

# Ring artifact and Poisson noise attenuation via volumetric multiscale nonlocal collaborative filtering of spatially correlated noise

Ymir Mäkinen,<sup>a\*</sup> Stefano Marchesini<sup>b</sup> and Alessandro Foi<sup>a</sup>

<sup>a</sup>Tampere University, Finland, and <sup>b</sup>SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA. \*Correspondence e-mail: ymir.makinen@tuni.fi

Received 31 August 2021  
Accepted 10 March 2022

Edited by M. Yamamoto, RIKEN SPring-8 Center, Japan

**Keywords:** BM4D; noise; image processing; tomography; artifact.

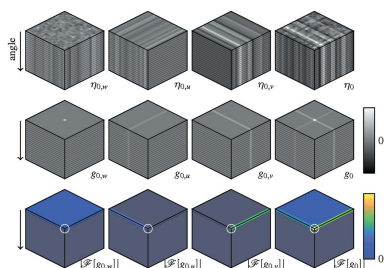
X-ray micro-tomography systems often suffer from high levels of noise. In particular, severe ring artifacts are common in reconstructed images, caused by defects in the detector, calibration errors, and fluctuations producing streak noise in the raw sinogram data. Furthermore, the projections commonly contain high levels of Poissonian noise arising from the photon-counting detector. This work presents a 3-D multiscale framework for streak attenuation through a purposely designed collaborative filtering of correlated noise in volumetric data. A distinct multiscale denoising step for attenuation of the Poissonian noise is further proposed. By utilizing the volumetric structure of the projection data, the proposed fully automatic procedure offers improved feature preservation compared with 2-D denoising and avoids artifacts which arise from individual filtering of sinograms.

## 1. Introduction

Computed tomography is commonly affected by streak noise in measured raw sinogram data (Jha *et al.*, 2013; Artul, 2013; Boas & Fleischmann, 2012), which can be caused by miscalibration of detector linear response, beam fluctuations, beam hardening, or dusty or damaged scintillator screens (Haibel, 2008; Vidal *et al.*, 2005; Anas *et al.*, 2010). Streak noise in projections causes ring artifacts in reconstructed volumes, which present as centered circles or half-circles (Croton *et al.*, 2019). As the sinogram data are obtained through a photon-counting detector, the statistics of the measured raw data can be further modeled through a Poisson distribution, which may result in high levels of Poissonian noise, commonly attenuated within the reconstruction process through iterative approaches (Mohan *et al.*, 2014; Venkatakrishnan *et al.*, 2013).

Although ring artifacts can be reduced by scanning protocols (Pelt & Parkinson, 2018), high-quality scintillator screens and detectors, it is difficult to completely avoid them and therefore achieve highest-quality reconstruction solely by experimental measures, requiring algorithmic processing of the acquisitions.

Popular methods to reduce ring artifacts include wavelet-FFT filters (Münch *et al.*, 2009), combinations of polynomial smoothing filters and careful calibration of the detector response function (Vo *et al.*, 2018; Croton *et al.*, 2019), smoothing filters with segmentation in the tomogram domain (Massimi *et al.*, 2018), ring removal in the tomogram domain upon polar coordinates transformation (Sijbers & Postnov, 2004; Li *et al.*, 2021), and iterative algorithms (Paleo & Mirone, 2015) that combine regularized reconstruction with denoising. Recently, in Mäkinen *et al.* (2021), we proposed



effective ring artifact attenuation through sinogram-domain collaborative filtering, presenting a multiscale architecture with a Block-matching and 3-D filtering (BM3D) image denoiser for correlated noise (Dabov *et al.*, 2008; Mäkinen *et al.*, 2020) at the core of the process. To the best of our knowledge, Mäkinen *et al.* (2021) offers state-of-the-art results in ring attenuation. In particular, it does not cause new artifacts around strong signal features, common to other popular ring removal algorithms. However, being based on a filter for 2-D data<sup>1</sup>, applied to individual sinograms, it may cause discontinuities across the third dimension.

In this work, we address both streak reduction and Poissonian noise removal from volumetric stacks of projections. The contribution of this work is threefold: (1) We propose a multiscale streak denoising framework for denoising of volumetric data. In particular, this framework can be seen as an extension of Mäkinen *et al.* (2021) for filtering of 3-D volumes. (2) After streak noise removal, and before reconstruction, we embed a distinct multiscale denoising step to attenuate the Poissonian noise component of the projections. This allows to apply the reconstruction process using milder regularization and improve the tradeoff between noise reduction and artifact suppression. (3) As a general-purpose algorithmic contribution, the filter used at the core of the multiscale denoising process is an improved version of the BM4D (Maggioni *et al.*, 2012) volumetric denoising algorithm. The included enhancements, discussed in Appendix A, allow the long-range noise correlation which characterizes the streaks to be dealt with.

The proposed filtering procedure for both streaks and Poissonian noise is fully automatic and includes self-calibration of the filtering strength. We demonstrate the denoising performance of the proposed approach on real data from the table-top Prisma XRM microCT at Sigray, and from the synchrotron-based microCT at the Advanced Photon Source (APS) in Argonne, available through Tomobank (De Carlo *et al.*, 2018).

## 2. Bright-field normalization

The following normalization of the raw projections and the streak model upon a logarithmic transformation follow that of Mäkinen *et al.* (2021).

The optical attenuation through the sample is determined experimentally via bright-field corrections through two separate acquisitions, the bright-field and the dark-field (Seibert *et al.*, 1998). The bright-field is obtained by the imaging procedure with no sample, and the dark-field is obtained with no beam; both are 2-D arrays the size of effective pixels of the detector. The Beer–Lambert law further relates the X-ray transform through the sample to the optical attenuation by a logarithmic transformation (Swinehart, 1962).

Hence, the raw projections  $P_{\text{raw}}$  are first normalized as

$$P_{\text{norm}} = \frac{P_{\text{raw}} - I_{\text{D}}}{I_{\text{B}} - I_{\text{D}}}, \quad (1)$$

where  $I_{\text{D}}$  is the dark-field and  $I_{\text{B}}$  is the bright-field<sup>2</sup>, and then log-transformed as

$$Z = \ln(P_{\text{norm}}). \quad (2)$$

### 2.1. Noise model for normalized projections

Apart from possible completely defective detectors<sup>3</sup> we treat the variation in detector response as normally distributed; as such, the streak noise will follow a normal distribution. Furthermore, we model the streak noise as locally stationary, meaning that the variance is presumed constant within the support of the denoising filter. Note that this does not mean that the noise is i.i.d. or white, as it is instead characterized by very long range correlation presenting as streaks.

As the data are obtained through a photon-counting detector, the statistics of the measured raw data can be further modeled through a Poisson distribution with nonstationary parameters after the bright-fielding.

Given the premises of normally distributed streak noise and Poissonian noise, noise in projections normalized by (1) can be formally written as

$$P_{\text{norm}} = A(1 + \eta_p) + \pi = A + A\eta_p + \pi, \quad (3)$$

where  $A$  are the underlying noise-free projections,  $\eta_p$  is the normally distributed streak noise component, and  $\pi$  is Poissonian noise which we model as white and zero-mean; all components of (3) are considered as 3-D arrays and multiplications are elementwise.

The natural logarithm of (2) acts as a variance-stabilizing transformation (VST) for the multiplicative noise component  $(1 + \eta_p)$ . Hence, we have

$$Z = \ln(P_{\text{norm}}) \approx \ln\left(A + \frac{\pi}{1 + \eta_p}\right) + \eta_p, \quad (4)$$

where the approximation comes from  $\ln(1 + \eta_p) \approx \eta_p$ .

### 3. Correlated noise

The denoising is conducted in two steps. First, we aim to estimate the ‘streak-free’ projections

$$Y = \ln\left(A + \frac{\pi}{1 + \eta_p}\right), \quad (5)$$

which are corrupted by white Poissonian noise. Then, as a separate denoising step, we consider the attenuation of the remaining noise originating from  $\pi$ .

<sup>2</sup> As  $P_{\text{raw}}$  is a 3-D array, the pixels of  $I_{\text{B}}$  and  $I_{\text{D}}$  are replicated through the angle dimension for the operations in (1).

<sup>3</sup> Extreme streak noise arising from defective detectors is addressed separately in Section 4.2.

Throughout this work, we will represent the volume to be filtered according to the correlated noise model presented in the following subsection. This model will assume different meaning at different parts of the algorithm. First, applying locally to the streaks as a type of long-range correlated noise; second, to the noise arising from the Poissonian component  $\pi$ .

### 3.1. Correlated noise model

We consider the noisy input  $z: X \rightarrow \mathbb{R}$  to be a combination of underlying data  $y$  and additive stationary spatially correlated noise  $\eta$  to be filtered,

$$z(x) = y(x) + \eta(x), \quad x \in X, \quad (6)$$

where  $x \in X \subset \mathbb{Z}^3$  is the coordinate in the finite three-dimensional volumetric domain  $X$  and

$$\eta = v \circledast g, \quad v(\cdot) \sim \mathcal{N}(0, 1), \quad (7)$$

$v$  being zero-mean i.i.d. Gaussian noise with unit variance, and  $\circledast$  denoting 3-D convolution with the kernel  $g$ . The kernel  $g$  defines the spatial correlation of the noise as well as the noise strength, with  $\|g\|_2 = \text{std}\{\eta\}$ . An equivalent way of representing correlated noise is by its power spectral density (PSD)  $\Psi$ ,

$$\Psi = \mathbb{E}\left\{|\mathcal{F}[\eta]|^2\right\} = \text{var}\{\mathcal{F}[\eta]\} = |X| |\mathcal{F}[g]|^2, \quad (8)$$

with  $\mathcal{F}$  being the 3-D Fourier transform, and  $|X|$  denoting the cardinality (*i.e.* number of elements) of  $X$ . Equivalently, a kernel  $g$  satisfying (7)–(8) can be defined from  $\Psi$  as

$$g = |X|^{-1/2} \mathcal{F}^{-1}[\text{std}\{\mathcal{F}[\eta]\}] = |X|^{-1/2} \mathcal{F}^{-1}[\sqrt{\Psi}]. \quad (9)$$

### 3.2. Estimation of noise standard deviation

When applying the above model to noisy data, it is essential to have knowledge of either the kernel  $g$  or, equivalently, the PSD  $\Psi$ , as they fully characterize the noise. Assuming  $g$  in (7) is known modulo a scaling factor  $\zeta$  from a known kernel  $g_s$ , the noise estimation can be simplified to estimating  $\zeta$ . In particular, in order to model the streak and Poissonian noise components arising from the particular composition of noise given in (3), the kernels  $g_s$  should induce either very long range correlation or near white noise across each dimension  $d$ . The estimation procedure can be built as a direct extension of the one adopted by Mäkinen *et al.* (2021) to 3-D. To reduce the signal-to-noise ratio (SNR) to acquire a better noise estimate, we convolve  $z$  with a 3-D anisotropic kernel  $g_d$  that provides either low-pass or high-pass filtering along different dimensions;  $g_d$  is designed based on the noise statistics so that it preserves the noise component of interest while attenuating signal contrast. Specific instances are given in Section 4.1.1 and Section 5.3. One can then compute an estimate of the standard deviation of  $\eta \circledast g_d$  via its median absolute deviation (Hampel, 1974),

$$\text{std}\{\widehat{\eta \circledast g_d}\} = 1.4826 \text{ smed}_X \left[ |z \circledast g_d - \text{smed}_X(z \circledast g_d)| \right], \quad (10)$$

where  $\text{smed}$  denotes the sample median and the factor 1.4826 calibrates the estimate with respect to a normal distribution of the noise. As  $\text{std}\{\eta \circledast g_d\} = \|\zeta g_s \circledast g_d\|_2$ , an estimate  $\hat{\zeta}$  of  $\zeta$  can be obtained through

$$\hat{\zeta} = \|g_s \circledast g_d\|_2^{-1} \text{std}\{\widehat{\eta \circledast g_d}\}. \quad (11)$$

## 4. Multiscale streak filtering

In the following, we treat the first dimension of the stack of projections as the angular dimension, and the second and third as the horizontal and vertical displacement dimensions.

Because the streaks are inherently low-frequency with respect to the angle, they are filtered entirely at a *coarse angular scale*; for this task, we extend the multiscale procedure of Mäkinen *et al.* (2021). The main changes in the proposed procedure arise from replacing the one-dimensional binning operations along the displacement dimension with corresponding 2-D binning operators  $\mathcal{B}_{2D}$  and  $\mathcal{B}_{2D}^{-1}$  executed across both displacement dimensions. Furthermore, instead of using a direct 3-D extension of the 2-D streak PSD, we adjust the streak model to account for possible long correlation also along the displacement dimensions.

In detail, the multiscale streak attenuation procedure proceeds as follows. We begin by an angular binning  $\mathcal{B}_\alpha$ . The result of the angular binning  $Z_0 = \mathcal{B}_\alpha(Z)$  is binned  $K$  times through  $\mathcal{B}_{2D}$  to obtain  $Z_K = \mathcal{B}_{2D}^K(Z_0)$ . The size of each binned volume is a quarter of the input size. Then, we process each scale in a coarse-to-fine fashion, where progressively for each  $k = K, K-1, \dots, 2, 1, 0$ , we obtain an estimate  $\hat{Y}_k$  of  $\mathcal{B}_{2D}^k(Y_0) = \mathcal{B}_{2D}^k[\mathcal{B}_\alpha(Y)]$ . We start by taking as noisy input  $Z_K^*$  of BM4D the smallest binned volume  $Z_K$ ; in this way, we obtain from  $Z_K^* = Z_K$  the coarsest estimate  $\hat{Y}_K$ , which is taken as initialization for the following recursive steps executed for each scale  $k = K-1, \dots, 0$ :

(1) Replace the coarser-scale components of  $Z_k$  by those of the estimate  $\hat{Y}_{k+1}$ :

$$\begin{aligned} Z_k^* &= Z_k - \mathcal{B}_{2D}^{-1}[\mathcal{B}_{2D}(Z_k)] + \mathcal{B}_{2D}^{-1}(\hat{Y}_{k+1}) \\ &= Z_k - \mathcal{B}_{2D}^{-1}(Z_{k+1} - \hat{Y}_{k+1}). \end{aligned}$$

(2) Denoise  $Z_k^*$  with BM4D to produce the estimate  $\hat{Y}_k$ .

Finally, we replace the coarse angular components of the full-size stack  $Z$  with those from the finest scale estimate  $\hat{Y}_0$ ,

$$\hat{Y} = Z - \mathcal{B}_\alpha^{-1}[\mathcal{B}_\alpha(Z)] + \mathcal{B}_\alpha^{-1}(\hat{Y}_0). \quad (12)$$

### 4.1. Multiscale noise model

For BM4D denoising, we regard  $Z_k^*$  of each scale  $k$  as  $z$  of the model (6), as

$$Z_k^* = \mathcal{B}_{2D}^k(Y_0) + \eta_k^*, \quad (13)$$

where

$$\eta_k^* = \begin{cases} \eta_K, & k = K, \\ \eta_k - \mathcal{B}_{2D}^{-1}[\mathcal{B}_{2D}(\eta_k)], & k < K, \end{cases} \quad (14)$$

and  $\eta_k = \mathcal{B}_{2D}^k(\eta_0) = \mathcal{B}_{2D}^k[\mathcal{B}_\alpha(\eta_p)]$ .

This definition for  $\eta_k^*$ ,  $k < K$ , follows from considering the coarser-scale estimate  $\hat{Y}_{k+1}$  as perfectly denoised. Similar to (8),  $\eta_k^*$  is treated as correlated noise with PSD,

$$\Psi_k^* = \text{var}\{\mathcal{F}[\eta_k^*]\} = |X_k| |\mathcal{F}[g_k^*]|^2, \quad K \geq k \geq 0, \quad (15)$$

where  $g_k^*$  is a correlation kernel and  $|X_k|$  is the pixel size of  $Z_k$ . As per (9), the kernel  $g_k^*$  can be defined as

$$g_k^* = |X_k|^{-1/2} \mathcal{F}^{-1}[\text{std}\{\mathcal{F}[\eta_k^*]\}]. \quad (16)$$

**4.1.1. Adaptive parametric model of  $\Psi_k^*$ .** We note that, in addition to the approximately white streak noise, the sinograms may contain streaks with very long range correlation across the displacement dimensions. As this correlation is aligned along the detector axis, it is not clearly observable in individual sinograms, but may create significant noise structure in the full volume. Hence, we approximate the streak noise  $\eta_0$  through three angularly constant streak noise components distinct in the displacement,

$$\eta_0 = \eta_{0,w} + \eta_{0,u} + \eta_{0,v}, \quad (17)$$

where  $\eta_{0,w}$  is streak noise white across both displacement dimensions,  $\eta_{0,u}$  is streak noise constant across horizontal displacement, and  $\eta_{0,v}$  is streak noise constant across vertical displacement.

Let us denote by  $\eta_{0,p} \in \{\eta_{0,w}, \eta_{0,u}, \eta_{0,v}\}$  a noise component of  $\eta_0$ . For each  $\eta_{0,p}$ , we can define a respective scaled correlation kernel

$$\varsigma_{0,p} g_{0,p} = |X|^{-1/2} \mathcal{F}^{-1}[\text{std}\{\mathcal{F}[\eta_{0,p}]\}], \quad (18)$$

where  $\varsigma_{0,p} = \text{std}\{\eta_{0,p}\}$ , and  $\|g_{0,p}\|_2 = 1$ . Example realizations, kernels, and PSDs for each of these components as well as  $\eta_k$  are shown in Fig. 1 (top).

We note that each  $\eta_{k,p} = \mathcal{B}_{2D}^k(\eta_{0,p})$  is characterized by a kernel  $\varsigma_{k,p} g_{k,p} = 2^k \varsigma_{0,p} g_{0,p}$ . In particular, this property arises from the noise structure of the corresponding components: as  $\mathcal{B}_{2D}$  operates through summation and elimination of adjacent pixels, the operation preserves both noise whiteness and constant noise. The ratio  $2^k$  follows from the summation along two dimensions, meaning that the variance of the coarser scale is four times that of the finer scale.

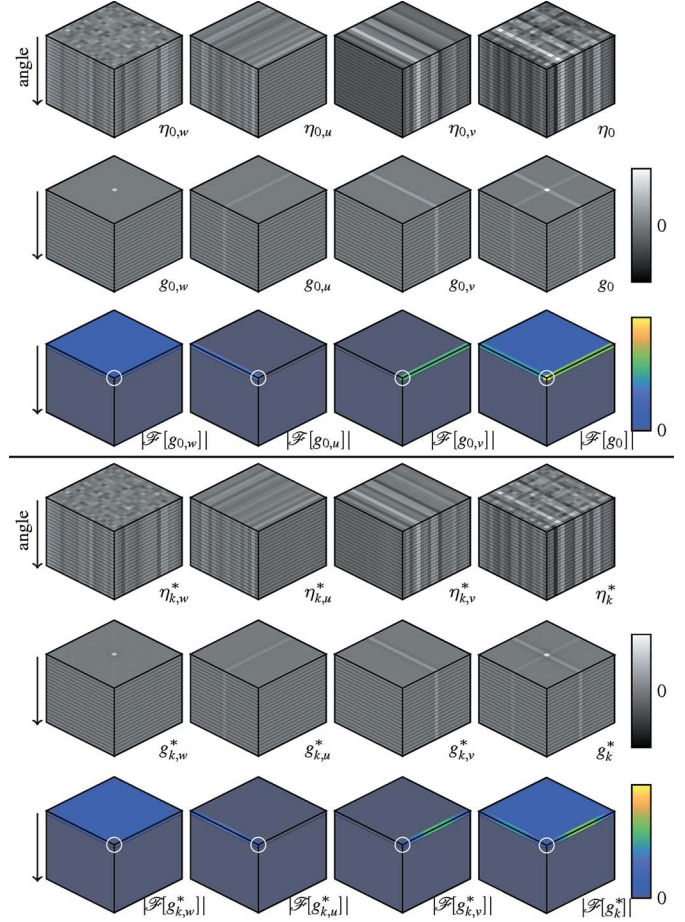
Disregarding the specific support size of their actual finite realizations, we can identify the stationary random fields as

$$\eta_{k,p} = 2^k \text{std}\{\eta_{0,p}\} \eta_{G,p} \quad (19)$$

where  $\eta_{G,p}$  is noise characterized by  $g_{k,p}$ , and hence  $\text{var}\{\eta_{G,p}\} = 1$ . We can then express the residuals of any of the components  $\eta_{0,p}$  as

$$\eta_{k,p}^* = \begin{cases} 2^k \text{std}\{\eta_{0,p}\} \eta_{G,p}, & k = K, \\ 2^k \text{std}\{\eta_{0,p}\} \{\eta_{G,p} - \mathcal{B}_{2D}^{-1}[\mathcal{B}_{2D}(\eta_{G,p})]\}, & k < K. \end{cases} \quad (20)$$

Then,



**Figure 1**

Top: example noise  $\eta_{0,p}$ , the corresponding kernels  $g_{0,p}$  and the root PSD  $|\mathcal{F}[g_{0,p}]|$  for each noise component in (17) with  $\varsigma_{0,w} = 6$ ,  $\varsigma_{0,u} = 5$ , and  $\varsigma_{0,v} = 8$ , as well as example noise, kernel, and root PSD corresponding to the compound noise  $\eta_k$ . Bottom: example noise, the corresponding kernels, and root PSD of the corresponding binning residuals  $\eta_k^*$ . For all visualizations, the angular dimension of the data is the vertical dimension in the figure. The DC corner of the Fourier spectra is marked by a circle. Note that all root PSDs are nonzero only on the angular DC plane, and the kernels and the noise consist of repeated planes across the angle.

$$\eta_k^* = \eta_{k,w}^* + \eta_{k,v}^* + \eta_{k,u}^*, \quad (21)$$

where the noise correlation kernel corresponding to a component  $\eta_{k,p}^* \in \{\eta_{k,w}^*, \eta_{k,v}^*, \eta_{k,u}^*\}$  can be written with  $\varsigma_{k,p} = 2^k \varsigma_{0,p}$  as

$$g_{k,p}^* = \begin{cases} \varsigma_{k,p} g_{k,p}, & k = K, \\ \varsigma_{k,p} g_{\mathcal{B}_{2D}} \otimes g_{k,p}, & k < K, \end{cases} \quad (22)$$

where  $g_{\mathcal{B}_{2D}}$  is a 2-D kernel across the displacement dimensions characterizing the residual from 2-D binning of white noise. Specifically,

$$g_{\mathcal{B}_{2D}} = |X_G|^{-1/2} \mathcal{F}^{-1}[\text{std}\{\mathcal{F}(\eta_{G,2D} - \mathcal{B}_{2D}^{-1}[\mathcal{B}_{2D}(\eta_{G,2D})])\}],$$

where  $\eta_{G,2D}$  is a two-dimensional white random field. The field size  $|X_G|$  is included only for the normalization of the Fourier transform, and the formula holds for an arbitrary size.

Then, the PSD of  $\eta_k^*$ ,  $K \geq k \geq 0$ , can be written as

$$\begin{aligned} \Psi_k^* &= \text{var}\{\mathcal{F}[\eta_{k,w}^*]\} + \text{var}\{\mathcal{F}[\eta_{k,u}^*]\} + \text{var}\{\mathcal{F}[\eta_{k,v}^*]\} \\ &= |X_k| \left( |\mathcal{F}[g_{k,w}^*]|^2 + |\mathcal{F}[g_{k,u}^*]|^2 + |\mathcal{F}[g_{k,v}^*]|^2 \right). \end{aligned} \quad (23)$$

As any  $\eta_k^*$  is constant in angle,  $\Psi_k^*$  is non-zero only across the DC plane with respect to the angular dimension. Example realizations, kernels, and PSDs for the residual components are shown in Fig. 1 (bottom).

Although (23) allows for modeling of very long range correlation, the streak noise is likely to contain minor correlation along the displacement not accounted for by this model. To adapt to such deviations, we allow the scaling parameters  $\varsigma_{k,p} \geq 0$  for each noise component to vary with each scale  $k$  by estimating them individually at each scale, effectively accounting for mild local correlation in the noise.

*Estimation of  $\varsigma_{k,w}$ ,  $\varsigma_{k,u}$ , and  $\varsigma_{k,v}$ .* Based on (22) and (23), the PSD is completely determined by the values assumed by the three parameters  $\varsigma_{k,w}$ ,  $\varsigma_{k,u}$ , and  $\varsigma_{k,v}$  and the known kernels  $g_{k,p}$  and  $g_{\mathcal{B}_{2D}} \otimes g_{k,p}$ . To adaptively obtain the parameters, we begin by obtaining three noise variance estimates  $\hat{\sigma}_{k,w}^2$ ,  $\hat{\sigma}_{k,u}^2$ , and  $\hat{\sigma}_{k,v}^2$ . For each estimate, we define a corresponding filtering kernel  $g_d^{(p)}$  such that  $\hat{\sigma}_{k,w}^2$  estimates the variance of high-frequency streaks,  $\hat{\sigma}_{k,u}^2$  estimates the variance of horizontally low-frequency streaks, and  $\hat{\sigma}_{k,v}^2$  of vertically low-frequency streaks. For this purpose, we define  $\phi_d$  as a 1-D Gaussian function along dimension  $d$ , and  $\psi_d$  as a 1-D high-pass kernel with Daubechies wavelet 'db3' of length 6 along  $d$ . Hence, convolution with  $\phi_d$  realizes low-pass filtering, and  $\psi_d$  realizes a high-pass filter. Then,  $g_d^{(p)}$  is realized as a tensor product of three one-dimensional kernels across the dimensions  $d$  chosen based on the noise statistics through that dimension,

$$\begin{aligned} g_d^{(w)} &= \phi_0 \otimes \psi_1 \otimes \psi_2, \\ g_d^{(u)} &= \phi_0 \otimes \phi_1 \otimes \psi_2, \\ g_d^{(v)} &= \phi_0 \otimes \psi_1 \otimes \phi_2. \end{aligned}$$

Specifically, with  $m_0$ ,  $m_1$ ,  $m_2$  as the pixel sizes of the three dimensions of  $Z_k^*$ ,  $\phi_0$  is a 1-D Gaussian function along the angular dimension with standard deviation  $m_0/8$ , and  $\phi_1$  and  $\phi_2$  are 1-D Gaussian functions along the two displacement dimensions with standard deviations of  $m_1/12$  and  $m_2/12$ , respectively. Through these kernels, we obtain estimates of the three coefficients  $\hat{\sigma}_{k,p}$  as described in (10) and (11) with  $g_s$  as either  $g_{k,p}$  ( $k = K$ ) or  $g_{\mathcal{B}_{2D}} \otimes g_{k,p}$  ( $k < K$ ).

We note that these three components do not directly correspond to  $\varsigma_{k,w}$ ,  $\varsigma_{k,u}$ , and  $\varsigma_{k,v}$ , as the frequencies of the white streak component  $\eta_{k,w}^*$  partly overlap with those of  $\eta_{k,u}^*$  and  $\eta_{k,v}^*$ , *i.e.*  $\eta_{k,w}^*$  includes also some low-frequency streak components. In particular, we have  $\hat{\sigma}_{k,u}^2 \approx \varsigma_{k,u}^2 + \varsigma_{k,w}^2/m_1$  and  $\hat{\sigma}_{k,v}^2 \approx \varsigma_{k,v}^2 + \varsigma_{k,w}^2/m_2$ . To this end, we can formulate a simple non-negative least-squares optimization as

$$\begin{aligned} \underset{\hat{\varsigma}_{k,w}, \hat{\varsigma}_{k,u}, \hat{\varsigma}_{k,v}}{\text{argmin}} & \left( \hat{\varsigma}_{k,w}^2 - \hat{\sigma}_{k,w}^2 \right)^2 + \left( \hat{\varsigma}_{k,u}^2 + \frac{\hat{\varsigma}_{k,w}^2}{m_1} - \hat{\sigma}_{k,u}^2 \right)^2 \\ & + \left( \hat{\varsigma}_{k,v}^2 + \frac{\hat{\varsigma}_{k,w}^2}{m_2} - \hat{\sigma}_{k,v}^2 \right)^2. \end{aligned}$$

Finally, we construct the PSD through (22) and (23) with  $\varsigma_{k,p} = \hat{\varsigma}_{k,p}$ .

**4.1.2. Nonstationarity of  $\eta_p$ .** Variance of the streak noise may differ across the sinogram due to changes in photon flux or noise in the bright-field. Thus, it may not be possible to denoise  $Z_k^*$  assuming an equal  $\Psi_k^*$  for all spatial positions without either oversmoothing or leaving noise artifacts in some areas. To adapt to nonstationarity, we further relax the streak model allowing the PSD to vary within each scale  $k$ . In particular, before noise estimation and denoising, we split  $Z_k^*$  into overlapping, volumetric segments. We apply BM4D separately on each segment of  $Z_k^*$ , using a PSD scaled by parameters estimated from the same segment, *i.e.* we consider each segment as a separate noisy volume  $z$  with a corresponding  $\Psi$ . After denoising, the segment estimates produced by BM4D are recombined with a windowing function to form the full estimate  $\hat{Y}_k$ .

## 4.2. Attenuation of extreme streaks

We note that the projections often include several streaks caused by defects in the scintillator. These streaks can be far stronger than that reasonably produced by the distribution of  $\eta_p$  and therefore require a specific pre-processing. To this end, after the bright-fielding and log-transform and before the multiscale denoising procedure, we apply the simple extreme streak attenuation procedure as described by Mäkinen *et al.* (2021), which applies median filtering on extreme streak values detected through local polynomial fit of angular medians.

## 5. Poisson denoising

A filter for additive noise is not immediately applicable to the approximately white noise of  $\hat{Y}$  originating from the Poissonian component  $\pi$ . Firstly, the bright-fielding (1) introduces substantial spatial variability in the Poisson model. As a result, for a given optical attenuation, noise in bright-fielded projections can be stronger or weaker in different parts of the detector, for example around edges in cone-beam acquisition. Secondly, while the logarithm (2) effectively makes the streak noise additive, it also changes the typical affine-variance model of the Poissonian noise to a nonlinear one where the variance is not constant, but asymptotically inversely proportional to the mean. In order to model the noise in  $\hat{Y}$  through (6), we take care of these two issues as follows.

### 5.1. Reducing nonstationarity induced by bright-fielding

The Poissonian noise component  $\pi$  originates from a counting process which takes place before bright-fielding (1), and specifically before the division by  $I_B - I_D$ , which intro-

duces a spatially variant scaling of the variances. To undo this scaling, we consider

$$S = \hat{Y} + \ln(I_L), \quad (24)$$

where  $I_L = I_B - I_D$ . Then,  $S$  can be treated as the log-scale version of a homogeneous Poissonian process;  $S$  is thus subject to signal-dependent noise where the variance of the noise can be expressed as a smooth nonnegative function of the underlying signal,

$$\text{var}\{S\} = F(\mathbb{E}\{S\}), \quad (25)$$

where the same  $F$  applies to each pixel. In particular, it can be shown that asymptotically for large flux

$$F(\mathbb{E}\{S\}) \underset{\mathbb{E}\{S\} \rightarrow \infty}{\propto} \frac{1}{\mathbb{E}\{S\}}.$$

### 5.2. Stabilization of variance

To turn a model like (25) into (6) we again resort to the use of a VST. As large-flux asymptotics are irrelevant for denoising problems characterized by low signal-to-noise ratio, and to pragmatically accommodate for model uncertainties, we model  $F$  as a polynomial with arbitrary data-driven coefficients. The method (Foi, 2009; Azzari & Foi, 2014) simultaneously identifies the coefficients for an arbitrary signal-dependent noise model where the variance is a positive power of an unknown polynomial, and returns the associated variance stabilizing transformation  $f$  as well as the corresponding exact-unbiased inverse VST  $f_{\text{EU}}^{-1}$ . An example of an estimated standard deviation function  $(\hat{F})^{1/2}$  and the corresponding VST  $f$  are illustrated in Fig. 2, where the effectiveness of the stabilization can be deduced by the estimates of  $\text{std}\{f(S)\}$  being scattered around 1.

### 5.3. Multiscale denoising of the stabilized Poisson noise

To avoid introducing structured artifacts that are present in the bright-field and dark-field images, we further consider a smoothed version  $I_L^{\text{smooth}}$  of the field component,

$$I_L^{\text{smooth}} = g_l \circledast \text{medfilt}(I_L), \quad (26)$$

where  $g_l$  is a 2-D Gaussian kernel, and  $\text{medfilt}$  denotes a 2-D median filter. The median filter is adopted in order to remove extreme outliers (e.g. from broken pixels), and the convolution with the Gaussian ensures a smooth result. Then,  $I_L^{\text{smooth}}$  can be used for approximate correction for the bright-field induced nonstationarity with  $\hat{Y} + \ln(I_L^{\text{smooth}})$ .

The stabilized noisy stack can then be written as

$$\tilde{Z} = f[\hat{Y} + \ln(I_L^{\text{smooth}})] \approx \tilde{Y} + \tilde{\pi}, \quad (27)$$

where  $\tilde{\pi}$  corresponds to the stabilized noise and  $\tilde{Y}$  to the signal upon stabilization.

We consider  $\pi$  white, and assume the streak denoising procedure to remove all streak noise frequencies, including those of  $\pi$ . Hence, we treat  $\tilde{\pi}$  as missing the streak frequencies, i.e. with a PSD,

$$\Psi_{\tilde{Z}} = \begin{cases} 0, & \text{on angular DC plane,} \\ c, & \text{elsewhere,} \end{cases} \quad (28)$$

where  $c$  is a constant such that  $\text{var}\{\tilde{\pi}\} = c|X|^{-1}(m_0 - 1)m_0^{-1}$ .

For multiscale denoising of the Poisson component, we define three-dimensional binning and debinning functions as  $\mathcal{B}_{3D} = \mathcal{B}_{2D} \circ \mathcal{B}_\alpha$  and  $\mathcal{B}_{3D}^{-1} = \mathcal{B}_\alpha^{-1} \circ \mathcal{B}_{2D}^{-1}$ , and obtain  $K_{\text{Poi}}$  scales of binned noisy volumes as  $\tilde{Z}_k = \mathcal{B}_{3D}(\tilde{Z})$ ,  $k \in \{0, \dots, K_{\text{Poi}}\}$ . Then, unlike the progressive denoising of the streaks, we begin by BM4D denoising of  $\tilde{Z}_k$  of each scale  $k$ ; at each scale, we model the noise through a PSD of the form (28). This way, we obtain an initial estimate  $\hat{Y}_k$  of the corresponding noise-free volume  $\tilde{Y}_k$  at each scale. Then, starting from  $k = K - 1$ , we combine only the denoised volumes of each scale by recursively replacing the low-scale components of  $\hat{Y}_k$ ,  $k = \{K - 1, \dots, 0\}$ , by those of the lower scale,

$$\hat{Y}_k^* = \hat{Y}_k - g_G \mathcal{B}_{3D}^{-1}[\mathcal{B}_{3D}(\hat{Y}_k) - \hat{Y}_{k+1}], \quad (29)$$

where  $g_G$  denotes a 3-D Gaussian kernel. Although the low frequencies are obviously denoised more effectively in the coarser scale, the higher frequencies of the coarser scale are commonly estimated worse than the respective estimate of the finer scale (Facciolo *et al.*, 2017). As such,  $g_G$  realizes a low-pass filtering which selects only low frequencies of the coarser estimate to be used in the full estimate.

To account for possible remaining nonstationarity and slight correlation of the noise, we perform the denoising in segments similar to as described in Section 4.1.2 for streak noise, and estimate a separate scaling parameter  $\zeta_Z^2$  in construction of the PSD at each scale. In particular, we estimate  $\zeta_Z^2$  as described in Section 3.2 with  $g_d = \psi_0 \otimes \psi_1 \otimes \psi_2$  and  $g_s =$

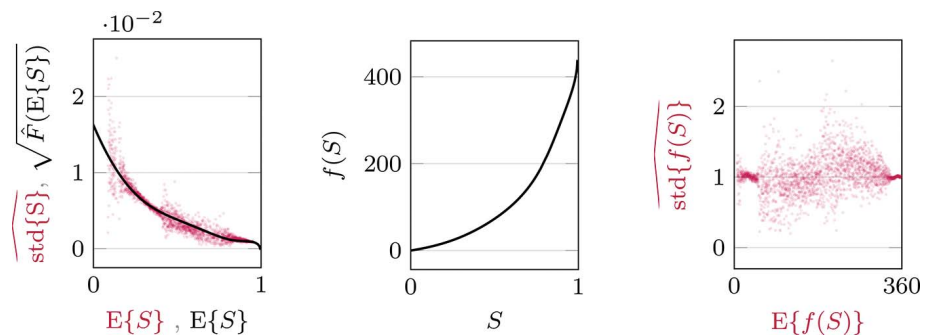
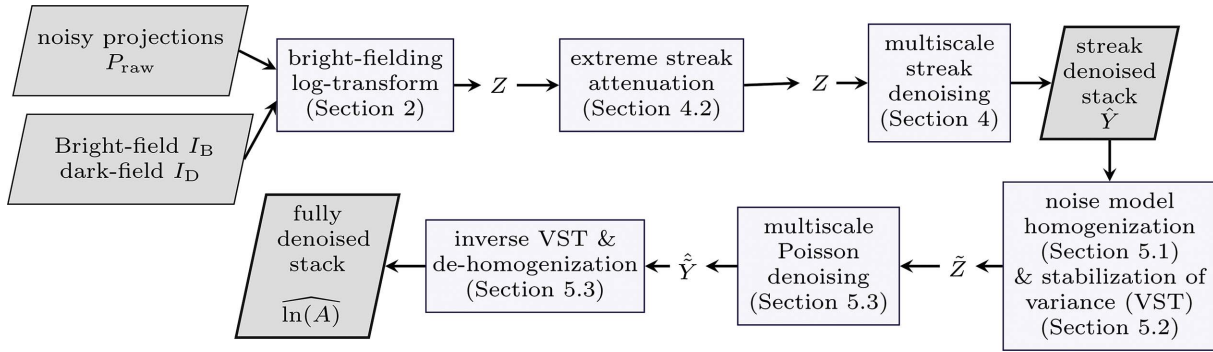


Figure 2

Illustration of the variance stabilization of *Fly*. Left: scatterplot of the sample standard deviation of  $S$  (red) as well as the square root of a polynomial approximation  $\hat{F}$  of  $F$  (black), computed on  $S$  scaled to the range  $[0, 1]$  as a function of its expectation using a fifth-degree polynomial variance model. Middle: the variance stabilizing transformation  $f$  for this  $\hat{F}$ . Right: sample standard deviation of  $f(S)$  as a function of its expectation. The standard deviations and  $f$  are both computed using Foi (2009) and Azzari & Foi (2014). Every point of the red scatterplot corresponds to the sample mean and sample standard deviation of a narrow segment of the image; the dispersion of the scatterplot is due to the finite sample size of each segment.



**Figure 3**

The full denoising process, requiring as inputs the noisy projections  $P_{\text{raw}}$  and the bright- and dark-fields  $I_B, I_D$ , (1) and producing as the output an estimate  $\widehat{\ln(A)}$  (30) of the underlying stack of projections  $\ln(A)$  (3). As an intermediate output, an estimate  $\hat{Y}$  (12) of the streak-free yet noisy stack of projections  $Y$  (5) is also produced.

$\mathcal{F}^{-1}[(\Psi_{\tilde{z}}/\|\Psi_{\tilde{z}}\|)^{1/2}]$  defining the unscaled noise correlation kernel, and finally construct the PSD through (28) with  $c = m_0(m_0 - 1)^{-1}\zeta_{\tilde{z}}^2$ .

The final estimate of the underlying stack of projections can be obtained by applying  $f_{\text{EUI}}^{-1}$  to the finest scale estimate  $\hat{Y}_0^*$  and then removing the field  $I_L^{\text{smooth}}$ ,

$$\widehat{\ln(A)} = f_{\text{EUI}}^{-1}(\hat{Y}_0^*) - \ln(I_L^{\text{smooth}}). \quad (30)$$

As (30) negates the field correction, we note that had we used the non-smooth field  $I_L$  in (27) [and respectively in (30)], any noise or spurious structures present in  $I_L$  could be introduced into  $\widehat{\ln(A)}$ , as they might have been denoised by BM4D and hence not preserved in  $\hat{Y}_0^*$ .

Upon variance stabilization, Poissonian data become asymptotically normal (Curtiss, 1943). Due to the additional Gaussianization induced by the binning and by the linear transformations operated by the filter, the assumption of normality in (7) can be adopted for denoising of the Poissonian component in this work even for low-count data.

The full denoising process is shown in Fig. 3.

## 6. Experiments

We test our pipeline on synthetic data as well as two real acquisitions displaying ring artifacts and Poisson noise.

As a comparison, we show results for Mäkinen *et al.* (2021) available on PyPI as *bm3d-streak-removal*, the proposed algorithm embedding the conventional BM4D denoiser (Maggioni *et al.*, 2012), as well as two leading streak-removal procedures from the *tomopy* Python library (Gürsoy *et al.*, 2014): Münch *et al.* (2009) and Vo *et al.* (2018). In particular, for the latter we combine ‘Algorithm 3’, ‘Algorithm 5’, and ‘Algorithm 6’, which is demonstrated by Vo *et al.* (2018) to attenuate a variety of different streaks. These streak denoising algorithms are run with the default parameters provided by the software library. To evaluate the benefit of the proposed Poisson denoising procedure with reconstruction which includes further regularization of the data, we include experiments with the iterative Total Variation (TV) reconstruction (Goldstein & Osher, 2009) of Marchesini *et al.* (2020).

For the synthetic experiments, we replicate the noise generation setup of Mäkinen *et al.* (2021) on a stack of projections ( $238 \times 181 \times 238$  pixels) obtained from a 3-D BrainWeb phantom (Cocosco *et al.*, 1997) obtained through a padding and Radon transform upon a sign change and an exponential transformation. Specifically, we regard this stack as the underlying projections  $A$  and generate noise according to (3) with  $g$  as a constant of size  $m_0 \times 1 \times 1$  (equal to  $g_{0,w}$  of Fig. 1). To obtain streak noise of different strengths, the streak noise component  $(1 + \eta_p)$  is generated with  $\text{std}\{\eta_p\} = 0.005, 0.01, 0.02, 0.05$ . Next, to generate noisy measurements with different SNR levels for the Poisson component, we separately scale  $A$  to the ranges [2560, 5120] (higher SNR), [1280, 2560], and [640, 1280] (lower SNR) and generate a Poisson variate with mean and variance  $A(1 + \eta_p)$ , thus defining the Poissonian noise  $\pi$  as the difference between this Poisson variate and  $A(1 + \eta_p)$ . Furthermore, we include experiments with  $\pi = 0$  (infinite SNR), thus resulting in a total of 16 combinations of Poisson and streak noise strengths. We do not simulate extreme streaks or the bright-fielding ( $I_B = 1$  and  $I_D = 0$ ). For the streak removal, we consider  $\ln[A + \pi/(1 + \eta_p)]$  as the streak-free yet noisy stack  $Y$ .

The results of the phantom experiments<sup>4</sup> for streak attenuation are collected in Table 1, and, for full denoising, evaluating the reconstructed volumes, in Table 2 using iterative regularized TV reconstruction with optimized regularization parameter strength  $r$ . The experiments for both streak and Poisson denoising are illustrated in Figs. 4 and 5. All reconstructions are performed upon a sign change.

The *Fly* dataset consists of 180 projections with 50 s exposure (detector pixel size  $27 \mu\text{m}$ , demagnified to  $15.7 \mu\text{m}$  by cone-beam geometry) collected using a Sigray Prisma X-ray micro-tomography instrument at 34 kV; the detector size is  $512 \times 512$  pixels. The denoising results for two different sinograms, as well as a corresponding tomogram after streak

<sup>4</sup> For the purpose of visualization and objective SNR comparison, we have corrected the intensity response of every estimate through a cubic polynomial such that it matches the ground truth. This is done particularly to improve the results of those methods such as Münch *et al.* (2009) whose intensity response significantly deviates from the ground truth.

**Table 1**

Average signal-to-noise ratio for attenuation of streaks in the BrainWeb phantom subject to mixed streak and Poissonian noise as in (3), with different combinations of  $\text{std}\{\eta_p\}$  and peak values of  $A$ , with ‘peak’ =  $\infty$  being the limiting case for which  $\pi = 0$ .

Left-to-right: noisy stack of projections  $Z$  (4), and estimates  $\hat{Y}$  of the stacks of projections  $Y$  (5) denoised by the proposed procedure (12) (‘new BM4D’), proposed procedure embedding BM4D of Maggioni *et al.* (2012) (‘old BM4D’), Mäkinen *et al.* (2021), Münch *et al.* (2009), and Vo *et al.* (2018). As the table compares only streak removal, the SNR values are calculated with respect to the streak-free yet noisy projections  $Y = \ln[A + \pi/(1+\eta_p)]$  as  $\text{SNR}(\hat{Y}) = 10 \log_{10}\{\text{svar}_X\{Y\}/\text{smean}_X\{(\hat{Y}-Y)^2\}$ , where  $\text{svar}$  and  $\text{smean}$  denote sample variance and sample mean, respectively. Each value of the table is the average SNR over 10 different noise realizations.

Peak	$\text{std}\{\eta_p\}$	$Z$	Proposed (new BM4D)	Proposed (old BM4D)	Mäkinen <i>et al.</i> (2021)	Munch <i>et al.</i> (2008)	Vo <i>et al.</i> (2018)
$\infty$ ( $\pi = 0$ )	0.005	27.80	35.10	32.53	33.70	11.28	26.30
	0.01	21.85	32.60	28.97	29.66	11.26	25.33
	0.02	16.06	29.40	23.40	24.96	11.17	23.10
	0.05	8.82	24.44	14.64	18.22	10.61	18.53
5120	0.005	27.87	32.88	28.80	32.19	11.11	26.65
	0.01	21.91	31.00	26.91	29.00	11.09	25.49
	0.02	16.09	28.38	22.87	24.74	11.00	23.16
	0.05	8.83	24.04	14.64	18.25	10.47	18.55
2560	0.005	27.93	31.63	27.23	31.06	10.96	26.58
	0.01	21.97	30.03	25.55	28.38	10.94	25.36
	0.02	16.13	27.71	22.40	24.49	10.86	23.07
	0.05	8.85	23.69	14.63	18.23	10.34	18.54
1280	0.005	28.07	30.04	25.66	29.52	10.71	26.21
	0.01	22.09	28.72	23.79	27.44	10.69	25.03
	0.02	16.22	26.77	21.57	24.09	10.62	22.85
	0.05	8.90	23.16	14.61	18.21	10.15	18.49

**Table 2**

Average SNR for the reconstructed volumes of the BrainWeb phantom for the set of experiments shown in Table 1.

Each reconstruction is performed with TV, either on the estimate  $\hat{Y}$  of the streak-free projections  $Y$  (5), or the estimate  $\ln(\hat{A})$  of the underlying stack of projections  $\ln(A)$  (4). Left-to-right: reconstructed volumes of estimates produced by the proposed full procedure with improved BM4D, proposed streak removal with improved BM4D, proposed full procedure with old BM4D, Mäkinen *et al.* (2021), Münch *et al.* (2009), and Vo *et al.* (2018). Each regularization parameter  $r$  of TV is optimized individually for best SNR for each realization and method. The SNR values are each computed against the ground-truth noise-free phantom.

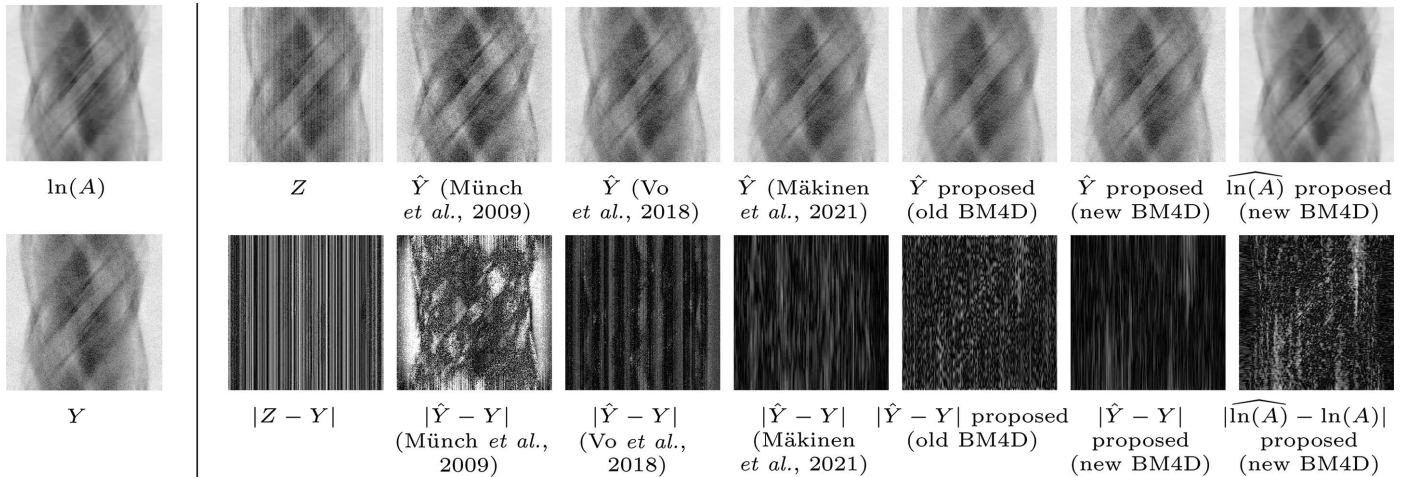
Peak	$\text{std}\{\eta_p\}$	Proposed (new BM4D) TV $\ln(\hat{A})$	Proposed (new BM4D) TV $(\hat{Y})$	Proposed (old BM4D) TV $\ln(\hat{A})$	Mäkinen <i>et al.</i> (2021) TV $(\hat{Y})$	Munch <i>et al.</i> (2008) TV $(\hat{Y})$	Vo <i>et al.</i> (2018) TV $(\hat{Y})$
$\infty$ ( $\pi = 0$ )	0.005	23.15	23.15	19.59	20.26	8.32	15.89
	0.01	21.05	21.05	17.09	18.01	8.25	15.39
	0.02	18.38	18.56	13.73	15.23	8.27	14.39
	0.05	14.23	14.75	11.29	11.05	8.04	12.11
5120	0.005	18.02	16.37	16.75	15.99	8.11	14.80
	0.01	17.45	16.00	15.70	15.41	8.10	14.49
	0.02	16.47	15.28	13.95	14.10	8.08	13.78
	0.05	14.03	13.41	11.02	10.82	7.92	11.87
2560	0.005	17.15	15.30	16.14	14.97	7.99	14.12
	0.01	16.75	15.03	15.21	14.53	7.99	13.82
	0.02	15.92	14.46	13.82	13.45	7.97	13.23
	0.05	13.76	12.85	10.94	10.67	7.87	11.67
1280	0.005	16.24	14.32	15.23	14.01	7.90	13.62
	0.01	15.95	14.13	14.71	13.71	7.90	13.40
	0.02	15.28	13.70	13.59	12.90	7.88	12.90
	0.05	13.41	12.43	10.87	10.50	7.77	11.43

attenuation, are shown in Fig. 6. A comparison of denoising on a vertical slice of the stack of tomograms is shown in Fig. 7, and a comparison for fully denoised reconstructions is shown in Fig. 8.

We also test the algorithm on a soft tissue sample 00072 displaying severe ring artifacts freely available in TomoBank (De Carlo *et al.*, 2018). The data contain 1500 projections with 1.43  $\mu\text{m}$  pixels, obtained at the Advanced Photon Source, 2-BM beamline; other experimental parameters are X-ray

energy of 20 keV, 10  $\mu\text{m}$  LuAG scintillator, and sample-to-detector distance of 15 mm. The detector size is 2160  $\times$  2560 pixels. Included are ten samples for bright- and dark-fields, which are averaged to obtain a single bright-field and dark-field. A sinogram and a corresponding tomogram from the denoising results for streak removal are shown in Fig. 9, and slices of the stack of tomograms are compared in Fig. 10. Reconstructions of fully denoised projections are further compared in Fig. 11.





**Figure 4** Denoising of the 3-D BrainWeb phantom with noise as in (3) with  $\text{std}\{\eta_p\} = 0.02$  and signal peak 2560, displaying a single sinogram extracted from the stack. Left: a sinogram of  $\ln(A)$  (3) and the corresponding streak-free but noisy sinogram of  $Y$  (5). Right: on top, noisy sinogram  $Z$  (4) and the comparison of estimates:  $\hat{Y}$  for Münch *et al.* (2009), Vo *et al.* (2018), Mäkinen *et al.* (2021), the proposed framework using old BM4D and using improved BM4D, and  $\ln(A)$  with the proposed framework and new BM4D. Below, corresponding estimation errors. Note that the errors  $|\hat{Y} - Y|$  show only the effectiveness of streak removal, as the compared algorithms are designed for streak attenuation, whereas  $|\ln(A) - \ln(A)|$  shows the complete denoising error, scaled separately. Notably, the proposed algorithm offers superior performance, but only when embedded with the improved BM4D.

The proposed method achieves superior SNR values in streak removal in all simulated noise experiments. Although the difference to the 2-D implementation of Mäkinen *et al.* (2021) is not immediately visually obvious from individual sinograms or tomograms, the displayed vertical slices of the reconstructed objects show clear improvement in both signal preservation and avoiding discontinuity between different tomograms. Compared with Münch *et al.* (2009) and Vo *et al.* (2018), the proposed method avoids creation of shadow artifacts around strong signal features. Furthermore, performing the Poisson denoising through the proposed framework allows application of standard filtered back-projection reconstruction to data originally corrupted by Poisson noise, but can also improve quality of iterative TV reconstruction.

### 6.1. Parameters

For streak attenuation, we calculate  $K$  following the formula of horizontal binning from Mäkinen *et al.* (2021), using as the base the size of the smallest displacement dimension. As a result, we use  $K = 5$  for 00072,  $K = 3$  for *Fly*, and  $K = 2$  for the phantom. These values were found to offer a reasonable compromise between denoising wide streaks versus preserving low-frequency signal components. Other processing parameters are adjusted for the smaller block size and processing neighborhood of BM4D. For angular binning, we use  $m_\alpha = \lceil m/\lceil m/32 \rceil \rceil \approx 32$  pixels, where  $m$  is the original angular size and  $m_\alpha$  the output size; the resulting size is half of that used by Mäkinen *et al.* (2021). For segmentation of the streak denoising, we use a window of size  $\lceil m_\alpha/2 \rceil \times 19 \times 19$  pixels. For the Poisson denoising, we use  $K = 1$  and  $m \times 19 \times 19$  segments. For variance stabilization, we use the implementation ClipPoisGaus (Azzari & Foi, 2015) of Foi (2009) and Azzari & Foi (2014), and use a quadratic polynomial for the variance model  $F$ .

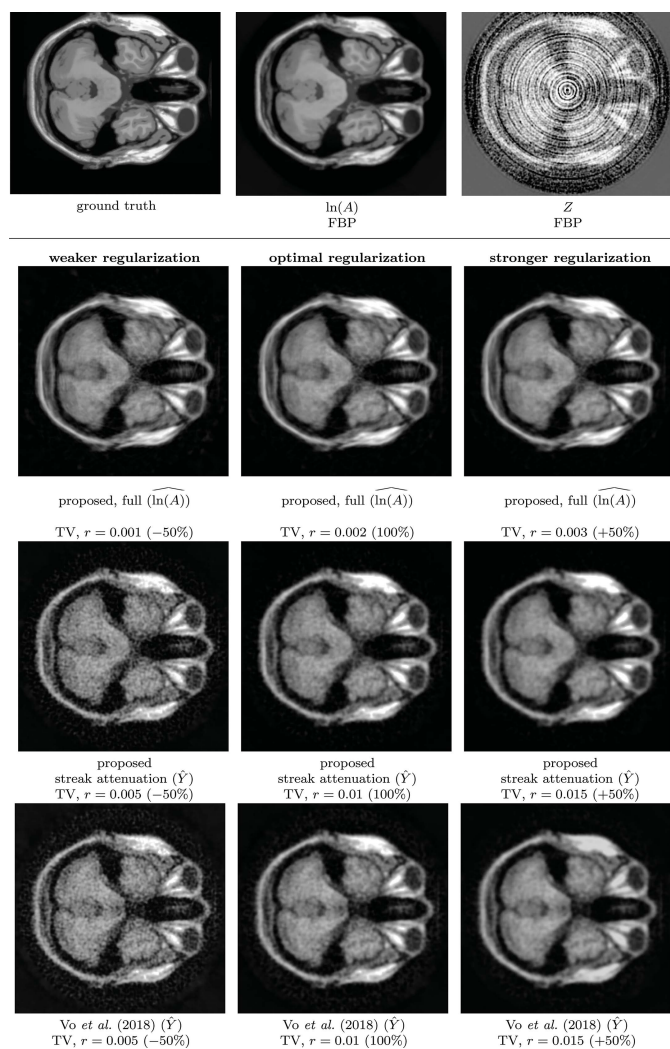
## 7. Discussion and conclusions

We have presented a framework for three-dimensional attenuation of streak noise extending the 2-D framework of Mäkinen *et al.* (2021), as well as a BM4D denoiser utilizing the algorithmic improvements of Mäkinen *et al.* (2020). Furthermore, we have included a denoising step for Poisson noise in the sinogram domain through BM4D and the adaptive variance stabilization of Foi (2009) and Azzari & Foi (2014).

We test the algorithm on both synthetic and real data, demonstrating superior SNR compared with other popular streak removal algorithms, and showing improvements in streak attenuation over Mäkinen *et al.* (2020). Furthermore, we compare the results with those which use the conventional BM4D for correlated noise, demonstrating that the included improvements in BM4D for correlated noise are essential for successful streak attenuation. The included Poisson denoising allows for full sinogram-domain denoising within the framework. By operating fully in the 3-D stack of projections, the 3-D structure of the data can be leveraged for more accurate noise removal. The proposed procedure is fully automatic and does not require extra input parameters.

To compare different methods under their own ideal conditions, we have specifically selected the TV regularization parameter values that provide the best reconstruction quality. However, in real-world applications, these values cannot be identified precisely, and too small or too large parameter values may lead to residual noise or excess smoothing of the reconstructions. Inclusion of the proposed Poisson denoising step allows for weaker regularization, but notably also reduces the effects of relative shifts in the parameter values, meaning that the reconstruction can be safely deployed even when the regularization cannot be precisely tuned.

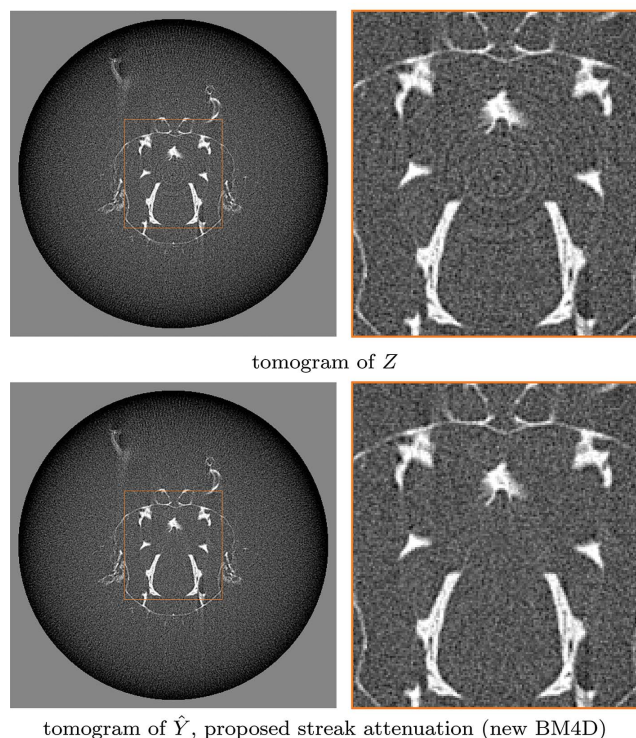
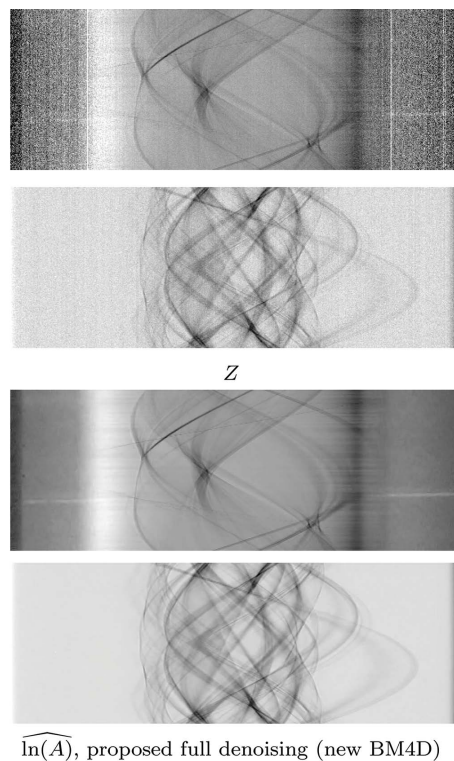
To consider the computational cost, we note that both denoising steps of *Fly* ( $181 \times 512 \times 512$  pixels) run single-



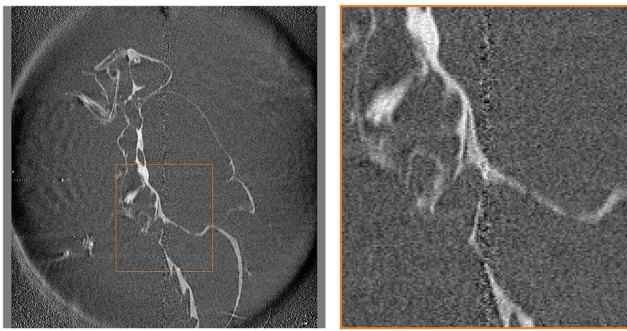
**Figure 5** Comparison of reconstructions of the 3-D BrainWeb phantom corrupted with streak and Poisson noise as in (3), corresponding to the sinograms shown in Fig. 4. Top: ground truth volume, and reconstructions of  $\ln(A)$  (3) and  $Z$  (4) obtained through filtered back-projection. Bottom: comparison of TV reconstruction of estimates with various regularization strengths  $r$ , where the percentage implies a multiplier to the regularization optimized to maximize SNR, *i.e.* 100% means ‘SNR-optimal’ regularization. Top-to-bottom: proposed full estimate  $\ln(A)$ , proposed streak-free estimate  $\hat{Y}$ , and streak-free estimate of  $\hat{Y}$  (Vo *et al.*, 2018), each with 100%, 50%, and 150% relative regularization strengths. Proposed estimates are computed embedding the improved BM4D. Notably, the full filtering offers improved reconstruction quality, and is also less sensitive to variations in the regularization parameters.

threaded on an AMD Ryzen 7 1700 processor each take about one hour. The computational cost is mostly due to the BM4D denoising in CPU. Although the adopted implementation is single-threaded, the algorithm is embarrassingly parallel, and thus a highly parallel GPU-based implementation is expected to reduce the total run time to the scale of seconds (Davy & Ehret, 2020).

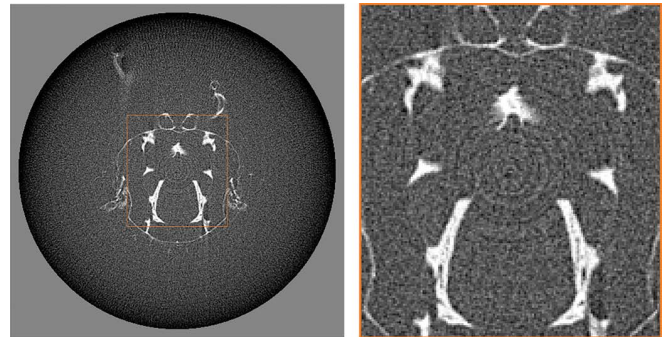
The Poissonian noise attenuation can also be performed without the preceding ring reduction step on data which do not display ring artifacts. In such case,  $\Psi_{\bar{z}}$  should be replaced by a flat PSD, as the Poissonian noise is approximately white



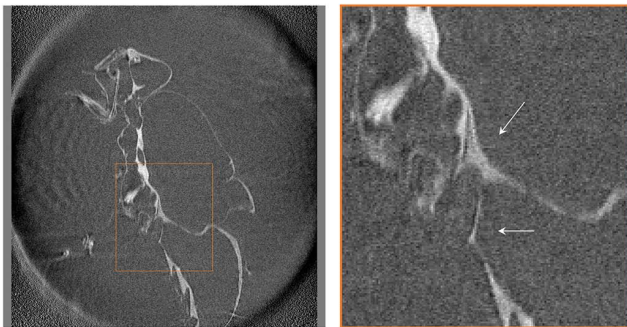
**Figure 6** Denoising of the stack of projections of *Fly*, showing two sinograms of the noisy stack of projections  $Z$  (4) and the corresponding estimates  $\ln(A)$  of the underlying stack of projections  $\ln(A)$  (3) obtained with the proposed framework (top), and the tomograms of the second sinogram (bottom), obtained with filtered back-projection using cone-beam geometry (Feldkamp *et al.*, 1984), for both the noisy data  $Z$  and the proposed estimate  $\hat{Y}$  of the streak-free projections  $Y$  (5). The tomogram for  $\ln(A)$  is shown in Fig. 8. The first sinogram shows significant model nonstationarity in both streaks and the Poissonian component due to the bright-fielding.



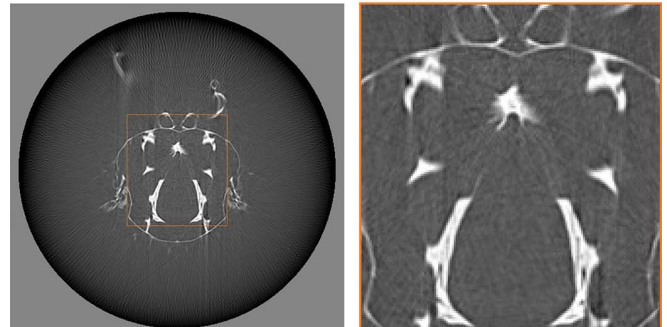
slice of tomograms of  $Z$



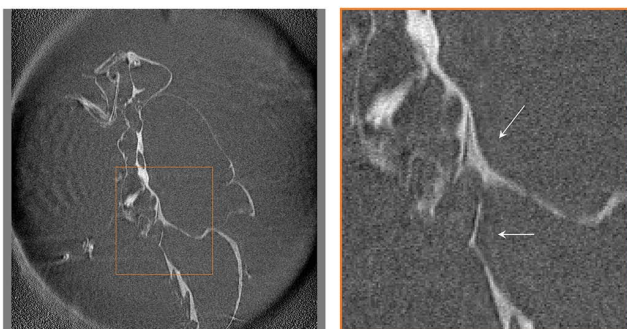
tomogram of  $Z$



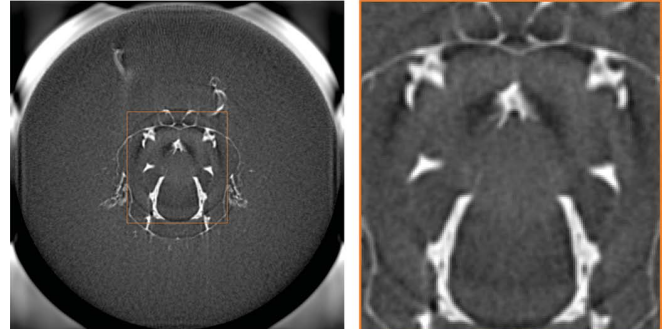
slice of tomograms of  $\hat{Y}$ , Mäkinen *et al.* (2021)



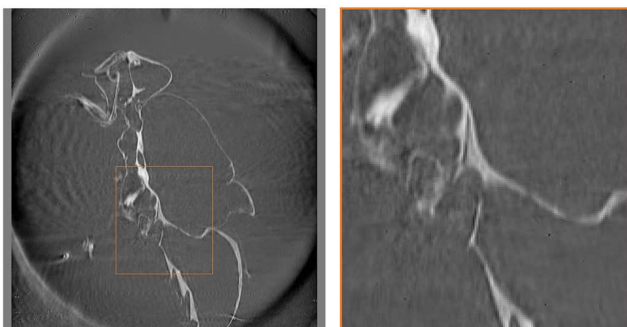
tomogram of  $\widehat{\ln(A)}$ , proposed full denoising (new BM4D)



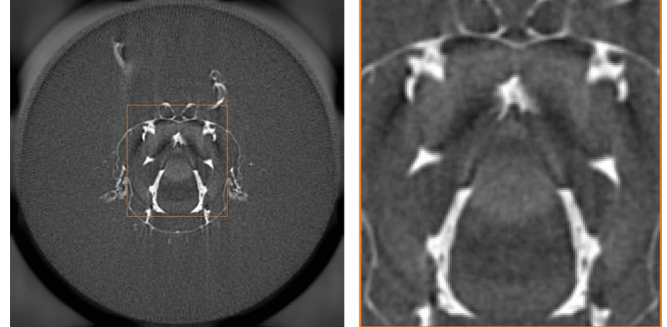
slice of tomograms of  $\hat{Y}$ , proposed streak attenuation (new BM4D)



tomogram of  $\hat{Y}$ , Münch *et al.* (2009), TV



slice of tomograms of  $\widehat{\ln(A)}$ , proposed full denoising (new BM4D)



tomogram of  $\hat{Y}$  Vo *et al.* (2018), TV

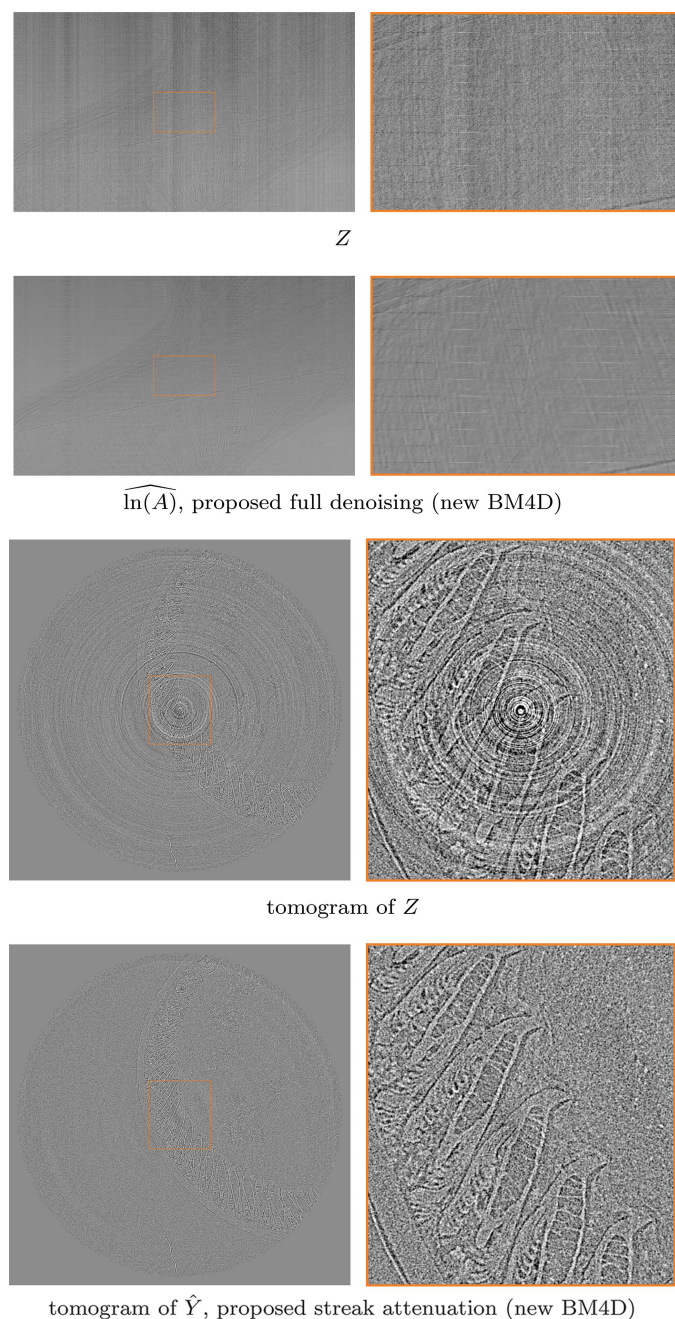
**Figure 7**

Comparison of resulting stack of tomograms after different denoising procedures on *Fly*, each obtained with filtered back-projection, where individual tomograms are horizontal lines. Displayed are slices of tomograms, reconstructed from, top-to-bottom: noisy projections  $Z$  (4), estimates  $\hat{Y}$  of the streak-free projections  $Y$  (5) denoised with Mäkinen *et al.* (2021),  $\hat{Y}$  from the proposed streak denoising, and the result of the proposed full denoising  $\widehat{\ln(A)}$ . Note the horizontal ‘streaks’ in the estimate produced by Mäkinen *et al.* (2021), which arise from differences in estimates for consecutive slices; the proposed method is not prone to such artifacts, as it considers the full stack of projections in the denoising.

**Figure 8**

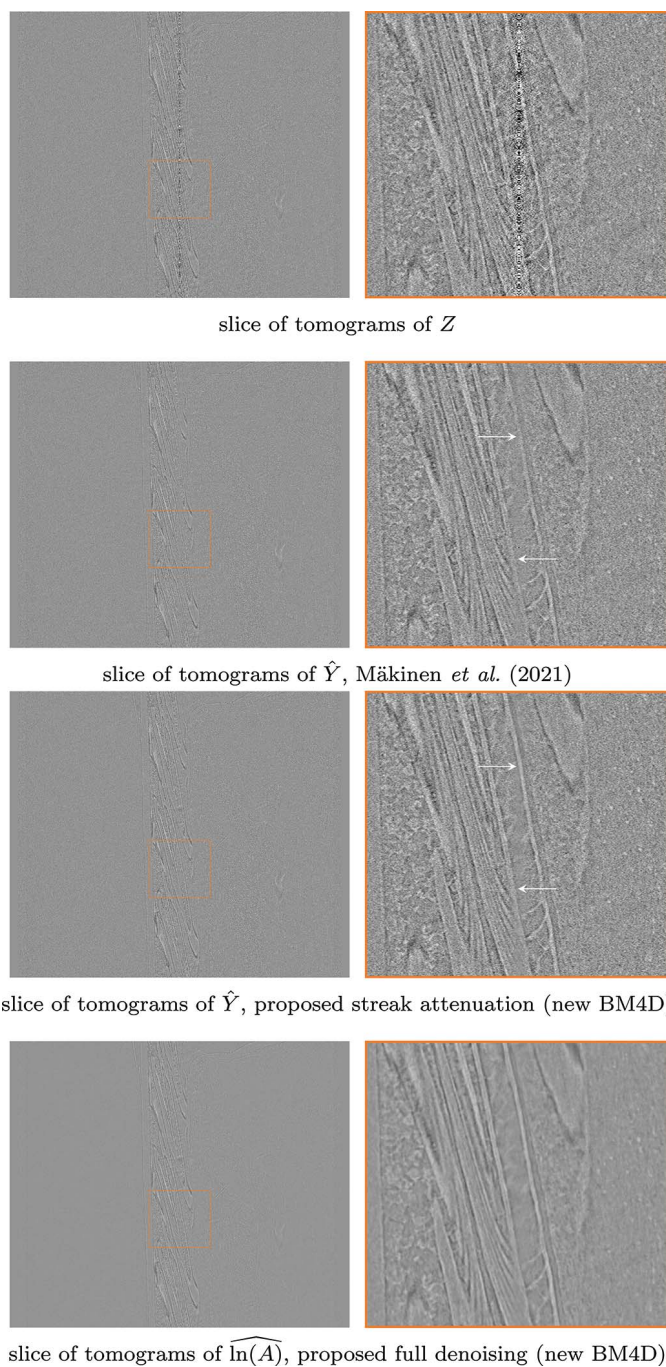
Comparison of fully denoised tomograms of *Fly*. Top-to-bottom: tomogram reconstructed from a noisy sinogram of the stack of projections  $Z$  (4), from the estimate of stack of underlying projections  $\widehat{\ln(A)}$  with proposed procedure with FBP reconstruction, and tomograms of the estimates for streak-free stacks  $\hat{Y}$  of Münch *et al.* (2009) with TV reconstruction, and  $\hat{Y}$  of Vo *et al.* (2018) with TV reconstruction. TV regularization was tuned visually, balancing residual noise and smoothing of signal. Compared with the reference methods, the proposed procedure manages to remove most noise without creating shadowlike artifacts common to Münch *et al.* (2009) and Vo *et al.* (2018).

prior to streak attenuation, whereas (28) considers the streak noise frequencies removed. Running the full denoising procedure in the absence of either streak or Poisson noise will lead to very small estimates for the corresponding noise components, meaning that no significant denoising will be performed for that noise.

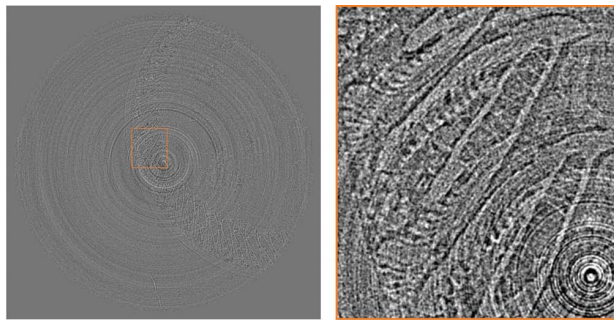


**Figure 9**  
 Denoising of the stack of projections of 00072. Top-to-bottom: display of a single sinogram of the noisy stack of projections  $Z$  (4), the corresponding estimate of the underlying projections  $\ln(A)$  from the proposed procedure, and the corresponding tomograms of  $Z$  and the estimates  $\hat{Y}$  of streak-free stacks  $Y$  (5), respectively, obtained with filtered back-projection. The tomogram for  $\ln(A)$  is shown in Fig. 11. Although the data present challenges through inconsistent noise intensities across the angular dimension, most streak noise and Poissonian noise is attenuated without notable loss of signal.

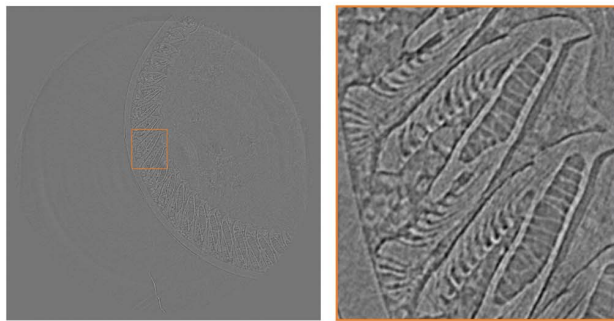
We note that although we have focused on the full denoising of the projections, typical reconstruction pipelines, such as the iterative TV, provide further noise attenuation. For best results in combining the proposed denoising procedure with



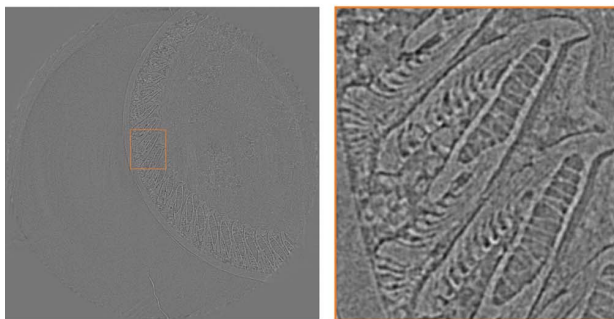
**Figure 10**  
 Comparison of central slices of resulting tomograms after different denoising procedures on 00072, where individual tomograms are horizontal lines each obtained with filtered back-projection. Displayed are slices of tomograms, reconstructed from, top-to-bottom: noisy projections  $Z$  (4), estimates  $\hat{Y}$  of the streak-free projections  $Y$  (5) denoised with Mäkinen *et al.* (2021),  $\hat{Y}$  from the proposed streak denoising, and the result of the proposed full denoising  $\ln(A)$ . Note the loss of signal near the central part of the Mäkinen *et al.* (2021) estimate, not observed in the proposed results.



tomogram of  $Z$



tomogram of  $\widehat{\ln(A)}$ , proposed full denoising (new BM4D)



tomogram of  $\hat{Y}$ , Munch *et al.* (2009), TV

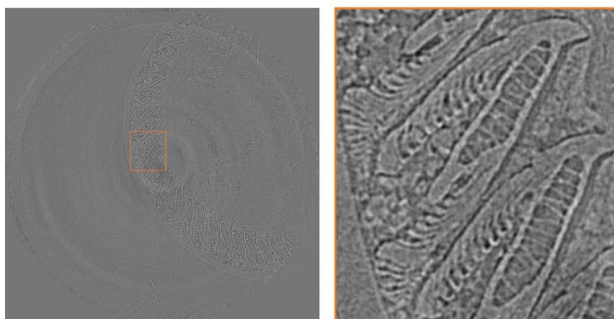


Figure 11 tomogram of  $\hat{Y}$ , Vo *et al.* (2018), TV

Comparison of fully denoised tomograms of 00072, corresponding to the sinogram in Fig. 9. Top-to-bottom: tomogram reconstructed from a noisy sinogram of the stack of projections  $Z$  (4), of the estimate of stack of underlying projections  $\ln(A)$  with the proposed procedure with FBP reconstruction, and tomograms of the estimates for streak-free stacks  $\hat{Y}$  of Munch *et al.* (2009) with TV reconstruction, and  $\hat{Y}$  of Vo *et al.* (2018) with TV reconstruction. TV regularization was tuned visually, balancing residual noise and smoothing of signal. Compared with the reference methods, the proposed procedure manages to remove most streaks without significant loss of detail, as well as most Poissonian noise without excess smoothing of the signal.

such pipelines, it may be necessary to adjust the filter strength for the denoising of Poissonian noise, *e.g.* for reduced attenuation of high-frequency noise, as it is further attenuated within the reconstruction. Likewise, integration of the proposed procedure within an iterative alternating reconstruction is left for future study.

## APPENDIX A

### Collaborative filtering and the BM4D denoising algorithm

#### A1. Collaborative filtering

The rationale of transform-domain filtering is to work with a representation of the signal where most of the signal is compacted to only a few coefficients, whereas the remaining coefficients mostly comprise noise. Hence, by attenuating the coefficients with a non-linear shrinkage operator, it is possible to attenuate noise while keeping most of the signal intact. Nonlocal collaborative filters utilize this property in the context of collective transform coefficients of groups of similar patches extracted from the input. One of the most popular collaborative filters is the Block-Matching and 3-D filtering (BM3D) (Dabov *et al.*, 2007) denoising algorithm, which performs denoising on groups of blocks extracted from a 2-D image. In the BM4D volumetric denoiser (Maggioni *et al.*, 2012), the patches are 3-D volumes extracted from the volumetric data.

All operations of collaborative filters are made with regard to a reference patch moving through the volume. For each position of the reference patch, the following steps are executed:

- (1) Collect similar patches into a group through **patch-matching**.
- (2) Obtain a group transform spectrum by collectively transforming the group of patches.
- (3) Perform **shrinkage**.
- (4) Transform the shrunk spectra back to patch estimates and **aggregate** them to the original locations from which they were collected.

For details about the algorithm in arbitrary dimensionality, we refer the reader to Mäkinen *et al.* (2020). In the following section, we describe special considerations for the implementation of the algorithmic improvements in Mäkinen *et al.* (2020) for volumetric denoising.

#### A2. Improved BM4D for correlated noise

Most of the improvements described by Mäkinen *et al.* (2020) are directly applicable to the 3-D denoiser. In this section, we consider extensions which are not immediate from the inclusion of an extra dimension.

**A2.1. Shrinkage parameters.** For the selection of shrinkage parameters  $\lambda$  and  $\mu^2$ , we embed the parameter selection subroutine of Mäkinen *et al.* (2020), which is based on the 2-D input PSD; for simplicity, we adopt directly also the pre-computed features and parameters computed for a set of 2-D PSDs. To utilize this system with a 3-D PSD, we include a

simple procedure which obtains a 2-D projection of the 3-D PSD by preserving the two largest principal components, aiming to preserve the characterizing features of the PSD shape. This projection is then used to compute features as described for a 2-D PSD by Mäkinen *et al.* (2020) for the estimation of suitable  $\lambda$  and  $\mu^2$ .

**A2.2. Fast implementation.** We consider the fast implementation as suggested by Mäkinen *et al.* (2020). In particular, we perform all operations on a downscaled PSD of size  $N_f \times N_f \times N_f$  and compute exactly only the  $K_f$  first volumes of the 4-D spectrum and approximate the rest using the conventional variances. Furthermore, Fourier symmetries and sparsity of the transformed arrays can be exploited to reduce computational cost similar to the 2-D case.

**A2.3. Refiltering.** As noted by Mäkinen *et al.* (2020), even with exact modeling of the collaborative transform-domain noise spectrum, the accuracy of collaborative filtering is limited by the systemic factors arising from the used transforms, both in size and possible symmetries of the transform spectrum which may limit the modeling of the global PSD. As a result, the denoising may attenuate excess signal, leading to oversmoothing in some frequencies; Mäkinen *et al.* (2020) proposes the mitigation of these systemic issues through an extra filtering step performed on the denoising residual. The three-dimensional spectra are not exempt from these limitations, and as such we adopt the global Fourier thresholding and refiltering procedure through a 3-D FFT.

### Acknowledgements

We are very thankful to Dr Sheraz Gul of Sigray Inc. for providing us with the experimental *Fly* data.

### Funding information

This work was supported by the Academy of Finland (project No. 310779) and by the Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory, which is supported by the US Department of Energy, Office of Science, under Contract No.DE-AC02-76SF00515. Partial support by Hong Kong RGC 14302920.

### References

Abu Anas, E. M., Lee, S. Y. & Hasan, M. K. (2010). *Phys. Med. Biol.* **55**, 6911–6930.  
 Artul, S. (2013). *BMJ Case Rep.* **2013**, bcr-2013–201379.  
 Azzari, L. & Foi, A. (2014). *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, 4–9 May 2014, Florence, Italy, pp. 5357–5361. IEEE.  
 Azzari, L. & Foi, A. (2015). *ClipPoisGaus: Poissonian-Gaussian noise estimation and removal for single-image raw-data, Matlab code*, [https://webpages.tuni.fi/foi/sensornoise.html#ref\\_software](https://webpages.tuni.fi/foi/sensornoise.html#ref_software).

Boas, F. E. & Fleischmann, D. (2012). *Imaging Med.* **4**, 229–240.  
 Cocosco, C. A., Kollokian, V., Kwan, R. K.-S., Pike, G. B. & Evans, A. C. (1997). *NeuroImage*, **5**, S425.  
 Croton, L. C., Ruben, G., Morgan, K. S., Paganin, D. M. & Kitchen, M. J. (2019). *Opt. Express*, **27**, 14231–14245.  
 Curtiss, J. H. (1943). *Ann. Math. Stat.* **14**, 107–122.  
 Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2007). *IEEE Trans. Image Process.* **16**, 2080–2095.  
 Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. O. (2008). *Proc. SPIE*, **6812**, 681207.  
 Davy, A. & Ehret, T. (2021). *J. Real-Time Image Proc.* **18**, 57–74.  
 De Carlo, F., Gürsoy, D., Ching, D. J., Batenburg, K. J., Ludwig, W., Mancini, L., Marone, F., Mokso, R., Pelt, D. M., Sijbers, J. & Rivers, M. (2018). *Meas. Sci. Technol.* **29**, 034004.  
 Facciolo, G., Pierazzo, N. & Morel, J.-M. (2017). *SIAM J. Imaging Sci.* **10**, 1603–1626.  
 Feldkamp, L. A., Davis, L. C. & Kress, J. W. (1984). *J. Opt. Soc. Am. A*, **1**, 612–619.  
 Foi, A. (2009). *Signal Process.* **89**, 2609–2629.  
 Goldstein, T. & Osher, S. (2009). *SIAM J. Imaging Sci.* **2**, 323–343.  
 Gürsoy, D., De Carlo, F., Xiao, X. & Jacobsen, C. (2014). *J. Synchrotron Rad.* **21**, 1188–1193.  
 Haibel, A. (2008). *Advanced Tomographic Methods in Materials Research and Engineering*, pp. 141–160. Oxford University Press.  
 Hampel, F. R. (1974). *J. Am. Stat. Assoc.* **69**, 383–393.  
 Jha, A. K., Purandare, N. C., Shah, S., Agrawal, A., Puranik, A. D. & Rangarajan, V. (2013). *Indian J. Nucl. Med.* **28**, 232–233.  
 Li, Y., Zhao, Y., Ji, D., Lv, W., Xin, X., Zhao, X., Liu, D., Ouyang, Z. & Hu, C. (2021). *Phys. Med. Biol.* **66**, 105011.  
 Maggioni, M., Katkovnik, V., Egiazarian, K. & Foi, A. (2012). *IEEE Trans. Image Process.* **22**, 119–133.  
 Mäkinen, Y., Azzari, L. & Foi, A. (2020). *IEEE Trans. Image Process.* **29**, 8339–8354.  
 Mäkinen, Y., Marchesini, S. & Foi, A. (2021). *J. Synchrotron Rad.* **28**, 876–888.  
 Marchesini, S., Trivedi, A., Enfedaque, P., Perciano, T. & Parkinson, D. (2020). *Lecture Notes Comput. Sci.* **12137**, 248–261.  
 Massimi, L., Brun, F., Fratini, M., Bukreeva, I. & Cedola, A. (2018). *Phys. Med. Biol.* **63**, 045007.  
 Mohan, K. A., Venkatakrishnan, S., Drummy, L. F., Simmons, J., Parkinson, D. Y. & Bouman, C. A. (2014). *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, 4–9 May 2014, Florence, Italy, pp. 6909–6913. IEEE.  
 Münch, B., Trtik, P., Marone, F. & Stampanoni, M. (2009). *Opt. Express*, **17**, 8567–8591.  
 Paleo, P. & Mirone, A. (2015). *J. Synchrotron Rad.* **22**, 1268–1278.  
 Pelt, D. M. & Parkinson, D. Y. (2018). *Meas. Sci. Technol.* **29**, 034002.  
 Seibert, J. A., Boone, J. M. & Lindfors, K. K. (1998). *Proc. SPIE*, **3336**, 348–354.  
 Sijbers, J. & Postnov, A. (2004). *Phys. Med. Biol.* **49**, N247–N253.  
 Swinehart, D. F. (1962). *J. Chem. Educ.* **39**, 333.  
 Venkatakrishnan, S. V., Bouman, C. A. & Wohlberg, B. (2013). *2013 IEEE Global Conference on Signal and Information Processing (GlobalSIP 2013)*, 3–5 December 2013, Austin, Texas, USA, pp. 945–948. IEEE.  
 Vidal, F. P., Létang, J. M., Peix, G. & Cloetens, P. (2005). *Nucl. Instrum. Methods Phys. Res. B*, **234**, 333–348.  
 Vo, N. T., Atwood, R. C. & Drakopoulos, M. (2018). *Opt. Express*, **26**, 28396–28412.