

3D Deconvolution Microscopy with a Learned Regularizer

Thanh-an Pham

February 10, 2025

1 Introduction

This report is a summary of the 3D deconvolution with a learned regularizer [1, 2]. The algorithms are implemented within the Pyxu framework (v2), while the training procedure is performed via Pytorch (see Section 2.2.2). The code is freely available (Github or as a PIP package).¹ Although the performance evaluation was performed on a multi-view confocal microscope (i.e., Airyscan), the code is compatible for any deconvolution task, from one to any number of views.

2 Methods

2.1 Problem Formulation

We propose a 3D deconvolution method with a learned regularizer [2]. In this approach, we deploy a data-driven regularizer with a convergent image-reconstruction algorithm. Recent works [1, 2] have shown that it outperforms reconstruction methods with convex regularizers as well as the popular BM3D denoiser. We aim at recovering the 3D image $\mathbf{x} \in \mathbb{R}^N$ (i.e., a vectorized version of a 3D image) from the measurements $\mathbf{y} \in \mathbb{R}^M$. Their relation is given by

$$\mathbf{y} = \text{Pois}(\mathbf{H}\mathbf{x} + \mathbf{b}), \quad (1)$$

where $\mathbf{H} : \mathbb{R}^N \rightarrow \mathbb{R}^M$ models the acquisition system (i.e., the microscope) as a 3D low-pass filter and $\mathbf{b} \in \mathbb{R}^M$ denotes the background. The notation Pois models a signal-dependent componentwise Poisson noise.

To recover a 3D image, we optimize the minimization problem

$$\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathcal{C}} \mathcal{L}(\mathbf{H}\mathbf{x} + \mathbf{b}, \mathbf{y}) + \tau \mathcal{R}(\mathbf{x}), \quad (2)$$

¹<https://github.com/ThanhAnPham/pyxudeconv>

where $\mathcal{L} : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$ denotes a data-fidelity term or, from a Bayesian perspective, the negative log-likelihood term. The functional $\mathcal{R} : \mathbb{R}^N \rightarrow \mathbb{R}$ is a regularizer that imposes some prior knowledge on the deconvolved image. The parameter $\tau > 0$ is a tradeoff parameter which balances between the measurements fidelity and the enforcement of the prior knowledge. The set \mathcal{C} represents a set constraint, such as non-negativity.

Here, we consider a negative log-likelihood of Poisson noise and the (differentiable) learned weakly-convex regularizer which was recently proposed by Goujon et al [2]. The minimization problem in (2) reads as

$$\mathbf{x}^* \in \arg \min_{\mathbf{x} \in \mathcal{C}} \sum_{m=0}^M [\mathbf{Hx} - \mathbf{y} \odot \log(\mathbf{Hx} + \mathbf{b})]_m + \tau \mathcal{R}_{(\boldsymbol{\theta}, \sigma)}(\mathbf{x}), \quad (3)$$

where the \odot denotes the Hadamard product, $\log(\cdot)$ is a componentwise natural logarithm, and $\boldsymbol{\theta}$ are the parameters of the regularizer. Here, $\tau, \sigma > 0$ act as parameters that tune the regularization strength (σ relates to the noise level in a Gaussian noise setting, see Section 2.2.2). The set \mathcal{C} is a non-negative set (i.e., $[0, \infty)$).

2.2 Training a 3D Goujon weakly-convex regularizer

2.2.1 Parametric Form

In our method, the functional $\mathcal{R}_{(\boldsymbol{\theta}, \sigma)}(\mathbf{x})$ is a weakly-convex regularizer which was recently proposed by Goujon et al [2]. Here, we concisely describe the approach and invite the interested reader to refer to the original work [1, 2]. The regularizer is defined as

$$\mathcal{R}_{(\boldsymbol{\theta}, \sigma)}(\mathbf{x}) = \sum_{l=1}^{N_C} \alpha_l(\sigma)^{-2} \psi_{\mathbf{c}}(\alpha_l(\sigma) \mathbf{Wx}), \quad (4)$$

where $\mathbf{W} \in \mathbb{R}^{NN_C \times N}$ encodes convolutions with a collection of linear and shift-invariant filters, $\psi_{\mathbf{c}}$ is a potential function with a Lipschitz continuous derivative which is parametrized by $\mathbf{c} \in \mathbb{R}^P$. The parametrization is designed in such a way that $\psi_{\mathbf{c}}$ promotes sparsity. The scaling parameters $(\alpha_l(\sigma))_{l=1}^{N_C}$ are defined as

$$\alpha_l(\sigma) = \frac{e^{s_{\alpha_l}(\sigma)}}{\sigma + \epsilon}, \quad (5)$$

where $s_{\alpha_l}(\sigma)$ is a learnable linear spline with underlying parameters $\mathbf{c}_{\alpha_l} \in \mathbb{R}^S$.

In other words, \mathbf{Wx} is a multi-channel filtered version of the 3D image \mathbf{x} . We will optimize the parameters $\boldsymbol{\theta} = \{(\mathbf{c}_{\alpha_l})_{l=1}^{N_C}, \mathbf{c}, \mathbf{W}\}$ with constraints fully described in [2].

2.2.2 Training procedure

As described in [2], we train the regularizer on a training set where each clean 3D image $\mathbf{x}^{(k)}$ is corrupted as $\mathbf{z}^{(k)} = \mathbf{x}^{(k)} + \sigma \mathbf{n}^{(k)}$ with $\mathbf{n} \sim \mathcal{N}(\mathbf{0}_N, \mathbf{I}_N)$.

$$\boldsymbol{\theta}^* \in \arg \min_{\boldsymbol{\theta}} \|D_{(\boldsymbol{\theta}, \sigma)}(\mathbf{z}^{(k)}) - \mathbf{x}^{(k)}\|_2^2 \quad (6)$$

with

$$D_{(\boldsymbol{\theta}, \sigma)}(\mathbf{z}^{(k)}) = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{x} - \mathbf{z}^{(k)}\|_2^2 + \mathcal{R}_{(\boldsymbol{\theta}, \sigma)}(\mathbf{x}). \quad (7)$$

More details are available in [2].

2.2.3 Generation of a 3D training set

To reproduce features observed in microscopy, we simulate microtubules-alike 3D images with a generative procedure which mimics fluorophores. Each 3D image includes a random number of tubes with diverse inner and outer radii. For each tube, continuously-defined positions are randomly sampled within the inner and outer radii. Each position (i.e., shifted Dirac) is then (continuously) convolved with a Gaussian window function with fixed width, and we sum all the convolved positions to obtain a continuous simulated 3D image. Finally, we sample the 3D image at a desired resolution (e.g., anisotropic resolution) to generate a 3D discrete image made of pseudo-microtubules.

We generate 1000 such samples with varying resolutions. To match microscopy data, we only simulate anisotropic resolutions where the axial is worse than lateral resolution.

2.3 Reconstruction Algorithm: Goujon Accelerated Richardson-Lucy (GARL)

To optimize (3), we deploy an accelerated projected gradient descent. For the sake of simplicity, we first describe its non-accelerated version, which is an iterative algorithm whose steps are, for $k = 0, \dots, K_{\max} - 1$,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \boldsymbol{\gamma}^{(k)} \odot \left(\mathbf{H}^T \mathbf{1}_M - \mathbf{H}^T (\mathbf{y} \oslash (\mathbf{Hx}^{(k)} + \mathbf{b})) + \tau \nabla_{\mathbf{x}} \mathcal{R}_{(\boldsymbol{\theta}, \sigma)}(\mathbf{x}^{(k)}) \right). \quad (8)$$

The vector $\boldsymbol{\gamma}^{(k)} \in \mathbb{R}^N$ is the componentwise step size, $\mathbf{1}_M \in \mathbb{R}^M$ is a vector whose entries are ones, and \oslash denotes the pointwise division. The operator \mathcal{P}_C denotes a projection on the set \mathcal{C} . To prevent division by 0, the denominator $\mathbf{Hx}^{(k)}$ in the pointwise division is clipped by $[\epsilon, \inf]$ with $\epsilon = 10^{-6}$. In the current implementation, the background is estimated from the measurements, but other schemes are possible.

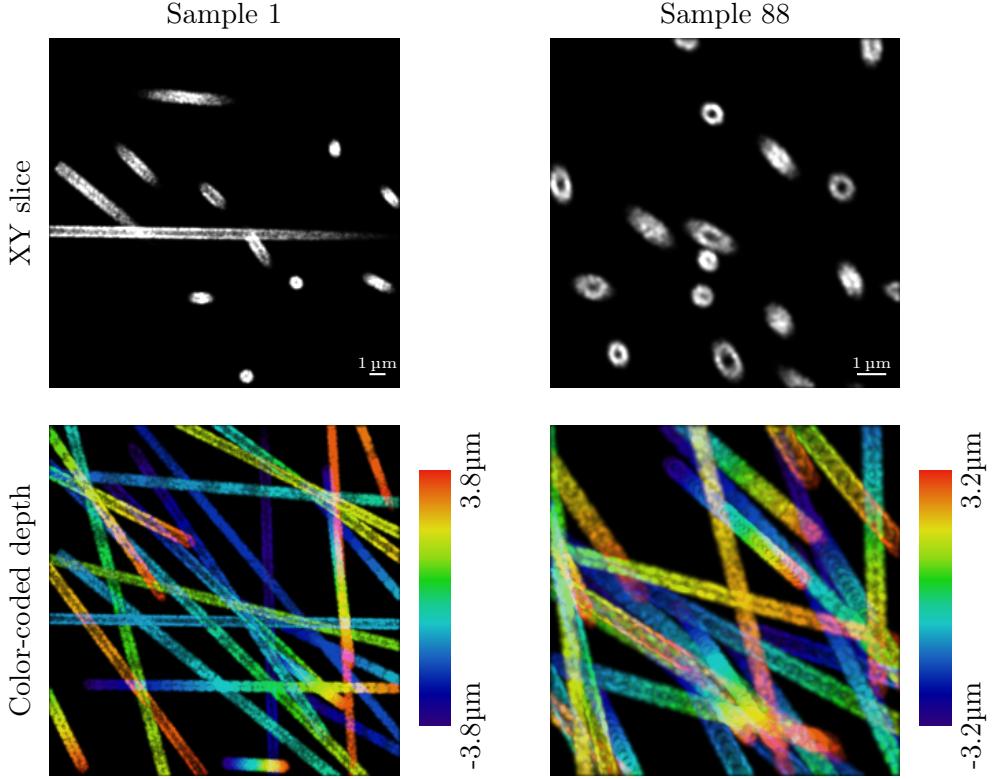


Figure 1: 3D simulated microtubules samples. Top row: XY slice at $z = -1.3\mu\text{m}$ (left) and $z = -1.1\mu\text{m}$ (right). Bottom row: color-coded depth map.

We set the step size $\gamma^{(k)} = \mathbf{x}^{(k)} \oslash (\mathbf{H}^T \mathbf{1}_M)$, similarly to the Richardson-Lucy algorithm (RL). Then, we obtain a multiplicative update rule

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} \odot \left(\mathbf{H}^T (\mathbf{y} \oslash (\mathbf{H}\mathbf{x}^{(k)} + \mathbf{b})) - \tau \nabla_{\mathbf{x}} \mathcal{R}_{(\theta, \sigma)}(\mathbf{x}^{(k)}) \right) \oslash (\mathbf{H}^T \mathbf{1}_M). \quad (9)$$

The accelerated version is fully described in Algorithm 1. We also display a convergence plot for the accelerated (GARL) versus non-accelerated (GRL) versions in Figure 2.

Algorithm 1 Goujon Accelerated Richardson-Lucy (GARL)

Require: $\mathbf{y}, \mathbf{b} \in \mathbb{R}^M$, $\mathbf{x}^0 \in \mathbb{R}^N$, $K_{\max} > 0$, $t_0 = 1$, $k = 0$

```

1: while  $k < K_{\max}$  or convergence do
2:    $t^{(k+1)} \leftarrow \frac{1+\sqrt{1+4(t^{(k)})^2}}{2}$ 
3:    $\mathbf{p}^{(k)} \leftarrow \mathbf{x}^{(k)} + \frac{t^{(k)}-1}{t^{(k+1)}}(\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)})$ 
4:    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{p}^{(k)} \odot \left( \mathbf{H}^T (\mathbf{y} \oslash (\mathbf{H}\mathbf{p}^{(k)} + \mathbf{b})) - \tau \nabla_{\mathbf{x}} \mathcal{R}_{(\theta, \sigma)}(\mathbf{p}^{(k)}) \right) \oslash (\mathbf{H}^T \mathbf{1}_M)$ 
5:    $k \leftarrow k + 1$ 
6: end while

```

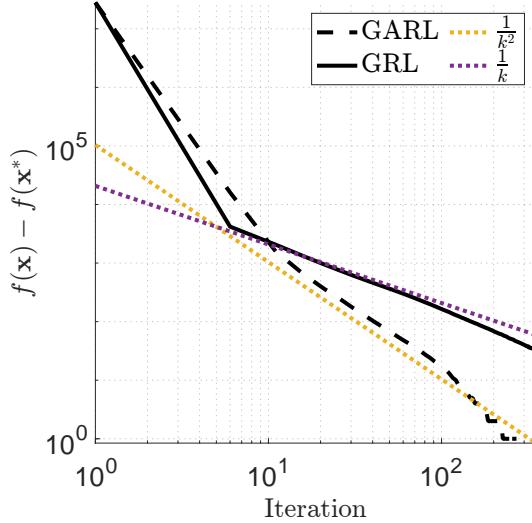


Figure 2: Convergence plot for GARL. The objective function f is described in (3). The Nesterov acceleration provides a convergence rate of $O(\frac{1}{k^2})$, in contrast to the RL convergence rate of $O(\frac{1}{k})$.

2.4 Noise amplification in Richardson-Lucy deconvolution

A well-known drawback of RL deconvolution is that the noise is increasingly amplified over the iterations. The set constraint is insufficient to prevent noise amplification, as shown in Figure 3. To mitigate the amplification, early stopping is a popular regularization scheme for RL, but, in some cases, more iterations are required to properly deconvolve the image (Figure 3, iterate 50). In addition, early stopping requires an appropriate criterion.

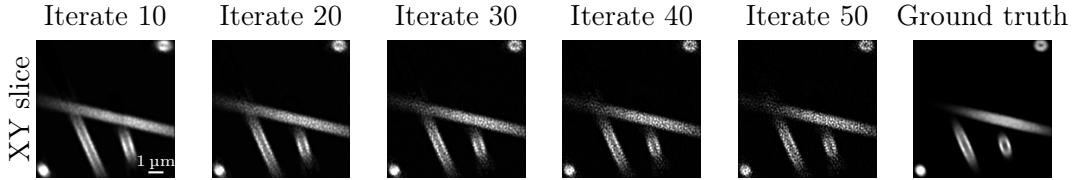


Figure 3: Deconvolved images with the Richardson-Lucy algorithm over the iterations. The noise is increasingly amplified over the iterations.

An alternative to early stopping is the addition of some prior knowledge about the sample. Here, we add a differentiable term in the minimized cost, although other approaches are possible [3].

We will thus assess the classical Richardson-Lucy (RL), a total-variation RL (e.g., RLT), and the proposed learned regularizer (GARL). For RLT, we use a smoothed TV [4, 5], which allows us to deploy the same algorithm as our proposed method. All the methods have the same set constraint.

3 Results

3.1 Learned parameters

The Goujon regularizer’s shallow architecture enables us to gain insights into the enforced priors. We can visualize the features that are favored in the reconstructed volumes. In Figure 4, we display examples of filters learned from the simulated microtubule dataset. We can see that several kinds of finite differences are learned as well as filters that may promote structure on a tube section.

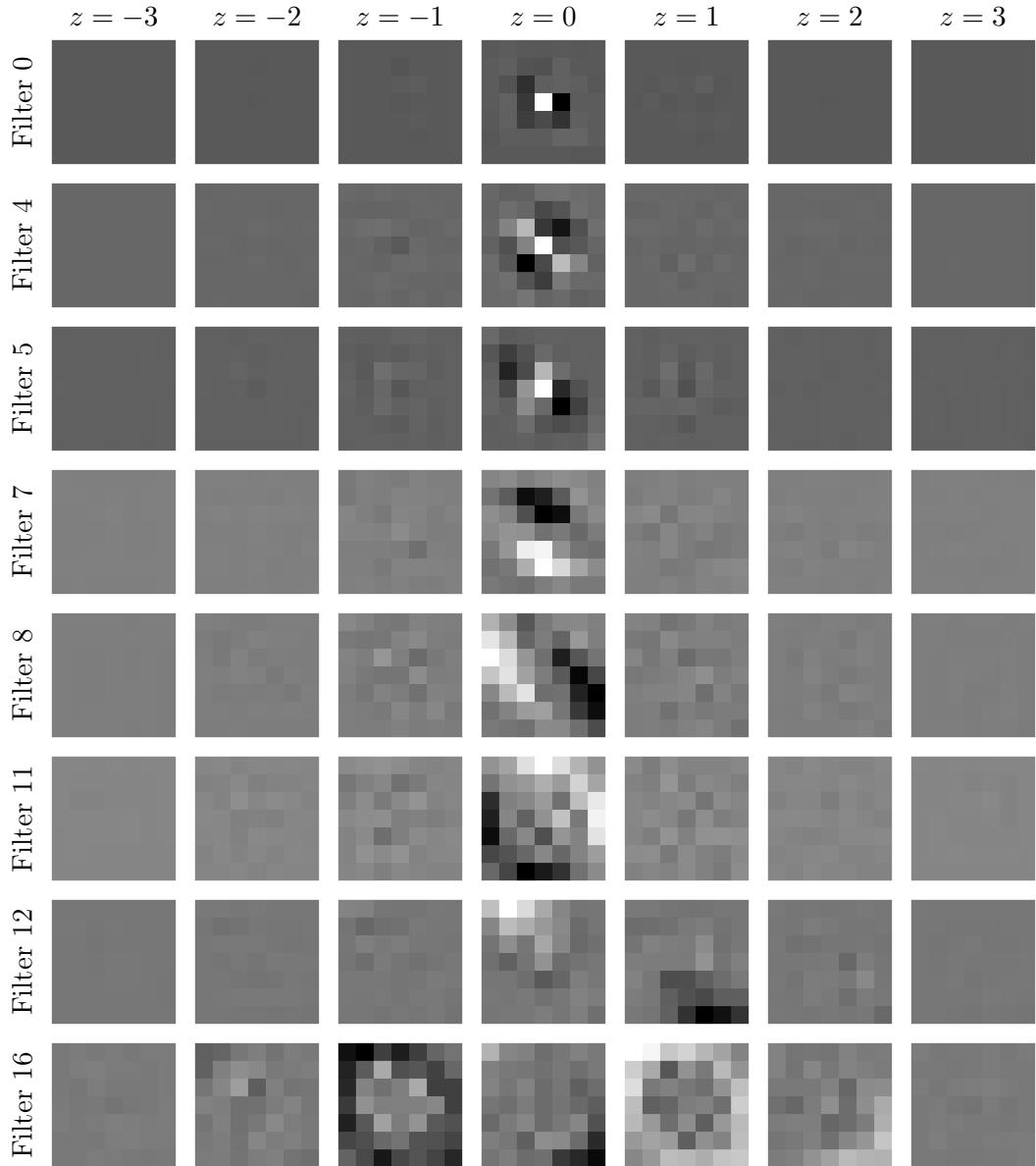


Figure 4: Examples of 3D impulse response learned by the Goujon regularizer.

The potential function is learned as well. In practice, we optimize the potential-function derivative (Figure 5, right), which is represented by three parameters. In Figure 5, we see that the potential function (left) is non-convex and is typically enforcing sparsity.

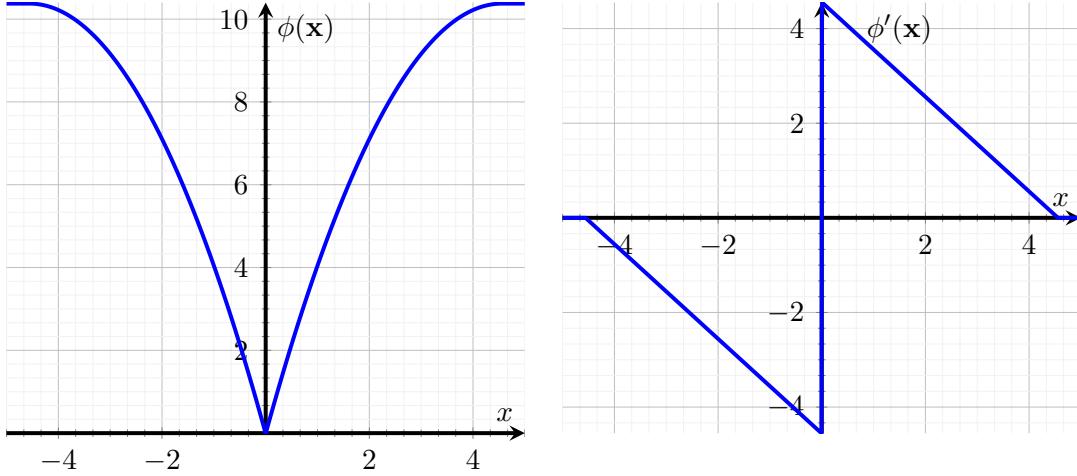


Figure 5: Potential function and its derivative learned on the microtubules dataset.

3.2 Deconvolution of microtubules and calibration samples

We display in Figure 6 deconvolved 3D images of a sample similar to the training set. We can see that while more RL iterations enhance the deconvolution, they simultaneously amplify the noise. In the axial slice (ROI 4), the effect is quite visible. By contrast, GARN and TV allow more iterations to deconvolve properly the blurred measurements by injecting prior knowledge. Here, the piecewise constant prior promoted by TV seems unsuitable for the considered sample and staircase effects are quite prominent (e.g., ROI 1, 4). Our observations are corroborated by the computed regressed signal-to-noise ratio (RSNR) over the whole 3D images (Table 1).

Furthermore, we deconvolved 3D images of a calibration samples with two different PSFs (Figure 7 and Figure 8 for a narrower PSF), where similar effects are observable. For the narrower PSF, RL performs relatively well as complete deconvolution could be performed before noise amplification except on larger structure (ROI 3, 5, 7).

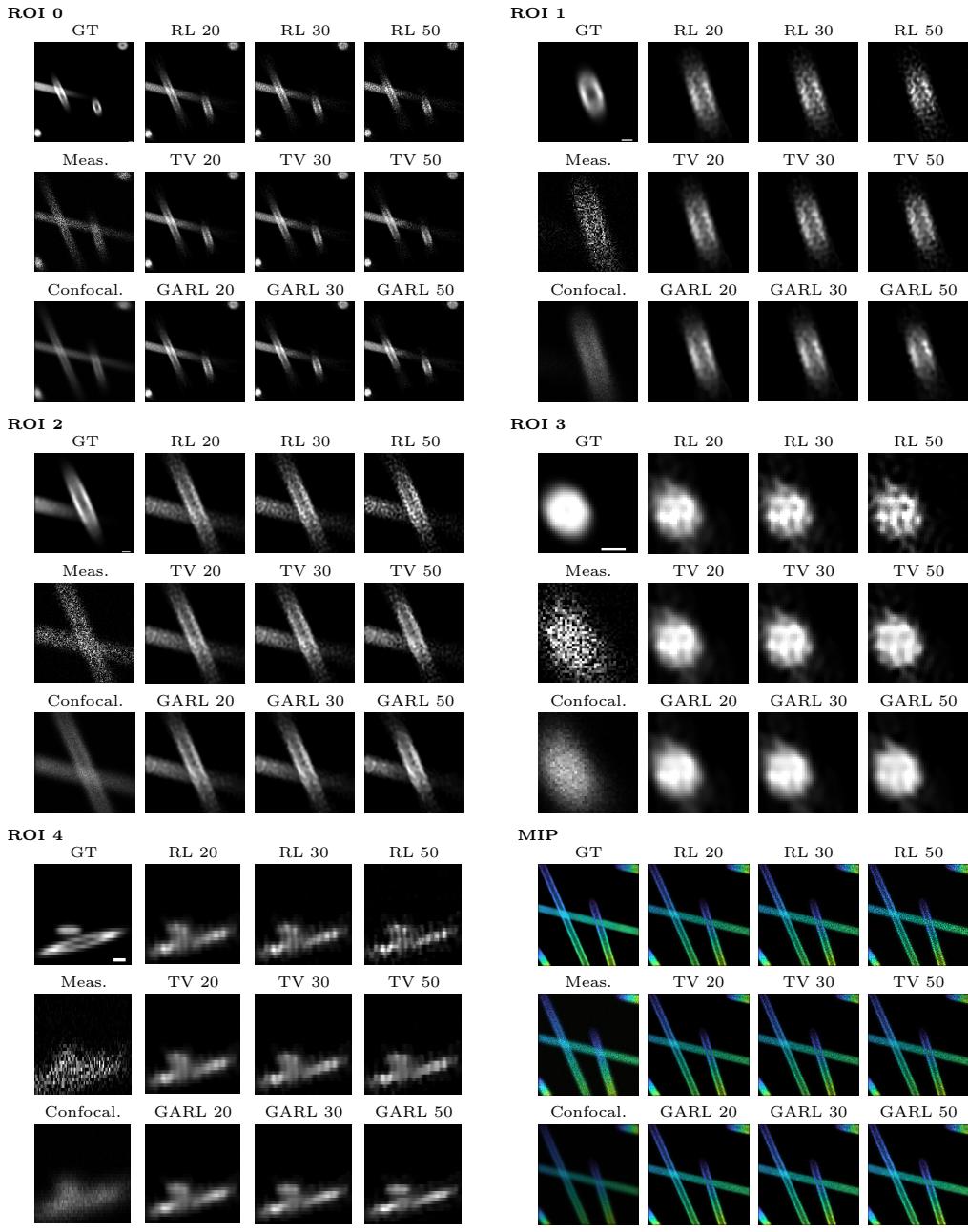


Figure 6: 3D images of simulated microtubules. The measurements (Meas.) are different planes from the 16th view / PSF (out of 32 views). Confocal images were computed by summing all the measurements. Scale bar: 357nm.

Table 1: Regressed signal-to-noise ratio (RSNR) on the 3D images for the reconstructions displayed in Figure 6. RL: Richardson-Lucy, TV: Total variation, GARL: Goujon accelerated Richardson-Lucy.

Methods	RSNR [dB]		
	Iterations 20	Iterations 30	Iterations 50
RL	10.24	10.06	8.13
TV	10.02	10.1	9.87
GARL	10.68	11.55	12.31

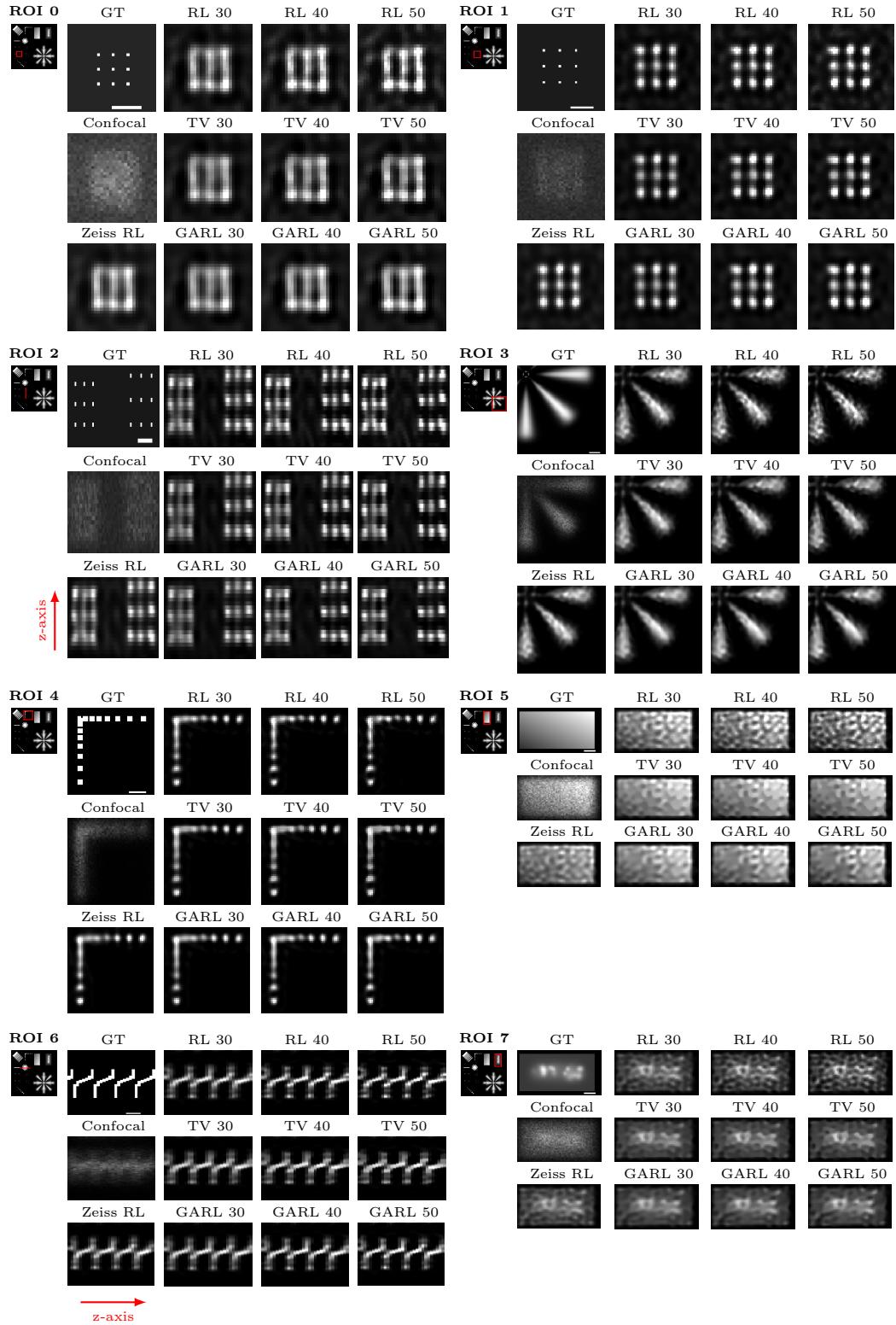


Figure 7: 3D images of calibration sample (low SNR setting). Confocal images were computed by summing all the measurements. Scale bar: 357nm. The number of iterations is indicated for each subplot if relevant. All methods are based on the RL algorithm. GT: ground-truth.

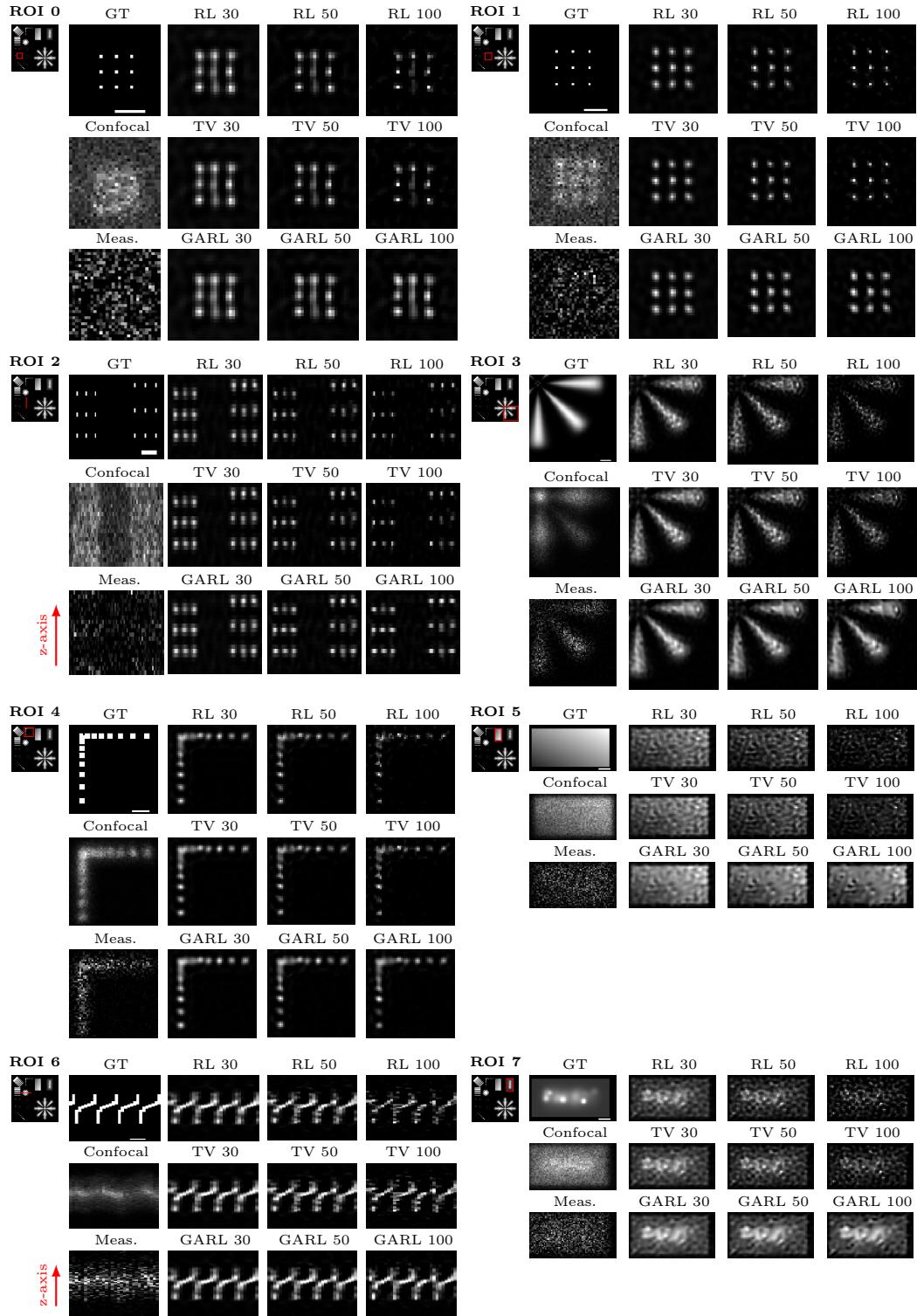


Figure 8: 3D images of calibration sample (low SNR setting and narrow PSF). The measurements (Meas.) correspond to planes from the 16th view (out of 32 views). Confocal images were computed by summing all the measurements. Scale bar: 357nm. The number of iterations is indicated for each subplot if relevant. All methods are based on the RL algorithm. GT: ground-truth.