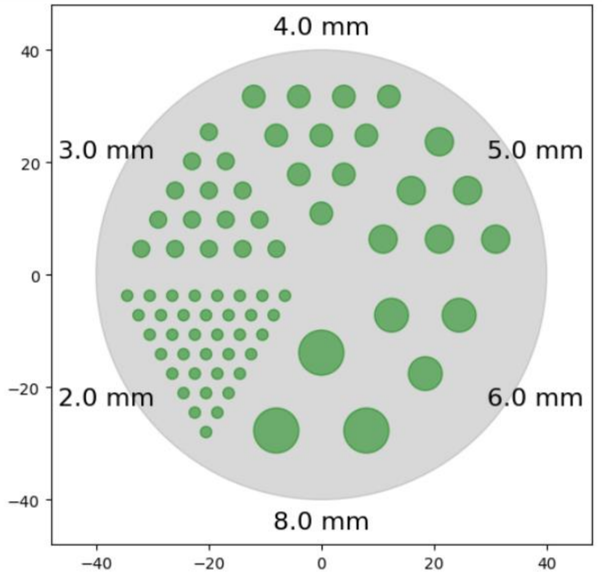
80 Projections Sparse Angle Case

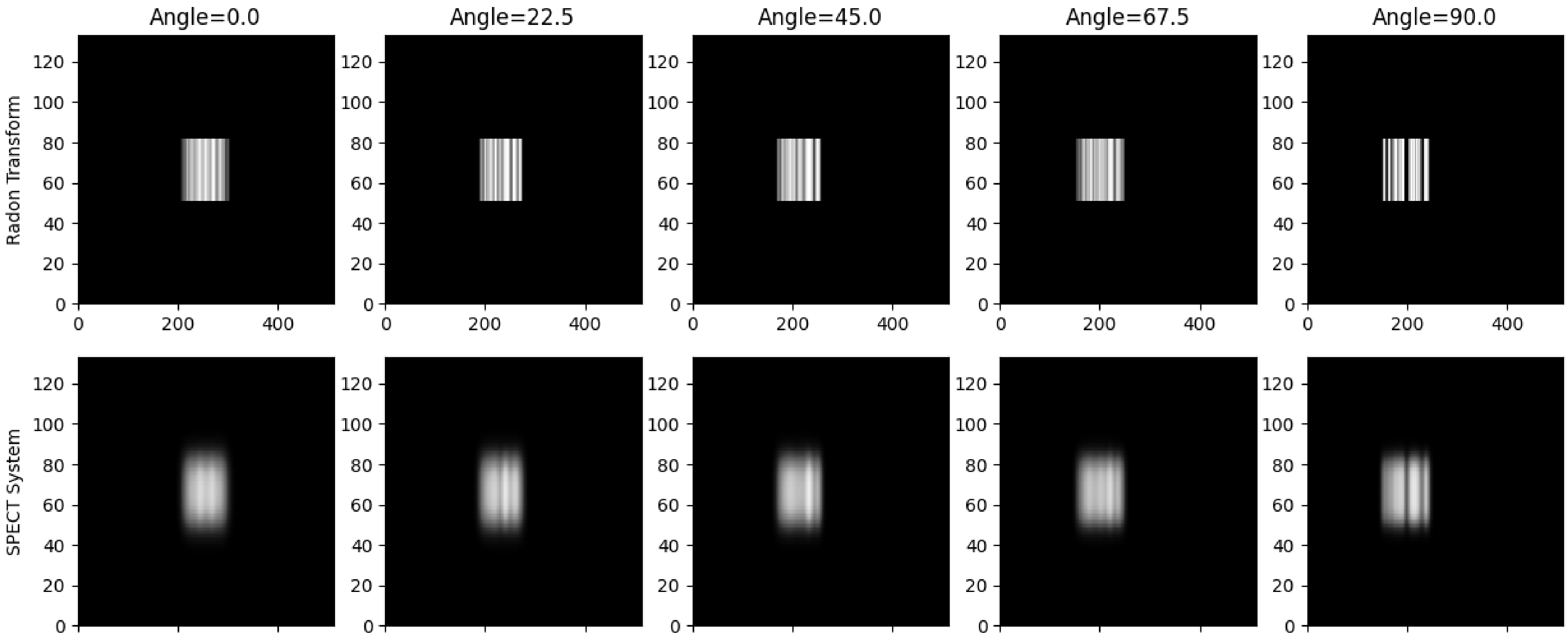




https://colab.research.google.com/github/BillWorstell/derenzo_phantom

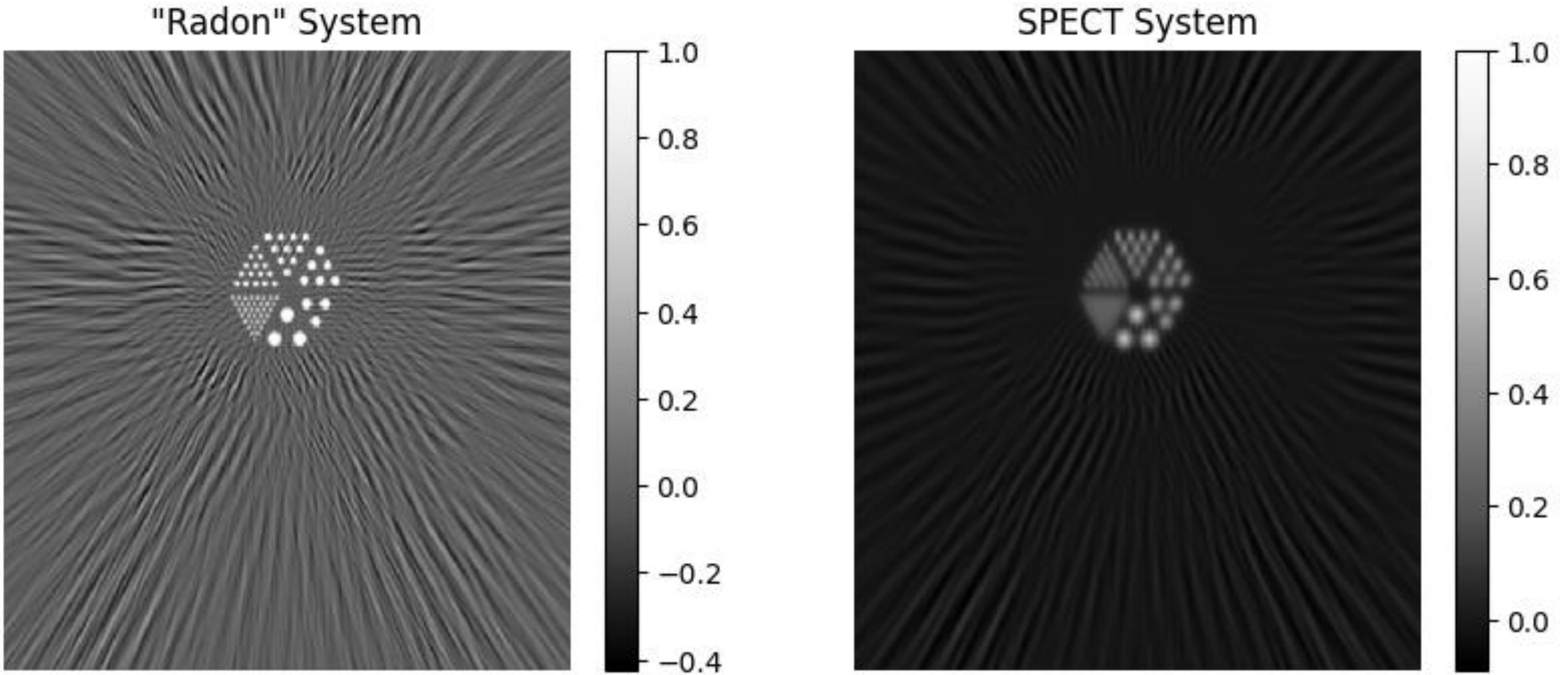
Interface with pytomography supports radon forward projection, as well as modeling a conventional SPECT system with the same source and arbitrary angular sampling





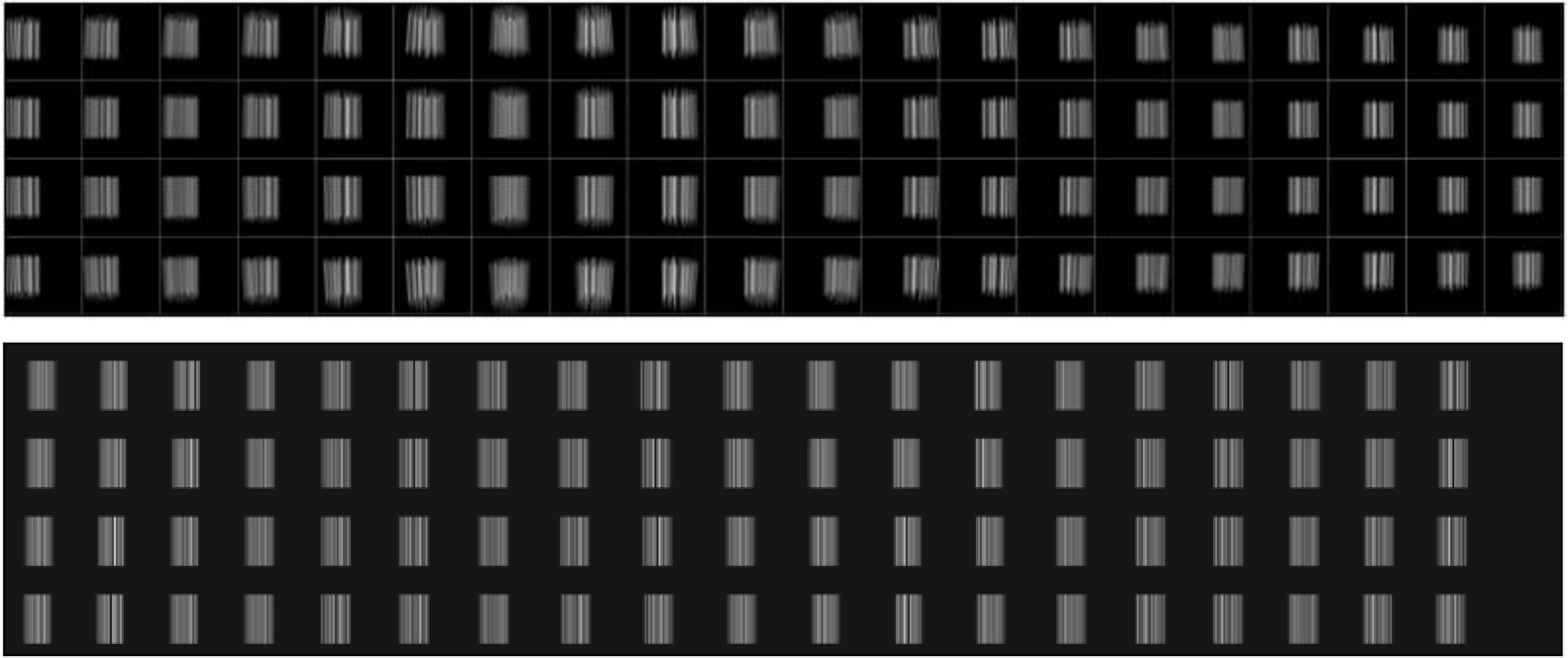
https://colab.research.google.com/github/BillWorstell/derenzo_phantom

Interface with pytomography supports FBP reconstruction, as well as modeling a conventional SPECT system with the same source and arbitrary angular



https://colab.research.google.com/github/BillWorstell/derenzo_phantom

By pairing DiffDRR mphSPECT projections (top) with pytomography radon and SPECT forward projection we can learn our acquisition geometry and map to convention system geometries



https://github.com/wolny/pytorch-3dunet

README | MIT license

DOI 10.5281/zenodo.10456011 | Conda Build passing | Last updated 03 Jan 2024 | license MIT

pytorch-3dunet

PyTorch implementation of 3D U-Net and its variants:

- `UNet3D` Standard 3D U-Net based on 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation
- `ResidualUNet3D` Residual 3D U-Net based on Superhuman Accuracy on the SNEMI3D Connectomics Challenge