Reducing the number of projections in CT imaging using domain-transform manifold learning

Avilash Cramer, Neha Koonjoo, Bo Zhu, Rajiv Gupta, Matthew S. Rosen

Contact info: Avilash@mit.edu

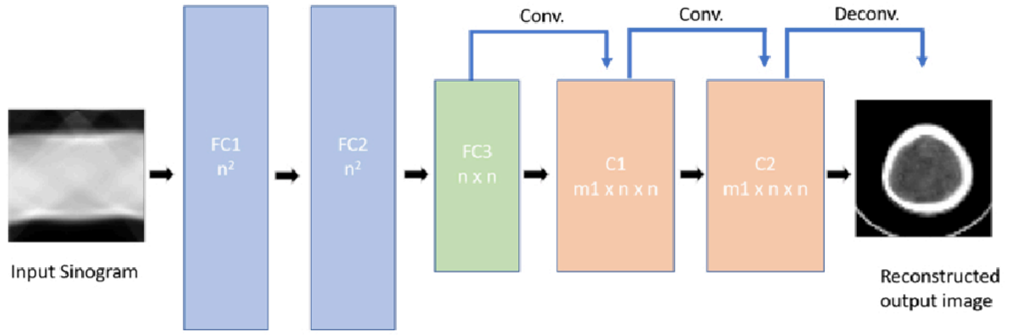
PURPOSE

Many non-rotating x-ray computed tomography (CT) frameworks are being developed, with discrete sources, including a system previously reported by our group (Cramer et al *AAPM* 2018, 2019; Cramer et al *Sci Rep* 2018). For these discrete-source systems, the number of x-ray sources is directly linked to the number of projection angles.

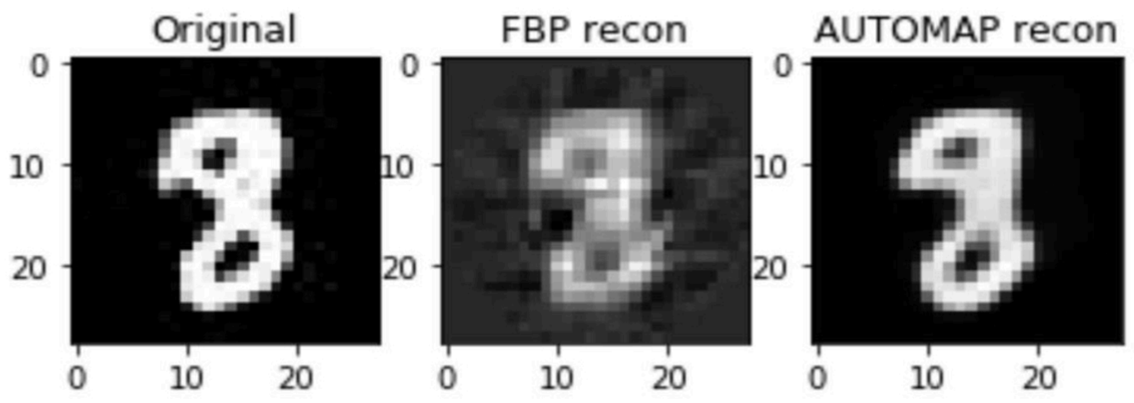
If a CT image could be reconstructed with fewer projections, it could mean not only less dose to the patient, but also ease the engineering constraints or a non-rotating CT system as each individual source can occupy a larger arc angle.

METHODS

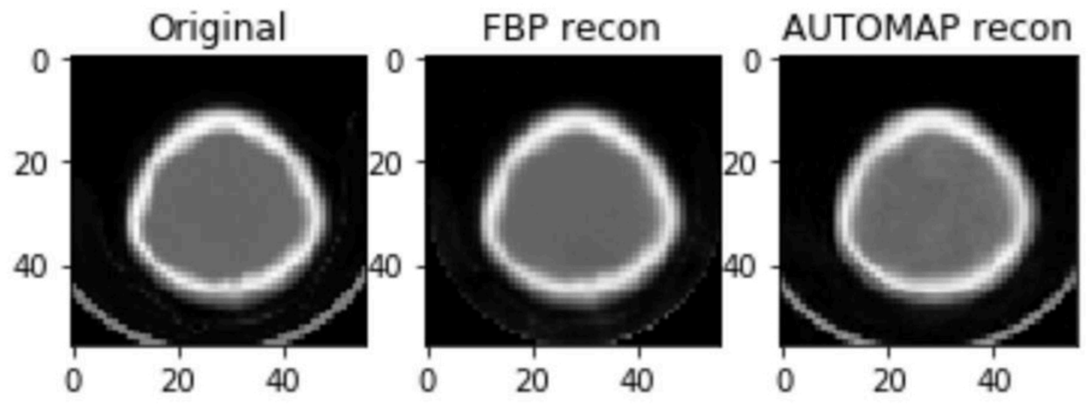
We adapted AUTOMAP (AUTOmated transform by Manifold APproximation) for CT image reconstruction (Zhu, *Nature*, 2018). AUTOMAP is an end-to-end generalized reconstruction framework, implemented with a deep neural network architecture composed of two fully connected layers followed by a sparse convolutional autoencoder.



RESULTS



AUTOMAP clearly outperforms conventional filtered back project (FBP) in the reconstruction of 28-projection sinograms. Trained on 60,000 sinograms of MNIST numbers.



Example reconstruction using AUTOMAP, on 56 x 56 downsampled CT images. Sinogram created from 56 evenly spaced projections, trained on a dataset of 60,000 images. RMSE error from FBP = 0.344; from AUTOMAP = 0.036.

NEXT STEPS

The memory requirements of extending AUTOMAP to full-sized CT imaging are a significant technical hurdle.

Due to the 2 fully connected layers of AUTOMAP, the individual matrix multiplications become too large (by an order of magnitude) for a standard 16 GB GPUs. We are exploring three hardware architectures, detailed below:

(1) Using a 512 GB CPU, and accepting a ~5x computation time penalty
(2) Breaking up individual matrix multiplications (block matrix decomposition), and performing each sub-matrix multiplication on a separate GPU
(3) Breaking up individual matrix multiplications (block matrix decomposition), and performing each sub-matrix multiplication on a single Large Model Support (LMS) 16 GB GPU optimized for fast loading and unloading