

# Iterative reconstruction artefact removal using null-space networks

Author(s) names withheld

EMAIL(S) WITHHELD

Address withheld

Editors: Under Review for MIDL 2020

## Abstract

Incorporation of resolution modelling (RM) into iterative reconstruction produces Gibbs ringing artefacts which adversely affect clinically used metrics such as  $SUV_{\max}$ . We propose the use of a null space network as a regulariser to compensate for these artefacts without introducing bias.

**Keywords:** Null net, data consistency, image reconstruction, PET

## 1. Introduction

Maximum likelihood expectation maximisation (MLEM) is a versatile iterative algorithm frequently used in clinical PET image reconstruction. Incorporating resolution modelling (RM) into MLEM results in resolution recovery but also introduces Gibbs ringing artefacts. Since these artefacts affect clinically important metrics such as maximum standardised uptake values ( $SUV_{\max}$ ), there is debate as to whether RM should be used at all (Alessio, Rahmim, and Orton 2013).

Ringing artefacts are caused by a sharp drop in recovery of high frequency components. The unrecoverable data exists in the null space of the imaging system. We propose the use of a deep null space network (Schwab, Antholzer, and Haltmeier 2019a) to fill in this null space, thereby removing artefacts in a data-consistent manner. Null nets have recently been applied to filtered backprojection (FBP) of sparse photoacoustic tomography (PAT) (Li et al. 2019) and undersampled Radon transforms (Schwab, Antholzer, and Haltmeier 2019b).

## 2. Methods

Data consistency (more specifically, model consistency) is enforced by incorporating MLEM reconstruction into the training of the proposed network. Once trained, the network can be applied as a post-reconstruction step to test data. Due to the presence of noise in real data, full consistency is not desirable. Instead, consistency is only enforced here with respect to resolution degradation and recovery.

**Data:** 20 *BrainWeb* phantoms are modified to have PET-like intensities and spherical lesions of diameters ranging from 2 to 15 mm as described in da Costa-Luis (2019). The *BigBrain* FDG PET phantom (Belzunce 2018) was also registered with the *BrainWeb* data and Poisson noise added corresponding to 30 M counts. Resolution degradation is achieved by Gaussian smoothing with 5 mm full width at half maximum (FWHM). This data is then

reconstructed using 300 MLEM iterations . In the case of purely performing resolution recovery, MLEM is equivalent to the Richardson-Lucy (RL) algorithm (Appendix A).

**Training:** 18 noise-free *BrainWeb* phantoms are used for training, and 2 for validation. The *BigBrain* phantom is used for testing. The task is to transform RL reconstructions into artefact-free ground truth predictions in a data-consistent manner. Consistency means the forward model (Gaussian smoothing) should produce identical results whether applied to the input (RL reconstructions) or to the predictions.

The proposed post-processing step is:

$$P(\hat{\mathbf{x}}) = \hat{\mathbf{x}} + N(\hat{\mathbf{x}}) - \mathcal{R}_k(G(N(\hat{\mathbf{x}}))), \quad (1)$$

where  $\mathcal{R}_k$  represents  $k$  MLEM (Richardson-Lucy) iterations,

$G$  applies the forward model (Gaussian smoothing),

$N$  is a 4-layer convolutional network, and

$\hat{\mathbf{x}}$  is reconstructed (using  $\mathcal{R}_k$ ) data.

$N$  consists of 4 fully 3D convolutional layers with unit stride, width 3, and zero-padding; each of which are followed by a *ReLU* activation function. The first 3 layers have 32 kernels, while the final layer has 1. The normalised root mean square error (NRMSE) is used as the loss function:

$$L(P; \mathbf{y}) = \sum_n \sqrt{\|\mathbf{y}_n - P(\mathcal{R}_k(G(\mathbf{y}_n)))\|^2 / \|\mathbf{y}_n\|^2}, \quad (2)$$

where  $\mathbf{y}_n$  is the  $n^{\text{th}}$  ground truth volume.

The Adam optimiser is used with learning rate  $10^{-3}$  and convolutional kernel regularisation with weighting factor  $10^{-4}$ . Once trained,  $P$  satisfies  $G(P(\hat{\mathbf{x}})) = G(\hat{\mathbf{x}})$ .

For comparison, a network  $C$  (with the same architecture as  $N$ ) is trained on the same data. NRMSE is also used as the loss for  $C$ , i.e.  $L(C; \mathbf{y})$ . The network  $C$  thus performs the same post-processing task as  $P$ , but without any consistency constraint.

### 3. Results

$P$  has better generalisability than  $C$  due to the consistency constraint. Similar qualitative results (not shown) were observed for real data. Future work will consider incorporating the PET forward and backprojectors into the reconstruction.

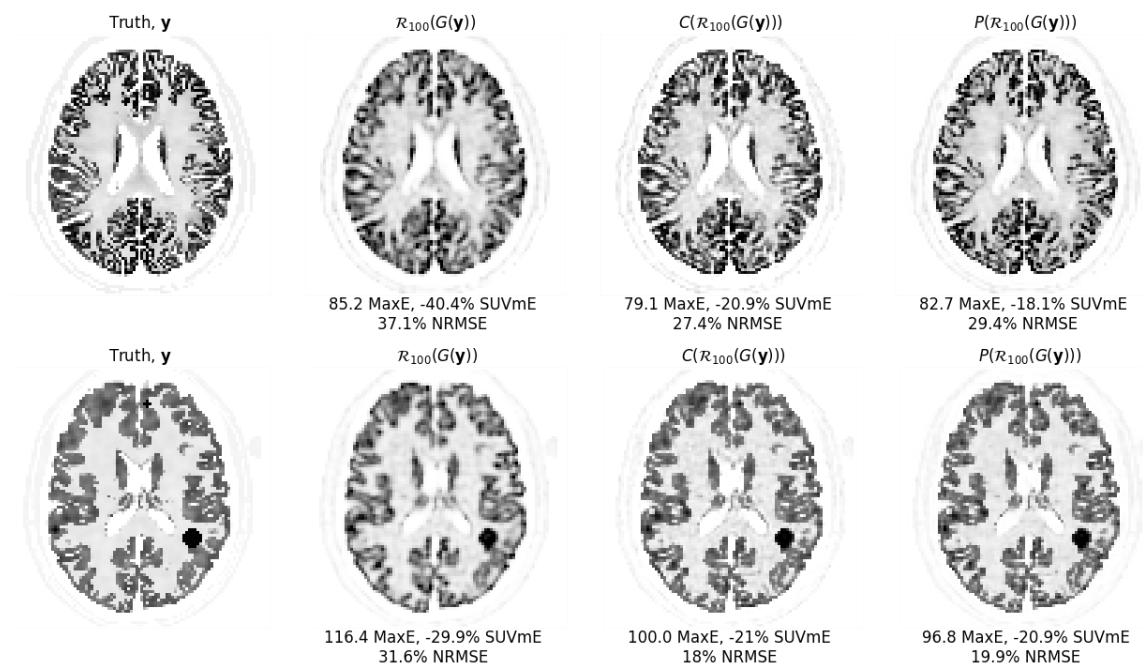


Figure 1: Results for test (top row) data (not used during training). Training data is also shown in the bottom row.

## References

- Alessio, Adam M, Arman Rahmim, and Colin G Orton. 2013. “Resolution modeling enhances PET imaging.” *Medical Physics* 40 (12). AAPM - American Association of Physicists in Medicine: 120601. doi:[10.1118/1.4821088](https://doi.org/10.1118/1.4821088).
- Belzunce, Martin Alberto. 2018. “High-Resolution Heterogeneous Digital PET [18F]FDG Brain Phantom based on the BigBrain Atlas.” Zenodo. doi:[10.5281/zenodo.1190598](https://doi.org/10.5281/zenodo.1190598).
- da Costa-Luis, Casper O. 2019. “BrainWeb-Based Multimodal Models of 20 Normal Brains.” doi:[10.5281/zenodo.3269888](https://doi.org/10.5281/zenodo.3269888).
- Li, Housen, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. 2019. “NETT: Solving Inverse Problems with Deep Neural Networks,” July, 1–21. <http://arxiv.org/abs/1803.00092v2>.
- Schwab, Johannes, Stephan Antholzer, and Markus Haltmeier. 2019a. “Deep null space learning for inverse problems: convergence analysis and rates.” *Inverse Problems* 35 (2): 025008. doi:[10.1088/1361-6420/aaf14a](https://doi.org/10.1088/1361-6420/aaf14a).
- . 2019b. “Big in Japan: Regularizing Networks for Solving Inverse Problems.” *Journal of Mathematical Imaging and Vision*, October, 1–23. doi:[10.1007/s10851-019-00911-1](https://doi.org/10.1007/s10851-019-00911-1).

**Appendix A. Richardson-Lucy Iterative Reconstruction**

---

**Algorithm 1:** Richardson-Lucy (RL)

---

**Input:**  $m$ : blurred image,  $\sigma$ : Gaussian blur parameter,  $k$ : number of iterations**Output:**  $y$ : reconstructed image $y \leftarrow \text{ones}(\text{shape}(m))$ **for**  $i \leftarrow 1$  **to**  $k$  **do**     $y \leftarrow y \times G_\sigma(m \div G_\sigma(y))$                       //  $G$  is a Gaussian blur function**end**

---