Ring artifact reduction via multiscale nonlocal collaborative filtering of spatially correlated noise

Ymir Mäkinen, Stefano Marchesini and Alessandro Foi

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

X-ray micro-tomography systems often suffer severe ring artifacts in reconstructed images. These artifacts are caused by defects in the detector, calibration errors, and fluctuations producing streak noise in the raw sinogram data. In this work, these streaks are modeled in the sinogram domain as additive stationary correlated noise upon logarithmic transformation. Based on this model, a streak removal procedure is proposed where the Block-Matching and 3-D (BM3D) filtering algorithm is applied across multiple scales, achieving state-of-the-art performance in both real and simulated data. Specifically, the proposed fully automatic procedure allows for attenuation of streak noise and the corresponding ring artifacts without creating major distortions common to other streak removal algorithms.

1. Introduction

Ring artifacts are ubiquitous in computed tomography (Jha et al., 2013; Artul, 2013; Boas & Fleischmann, 2012); they originate from angular streak noise in measured raw sinogram data used to reconstruct a tomographic volume (Croton et al., 2019) and appear as darker or lighter circles or arcs centered on the axis of rotation for data acquisition. Streak noise can be caused by mis-calibration of detector linear response, beam fluctuations, beam hardening, or dusty or damaged scintillator screens (Haibel, 2008; Vidal et al., 2005; Anas et al., 2010).

Minimization of ring artifacts by using adequate scanning protocols (Pelt & Parkinson, 2018), high quality scintillator screens and detectors is possible. It is, however, difficult to completely avoid such artifacts and therefore achieve highest quality reconstruction solely by experimental measures. Several algorithms have been proposed to reduce ring artifacts in tomographic imaging, including 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), or iterative algorithms (Paleo & Mirone, 2015) that combine regularized reconstruction with denoising.

In this work, we model the streak noise as a spatially correlated noise in the sinogram domain, and propose a denoising procedure aiming to remove the streak noise before reconstruction. The denoising procedure is based on collaborative filtering, which employs both non-local self-similarity and transform-domain shrinkage to denoise a noisy signal through jointly transformed grouped blocks. In particular, we use the image denoising algorithm BM3D (Dabov et al., 2007, 2008), leveraging the recent inclusion of exact transform-domain noise variances (Mäkinen et al., 2020), which allow for accurate modeling of long noise correlation within the jointly transformed blocks.
2. Transform-domain collaborative filtering of correlated noise

In this section, we interpret sinogram streaks as spatially correlated noise, formalizing streak removal as filtering of correlated noise where the streaks follow a basic stationary model. As a powerful tool to deal with this model, we adopt a recent BM3D designed for dealing with long-range correlation such as that which characterizes the streaks. This constitutes the denoising module at the core of a multiscale and nonstationary filtering architecture that will be presented in Section 3 for the more general case of real-world streak noise.

2.1. Correlated noise model

We consider a 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,

\[
z(x) = y(x) + \eta(x), \quad x \in X,
\]

where \( x \in X \subseteq \mathbb{Z}^2 \) is the coordinate in the finite two-dimensional image domain \( X \) (representing angles and displacements when \( z \) is a sinogram) and

\[
\eta = v \otimes g, \quad v(\cdot) \sim \mathcal{N}(0, 1),
\]

with \( v \) being the zero-mean independent and identically distributed (i.i.d.) Gaussian noise with unit variance, and \( \otimes \) denoting 2-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) \( \Psi \),

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

with \( \mathcal{F} \) being the 2-D Fourier transform and \( |X| \) denoting the cardinality (i.e. number of elements) of \( X \). Equivalently, a kernel \( g \) satisfying (2)–(3) can be defined from \( \Psi \) as

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

2.1.1. Basic model for constant stationary streaks

Although the streaks are originally multiplicative in nature, sinogram data are considered upon a logarithmic transformation and therefore the streaks can be modeled by the additive noise \( \eta \) in (1). The sinogram streak noise is fairly constant in the angular dimension, presenting very long-range correlation in the noise along this dimension. Treating angle as the vertical dimension and displacement as horizontal, we consider the basic case of horizontally white and vertically constant streak noise. Such noise can be modeled through (2), when setting \( g \) as a vertically constant line. This simple kernel as well as the corresponding PSD \( \Psi \) are demonstrated in Figure 1; an example of real streak noise viably approximated through this model is shown in Figure 2.

In practice, the above simple model cannot be used but for small segments of the sinograms, as streak noise can often feature horizontal correlation, vertical variations, or nonstationarities that are not described by the model. In Section 3, a complete processing pipeline is further proposed to allow modeling more complex cases of streak noise through (1)–(4), enabling their attenuation through the collaborative filter.

2.2. Transform-domain collaborative filtering and BM3D

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 blocks extracted from the image. In all of the following sections, we consider the recently proposed variant (Mäkinen et al., 2020) of BM3D for correlated noise denoising where the
input is \( z \) and the goal of denoising is to estimate \( y \) based on the statistics of \( \eta \) or equivalently knowledge of \( \Psi \) or \( g \).

In BM3D, all operations are made with regard to a reference block moving through the image. For each position of the reference block, the following steps are executed:

(i) Collect similar blocks into a group through block-matching.

(ii) Obtain the 3-D transform spectrum by collectively transforming the obtained blocks.

(iii) Perform shrinkage.

(iv) Transform the shrunk spectra back to block estimates and aggregate them to the original locations from which they were collected.

The 3-D transform spectrum of the grouped noisy blocks is obtained through first applying a 2-D transform \( \mathcal{T}^{2D} \) locally to each block, then a 1-D transform \( \mathcal{T}^{NL} \) through the ‘stack’ of grouped blocks. Denoting by \( \{z_{x_1} \ldots, z_{x_M}\} \) a group of \( M \) blocks of \( N \) pixels extracted from \( z \) at coordinates \( x_1, \ldots, x_M \), we obtain the \( \mathcal{T}^{2D} \) spectrum coefficients as \( s_i^y = \{z_{x_i}, \beta_i^{2D}\} \), for \( i = 1, \ldots, N, \) \( t = 1, \ldots, M \), where \( \beta_i^{2D} \) is the \( t \)th basis function of \( \mathcal{T}^{2D} \).

The 3-D spectrum coefficients are calculated through the direct tensor product of the \( \mathcal{T}^{2D} \) and \( \mathcal{T}^{NL} \) transforms, as

\[
\mathcal{s}_{i,j}^{x_1,\ldots,x_M} = \left[[z_{x_1}, \ldots, z_{x_M}], \beta_i^{2D} \otimes b_j^{NL}\right] = \left[[s_i^y, \ldots, s_M^y], b_j^{NL}\right] = \sum_{i=1}^{M} b_j^{NL}(t) s_i^y, \tag{5}
\]

where \( b_j^{NL}(t) \) is the \( j \)th element of the \( b_j^{NL} \) of \( \mathcal{T}^{NL}, \otimes \) denotes the tensor product, and \( [\ldots] \) denotes the stacking of the blocks along the third dimension. The indexing and notation of the transform spectrum coefficients are illustrated in Figure 3.

In each step of the algorithm, the variances of \( s_i^y \) play a key role; we denote them by \( \nu_i^{x_1,\ldots,x_M} \). For their calculation, we refer the reader to Määkinen et al. (2020). Here, we provide a summary of the macroscopic operations of the algorithm.

2.2.1. Block-matching. For each reference block, BM3D defines a local neighborhood from which similar blocks are collected. Each block in the neighborhood is ranked by

\[
L_{\nu_i}(s_i) = \left\| z_i - z_{\nu_i} \right\|^2 - 2\gamma \sum_{i=1}^{N} \nu_i^{x_2-x_1}, \tag{6}
\]

where \( z_{\nu_i} \) is the reference block, \( z_i \) is a potential match, \( \nu_i^{x_2-x_1} \) is the \( i \)th coefficient of the block-pair transform-domain variance corresponding to block difference, and \( \gamma \in \mathbb{R} \). The common aim of block-matching is to find blocks which are the most similar to the reference block in terms of the underlying noise-free content. When only a noisy image is available, the similarity is evaluated between noisy blocks and the term scaled by \( \gamma \) in (6) compensates for bias in the ranking caused by noise correlation. Specifically, with \( \gamma = 0 \) the matches would be mainly guided by the strong vertical correlation of the streak noise and thus be located along the streaks, largely ignoring any similarity of the underlying signal; setting \( \gamma > 0 \) mitigates this bias by promoting matching of blocks in which the noise is not correlated with that of the reference block. In particular, we employ \( \gamma = 3 \) as proposed by Määkinen et al. (2020) for the general case of correlated noise, facilitating further the matching of blocks which differ from the reference block mainly due to the variance of the block difference.

The common design of BM3D includes two distinct stages of denoising with different shrinkage operators, meaning that the full image is processed twice. In the second stage, the block-matching is commonly executed on the image estimate produced by the first denoising stage. As this image can be presumed noise-free, the second block-matching is executed without any compensation for noise correlation.

2.2.2. Shrinkage of the 3-D spectra. The core of BM3D is shrinkage performed on the 3-D transform spectrum of the grouped noisy blocks. For a given transform-domain coefficient of the group, a generic shrinkage can be expressed as

\[
\mathcal{s}_{i,j}^{x_1,\ldots,x_M} \mapsto \alpha_{i,j} \mathcal{s}_{i,j}^{x_1,\ldots,x_M}, \tag{7}
\]

where \( \alpha_{i,j} \) is a shrinkage attenuation factor which depends on \( s_i^{x_1,\ldots,x_M} \), the noise statistics, and possible other priors.

BM3D utilizes two shrinkage operations: in the first denoising stage, the denoising process performs shrinkage by hard-thresholding; the second stage employs a Wiener filter, utilizing the hard-thresholding image estimate as a pilot signal.

In hard-thresholding, the shrinkage is performed by setting spectrum coefficients smaller than a threshold to zero, as they are mostly composed of noise,

\[
\alpha_{i,j}^{HT} = \begin{cases} 1, & \text{if } \left| s_i^{x_1,\ldots,x_M} \right| \geq \left( \nu_i^{x_2-x_1} \right)^{1/2} / \lambda, \\ 0, & \text{otherwise}, \end{cases} \tag{8}
\]

where \( \lambda \geq 0 \) is a fixed constant.

In Wiener filtering, the attenuation coefficients of the transfer function are computed from the previous estimate, used as pilot signal, and from the variance of the noise spectrum coefficients as

\[
\alpha_{i,j}^{\text{w}} = \frac{\left\| \left[\hat{y}_{i,j}^{HT}, \ldots, \hat{y}_{i,j}^{HT}\right], \beta_i^{2D} \otimes b_j^{NL}\right\|_2}{\left\| \left[\hat{y}_{i,j}^{HT}, \ldots, \hat{y}_{i,j}^{HT}\right], \beta_i^{2D} \otimes b_j^{NL}\right\|_2 + \mu^2 \nu_i^{x_2-x_1}}, \tag{9}
\]

where \( \hat{y}^{HT} \) is the estimate of \( y \) obtained from the hard-thresholding stage, and \( \mu^2 \) is a scaling factor included due to

![Figure 3](image-url)

Figure 3: Notation and indexing of patch coordinates \( x_i \), patches \( z_{x_i} \), and coefficients \( s_i^y \) and \( s_i^{x_1,\ldots,x_M} \) in the corresponding \( \mathcal{T}^{2D} \) and \( \mathcal{T}^{NL} \) spectrum, reproduced with permission from Määkinen et al. (2020). The illustration is for a group of three blocks of size \( 2 \times 2 \) at coordinates \( x_1 = (4, 3), x_2 = (7, 5), x_3 = (8, 6) \) within a \( 10 \times 10 \) pixel image.
aggregation to influence the bias-variance ratio we wish to minimize through the Wiener filter.

2.2.3. Aggregation. After calculating the attenuation factors of the group, they can be applied to the 3-D transform spectra to obtain estimates for the grouped blocks,

\[ \hat{y}_j = Q^{2D}{\left( \{ \omega_{x_j}^L, S_j \} \right)}, \]

(10)

where \( Q^{2D} \) is the inverse transform of \( T^{2D} \), and \( \omega_{x_j}^L \) is the \( j \)th transform basis function of the inverse of \( T^{NL} \).

Hence, an estimate is produced for all blocks included in the group. As a new group is built for every position \( x_R \) of the reference block, there is a large amount of block estimates providing a highly redundant covering of the image. Let \( X_R \) be the set of coordinates of all reference blocks and denote by \( \{ \hat{y}_j, j=1 \ldots M \} \) the set of estimates (10) for the group of blocks matched to the reference block at position \( x_R \). Then, the block estimates of an image are \( \bigcup_{x \in X_R} \{ \hat{y}_j, j=1 \ldots M \} \) and they can all be distinct.

We aggregate all block estimates at their respective positions into the image through an adaptive weighted average,

\[ \hat{y} = \frac{\sum_{x \in X_R} \sum_{j=1}^{M} \omega_{x}^L \cdot W_j \cdot y_j}{\sum_{x \in X_R} \sum_{j=1}^{M} \omega_{x}^L \cdot W_j}, \]

(11)

where \( \omega_{x}^L \) is a block-specific weight and \( W_j \) is a windowing function over blocks at position \( x \). The weights \( \omega_{x}^L \), inversely proportional to the residuals of transform-domain noise variances, promote estimates with less residual noise to improve the quality of the final estimate.

The steps for denoising a group of blocks are demonstrated in Figure 4.

3. Processing pipeline

In this section, we consider the necessary steps for modeling the streak noise through (1) for real sinogram data, hence allowing the effective application of BM3D for streak removal.

3.1. Bright-fielding and log-transformation

The optical attenuation through the sample is determined experimentally via bright-field corrections requiring two additional inputs: the bright-field and the dark-field (Seibert et al., 1998). The bright-field is an acquisition 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. Furthermore, the Beer–Lambert law relates the X-ray transform through the sample to the optical attenuation by a logarithmic transformation (Swinehart, 1962).

Hence, the raw projections \( I_{raw} \) are first normalized as

\[ P_{norm} = \frac{I_{raw} - I_{D}}{I_{H} - I_{D}}, \]

(12)

where \( I_{D} \) is the dark-field and \( I_{H} \) is the bright-field,\(^1\) and then log-transformed as

\[ P_{log} = \ln(P_{norm}). \]

(13)

Bright-fielding (12) provides a partial, but not thorough, correction of the streak noise (Davidson et al., 2003); the denoising pipeline of the following sections is designed to attenuate the remaining streak noise.

3.1.1. Noise in the projections. Apart from possible completely defective detectors\(^2\) we treat the variation in detector response as normally distributed. We further model the streak noise as locally stationary, meaning that the streak variance is presumed constant within sufficient area (i.e., the block-matching search neighborhood) for the application of BM3D. 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. Considering both the approximately normally distributed streak noise and the Poissonian component, noise in projections normalized by (12) can be modeled as

\[ P_{norm} = A(1 + \eta_p) + \pi = \left( A + \frac{\pi}{1 + \eta_p} \right)(1 + \eta_p). \]

(14)

where \( A \) are the noise-free projections, \( \eta_p \) is the normally distributed streak noise component, and \( \pi \) (is approximately) white Poissonian noise with zero mean; all components of (14) are considered as 3-D arrays and multiplications are elementwise.

We note that the natural logarithm of (13) acts as a variance-stabilizing transformation for the multiplicative noise component \((1 + \eta_p)\). Hence, we have

\(^1\) As \( P_{raw} \) is a 3-D array, the pixels of \( I_{raw} \) and \( I_{D} \) are replicated through the angle dimension for the operations in (12).

\(^2\) Extreme streak noise arising from defective detectors is addressed separately in Section 3.4.

Figure 4

Denoising a part of \( Fly \) (a portion of Figure 2) with vertical streak noise with \( \Psi \) as in Figure 1. Left to right: (a) \( x_1 \ldots x_8 \) of one group of blocks with reference block in red; (b) contents of the eight matched blocks \( z_{i_1} \ldots z_{i_8} \); (c) the resulting \( T^{3D} \) spectrum coefficients \( s_{i_1}^{\Psi} \); (d) corresponding 3-D noise standard deviations \( (\sigma_{i_1}^{\Psi})^2 \); (e) hard-thresholded coefficients \( \omega_{i_1}^{\Psi} \); (f) the denoised group of blocks \( \hat{y}^{\Psi} \); and (g) the denoising result of hard-thresholding \( \hat{y}^{\Psi} \). For the spectrum coefficients and the standard deviations \((c, d, e)\), 50%-gray pixel color in the figure corresponds to value 0 in the data.
\[ P_{\text{log}} = \ln(P_{\text{norm}}) \approx \ln \left( A + \frac{\pi}{1 + \eta_r} \right) + \eta_r, \]  
(15)

where the approximation comes from \( \ln(1+\eta_r) \approx \eta_r \). The additive noise component \( \eta_r \) in (15) corresponds to the streak noise to be denoised. As here we only aim to attenuate the streak noise, through denoising we estimate \( \ln[A + \pi/(1 + \eta_r)] \); the embedded noise term \( \pi/(1 + \eta_r) \) although not i.i.d. is nevertheless (approximately) white and does not present streaks.

Individual sinograms, each of which is defined as a cross section of the stack of projections \( P_{\text{log}} \), are denoted as

\[ Z = Y + \eta_z, \]  
(16)

where \( Y \) denotes the underlying streak-free sinogram, and \( \eta_z \) is the corresponding cross section of \( \eta_z \). The sinograms \( Z \) are used as the input for the processes in the following Section 3.2.

### 3.2. Multiscale filtering architecture

In the following, we assume that sinograms \( Z \) are oriented such that streaks are oriented vertically, i.e. the angular component is vertical and the displacement is horizontal.

The streak noise \( \eta_z \) is characterized by very long-range correlation. In particular, because vertically there are no high-frequency streak noise components, the streaks can be filtered entirely at a coarse vertical scale, with subsequent benefits in terms of efficacy and computational efficiency. Furthermore, BM3D operates using blocks of fixed size within a limited neighborhood which may be too small to fully denoise the wider streaks. Thus, we also want to denoise across multiple horizontal scales to effectively attenuate streaks of varying sizes.

Our multiscale implementation is based on a simple and efficient pixel binning to go towards coarser scales by replacing adjacent pixels by their sum. To go back towards finer scales we leverage the iterative debinning approach from Azzari & Foi (2016), which is based on spline upsampling. The multiscale denoising process is illustrated in Figure 5 and proceeds as follows.

We begin with a single vertical binning of the full noisy sinogram \( Z \) of height \( m \) to a sinogram \( Z_0 \) of height \( m_c \leq m \) through a binning operator \( B_r \). On \( Z_0 \), we perform all consequent horizontal operations and denoising.

After vertical binning, the sinogram \( Z_0 \) is progressively halved in size \( K \) times through a horizontal binning operator \( B_h \): \( Z_k = B_h(Z_{k+1}) = B_h^k(Z_0) = B_h^k[B_h(Z)] \), \( k = 1, \ldots, K \). Denoting by \( n \) the width of \( Z \) and \( Z_0 \), \( Z_k \) has width \( 2^{-k}n \); with every binning, the streak width also gets halved. The multiscale denoising is operated in a coarse-to-fine fashion, where progressively for each \( k = K, K-1, \ldots, 2, 1, 0 \) we obtain an estimate \( \hat{Y}_k \) of \( B_h^k[B_h(Y)] \). We start by taking as noisy input \( Z_k^h \) of BM3D the smallest binned sinogram \( Z_k^h \); in this way, we obtain from \( Z_k^h = Z_k \) the coarsest estimate \( \hat{Y}_k \), which is taken as initialization for the following recursive steps executed for each scale \( k = K-1, \ldots, 0 \):

![Flowchart of the multiscale denoising process, starting from the noisy sinogram Z and resulting in the estimate \( \hat{Y} \) of the streak-free sinogram, both of size \( n \times m \). First, \( Z \) is vertically rescaled into a sinogram \( Z_0 \) of size \( n \times m_c \), with \( m_c \leq m \), through the binning operator \( B_r \). Then, by repeated horizontal binning \( B_h \), \( Z_0 \) is progressively downsampled into a series of sinograms \( Z_k = B_h^k(Z_{k+1}) \), \( k = 1, \ldots, K \), each of size \( 2^{-k}n \times m_c \). The coarsest scale noisy input \( Z_k^h \) is denoised with BM3D to produce \( \hat{Y}_k \). Then, recursively for \( k = K-1, \ldots, 0 \), the noisy input \( Z_k^h = Z_k \) is denoised with BM3D to produce \( \hat{Y}_k \). This definition of \( Z_k^h \) means that the coarse-scale horizontal components of \( Z_k \) are replaced by \( \hat{Y}_k \). The PSF for each scale is estimated as described in Section 3.3.2. The resulting estimate \( \hat{Y}_k \) of the horizontal debinning (size \( n \times m_c \)) similarly replaces the coarse-scale vertical components of \( Z \) to obtain the full-size estimate \( \hat{Y} \).](image-url)
Figure 6
Multiscale denoising of Fly. Left: the noisy sinogram $Z$. Center and right: three scales of the multiscale denoising process, each showing $Z_k$, $Z_k^*$, and $\hat{Y}_k$.

The full-size estimate $\hat{Y}$ is displayed in Fig. 12.

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

$$Z_k^* = Z_k - B_h^{-1}(B_h(Z_k)) + B_h^{-1}(\hat{Y}_{k+1})$$

$$= Z_k - B_h^{-1}(Z_{k+1} - \hat{Y}_{k+1}).$$

(2) Denoise $Z_k^*$ with BM3D to produce the estimate $\hat{Y}_k$.

The result $\hat{Y}_k$ of the last denoising step is the fully denoised estimate of $Z_0$. To produce the full-size estimate $\hat{Y}$ of $Y$, we replace the vertical coarse-scale components of $Z$ with those of $\hat{Y}_0$, similar to the Step (1) above,

$$\hat{Y} = Z - B_h^{-1}(B_h(Z_0)) + B_h^{-1}(\hat{Y}_0) = Z - B_h^{-1}(Z_0 - \hat{Y}_0).$$

Figure 6 illustrates the sinograms over the various stages of the multiscale denoising process.

3.3. Multiscale noise model

For BM3D denoising, we regard $Z_k^*$ of each scale $k$ as of the model (1), as

$$Z_k^* = B_h^k B_h(Y) + \eta_k^*,$$  

where

$$\eta_k^* = \begin{cases} \eta_k, & k = K, \\ \eta_k - B_h^{-1}(B_h(\eta_k)), & k < K, \end{cases}$$

and $\eta_k = B_h^k B_h(\eta)$.

This definition for $\eta_k^*$, $k < K$, follows from considering the coarser-scale estimate $\hat{Y}_k$ as perfectly denoised. Similar to (3), $\eta_k^*$ is treated as correlated noise with PSD

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

where $g_k^*$ is a correlation kernel and $|X_k| = [2^{-k}n]m$. As per (4), the kernel $g_k^*$ can be defined as

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

3.3.1. Multiscale PSD of white streak noise. Let $\eta_0 = B_h(\eta)$ be horizontally white and vertically constant streak noise like in Figure 1. Under this assumption for $\eta_0$, we have that all $\eta_k = B_h^k(\eta)$ for $0 \leq k \leq K$ are horizontally white and vertically constant, with variance $\text{var}(\eta_k) = 2^k \text{var}(\eta_0)$. The doubling of the variance with every horizontal binning follows from the noise whiteness, which means that each pixel of the coarser scale sinogram is a sum of two pixels with independent noise of equal variance. Therefore, disregarding the specific support size of their actual finite realizations, we can identify these stationary random fields as

$$\eta_k = 2^{k/2} \text{std}(\eta_0) \eta_k W,$$  

where $\eta_k W$ is a white streak noise like in Figure 1 with $\text{var}(\eta_k W) = 1$. We can hence rewrite (18) as

$$\eta_k^* = \begin{cases} 2^{k/2} \text{std}(\eta_0) \eta_k W, & k = K, \\ 2^{k/2} \text{std}(\eta_k W) (\eta_k W - B_h^{-1}(B_h(\eta_k W))), & k < K. \end{cases}$$

This together with (20) means that we can characterize $\eta_k^*$ through a kernel $g_k^*$ obtained by scaling either of two basic two kernels $g_c$ and $g_B$.

$$g_c = |X|^{-1/2} \mathcal{F}^{-1}[\text{std}(\mathcal{F}[\eta_k W])],$$

$$g_B = |X|^{-1/2} \mathcal{F}^{-1}[\text{std}(\mathcal{F}[\eta_k W - B_h^{-1}(B_h(\eta_k W))])],$$

by a factor

$$\zeta_k = 2^{k/2} \text{std}(\eta_0) = \text{std}(\eta_k),$$

as

$$g_k^* = \begin{cases} \zeta_k g_c, & k = K, \\ \zeta_k g_B, & k < K. \end{cases}$$

We note that although the equalities (23) formally depend on the realization size $|X|$, in practice this term only renormalizes the kernel with regard to the Fourier transforms; hence (23) can be computed for an arbitrary support.

The kernel $g_c$ is single-pixel wide and vertically constant like in Figure 1, with $\|g_c\|_\infty = \text{std}(\eta_0) = 1$. Example noise $\eta_k^*$, $k < K$, and the corresponding kernel $g_k^* = \zeta_k g_B$ and PSD $\Psi_k^*$ (19) are shown in Figure 7.

Estimation of $\zeta_0$. To estimate $\text{std}(\eta_0) = \zeta_0$, we first convolve $Z_0$ with a 2-D kernel $g_d = \phi \otimes \psi$ where $\phi$ is a 1-D column Gaussian function of length $m_c/2$ and standard deviation $m_c/12$ and $\psi$ is a horizontal high-pass Daubechies wavelet ‘db3’ of length 6, hence convolution with $g_d$ realizes low-pass filtering in the vertical and high-pass filtering in the horizontal. Thus, compared with $Z_0$, $Z_0 \otimes g_d$ offers a lower signal-to-noise ratio (SNR), which facilitates the estimation of noise statistics; an example of $Z_0$ and the corresponding $Z_0 \otimes g_d$ are shown in Figure 8 (top). One can compute an estimate of the standard deviation of $\eta_0 \otimes g_d$ via its median absolute deviation (Hampel, 1974).
Example of noise $\eta_0$ of $Z_0^*$, $k < K$, the correlation kernel $g^* = g_0 g_d$ where $g_d$ is produced by $B$ and $B^{-1}$, and the corresponding PSD $\Psi_g$. For the kernel, 50% gray pixel color in the figure corresponds to value 0 in the data; for the PSD, black is 0. Note the missing low frequencies at the center of the PSD, and the higher-frequency nature of $\eta_0^*$ compared with the white streak noise $\eta$ in Figure 1.

$$\text{std}(\eta_0 \otimes g_d) = 1.4826 \text{smed}\left(\frac{1}{N_0} \sum_{x_0} \left| Z_0 \otimes g_d - \text{smed}(Z_0 \otimes g_d) \right|\right).$$

(26)

where smed denotes the sample median and the factor 1.4826 calibrates the estimate with respect to a normal distribution of the noise. As $\text{std}(\eta_0 \otimes g_d) = \|g_0 g_d\|_2$, an estimate $\hat{\sigma}_0$ of $\sigma_0$ can be obtained through

$$\hat{\sigma}_0 = \|g_0 g_d\|_2^{-1} \text{std}(\eta_0 \otimes g_d).$$

(27)

3.3.2. Adapting the model to non-white streak noise. The above model (21)–(25) assumes that the streak noise $\eta_c$ is horizontally white and stationary; however, real streak noise is never exactly white across the displacement, and may thus have significant differences in noise power between scales.

To adapt to these deviations from the model, we relax the definition (24) of $\xi_k$ and allow the scaling parameter $\xi_k \geq 0$ to vary with each scale $k$, while assuming the kernels as in (25) for simplicity. In this way, to adaptively model the PSD (19), we require only the estimation of $\xi_k$ on each $Z_k^*$.

Estimation of $\xi_k$, $k \geq 0$. For $k = K$, $\xi_k$ can be estimated from $Z_K = Z_k^*$ by trivial substitutions of 0 with $K$ in (26) and (27). Although also for $k < K$ one could estimate $\xi_k$ similarly from $Z_k$, a more accurate estimate can be obtained using $Z_k^*$ as this leverages the denoising of the coarser scales and thus $Z_k^* \otimes g_d$ offers an even lower SNR than $Z_k \otimes g_d$. An example of $Z_k^*$ and the corresponding $Z_k^* \otimes g_d$ are shown in Figure 8 (bottom). Similar to (26) and for any $K \geq k \geq 0$, the standard deviation of the noise in $Z_k^* \otimes g_d$ can be estimated as

$$\text{std}(\eta_k^* \otimes g_d) = 1.4826 \text{smed}\left(\frac{1}{N_k} \sum_{x_k} \left| Z_k^* \otimes g_d - \text{smed}(Z_k^* \otimes g_d) \right|\right).$$

(28)

Noting that $\text{std}(\eta_k^* \otimes g_d) = \|g_k^* \otimes g_d\|_2$, we then estimate $\xi_k$ as

$$\hat{\xi}_k = \begin{cases} \|g_k^* \otimes g_d\|_2^{-1} \text{std}(\eta_k^* \otimes g_d), & k = K, \\ \|g_k \otimes g_d\|_2^{-1} \text{std}(\eta_k \otimes g_d), & k < K. \end{cases}$$

3.3.3. Horizontal nonstationarity of $\eta_c$. 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 a constant $\xi_k$ for all spatial positions without either oversmoothing or leaving noise artifacts in some areas. To adapt to horizontal nonstationarity, we further relax the model allowing $\xi_k$ to vary within each scale $k$.

In particular, before noise estimation and denoising, we split $Z_k^*$ and $Z_k \otimes g_d$ into overlapping, full-height segments. We apply BM3D separately on each segment of $Z_k^*$, using a PSD scaled by $\xi_k$; estimated on the corresponding segment of $Z_k^* \otimes g_d$, i.e., we consider each segment as a separate noisy image $z$ with a corresponding $\Psi$. After denoising, the segment estimates produced by BM3D are recombined with a windowing function to form the full estimate $Y_k$.

3.4. 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 what are reasonably produced by the distribution of $\eta_c$ and therefore require a specific pre-processing. To this end, after applying the bright-field and before the multi-scale denoising process, we run a simple procedure on $P_{\text{log}}$ which aims to detect and attenuate only the most extreme streaks. First, we calculate the median across the angular dimension of the 3-D stack of projections as

$$\hat{P} = \text{smed}(P_{\text{log}}),$$

resulting in a 2-D map in which the streaks present as pixels extremely brighter or darker than their surroundings. To detect extreme outliers, for each coordinate $x$ representing a single pixel of the detector and hence of $\hat{P}$, we fit a bivariate
The complete denoising process, requiring as inputs the noisy projections $P_{\text{raw}}$ and the bright- and dark-fields $I_b, I_d$, and producing as the output an estimate of the streak-free stack of projections composed of sinogram estimates $\hat{Y}$.

cubic polynomial $\varphi_x$ to a window $\hat{P}_x$ of $\hat{P}$ centered at $x$. Then, consistent with Gaussian modeling of $\eta_x$, we mark the center pixel $P(x)$ defective if $|P(x) - \varphi_x(x)| > 4 \text{std}(\hat{P}_x - \varphi_x)$, where std denotes the sample standard deviation; each marked pixel in $\hat{P}$ corresponds to a full column of the sinograms.

Each pixel of a defective column is replaced with the median of non-defective pixels within a 2-D window considering the displacement dimensions around it. We note that columns corrected in this way are unlikely to be completely free of streak noise; instead, the aim is to introduce less extreme pixel values that can be further denoised by the following applications of BM3D. In order not to overload the notation, we denote the output of this step as $P_{\text{log}}$ identical to its input.

The complete streak noise attenuation procedure is illustrated in Figure 9. The procedure is fully automatic, requiring as an input only the raw projections and the bright- and dark-fields.

4. Experiments

We test our pipeline on synthetic data as well as two real acquisitions displaying ring artifacts. As a comparison, we test our pipeline on synthetic data as well as two real acquisitions displaying ring artifacts freely available in TomoBank (De Carlo et al., 2018). The data contain 2000 projections with 2.2 μm pixels, 100 ms exposure time obtained at the Advanced Photon Source, 2-BM beamline; other experimental parameters are an X-ray energy of 60–70 keV, 10 μm LuAG scintillator, and sample-to-detector distance as 90 mm. The

| Peak | std(\eta) | SNR |
|------|-----------|-----|
| Noisy | Münch et al. (2009) | Vo et al. (2018) | Proposed |
| ∞ (π = 0) | 0.005 | 32.61 | 11.80 | 28.97 | 44.05 |
| 0.01 | 26.59 | 11.78 | 28.52 | 39.19 |
| 0.02 | 20.58 | 11.72 | 27.48 | 34.29 |
| 0.05 | 12.77 | 11.49 | 24.62 | 27.24 |
| 2560 | 0.005 | 32.66 | 11.85 | 28.32 | 38.41 |
| 0.01 | 26.64 | 11.82 | 27.89 | 35.90 |
| 0.02 | 20.63 | 11.77 | 26.90 | 32.63 |
| 0.05 | 12.82 | 11.54 | 24.21 | 26.67 |
| 1280 | 0.005 | 32.71 | 11.90 | 27.76 | 36.51 |
| 0.01 | 26.69 | 11.87 | 27.56 | 34.31 |
| 0.02 | 20.68 | 11.82 | 26.45 | 31.55 |
| 0.05 | 12.86 | 11.59 | 23.92 | 26.21 |

[640, 1280] (lower SNR) and generate a Poisson variate with mean and variance $A(1 + \eta_x)$, thus defining the Poisson noise $\pi$ as the difference between this Poisson variate and $A(1 + \eta_x)$. Furthermore, we include experiments with $\pi = 0$ (infinite SNR), thus resulting in a total of 12 combinations of Poisson and streak noise strengths. We do not simulate extreme streaks. As the underlying data consist of only a single sinogram, we have $Z = P_{\text{log}}$ and we consider $\ln(A + \pi/(1 + \eta_x))$ as the streak-free sinogram $\hat{Y}$. The results of the phantom experiments are collected in Table 1, and illustrated in Figures 10 and 11.

The Fly dataset consists of 180 projections with 50 s exposure (detector pixel size 27 μm, demagnified to 15.7 μm by cone-beam geometry) collected using a Sigray Prisma X-ray micro-tomography instrument at 34 kV; each sinogram is 512 pixels wide. The Fly contains both extreme streaks caused by defective detectors and approximately normally distributed streaks, although they are generally more intense towards the edges of the projections due to weak photon flux and Poisson noise affecting the bright-field. Thus, the Fly benefits greatly from both the extreme streak removal procedure of Section 3.4 and relaxing the stationarity assumption by performing PSD estimation and denoising in multiple parts for each sinogram as described in Section 3.3.3. The denoising results of two different sinograms are shown in Figure 12; the corresponding tomograms of the second sinogram of Figure 12 are shown in Figure 13.

We also test the algorithm on a soft tissue sample 00076 displaying severe ring artifacts freely available in TomoBank (De Carlo et al., 2018). The data contain 2000 projections with 2.2 μm pixels, 100 ms exposure time obtained at the Advanced Photon Source, 2-BM beamline; other experimental parameters are an X-ray energy of 60–70 keV, 10 μm LuAG scintillator, and sample-to-detector distance as 90 mm. The
sinograms are 2560 pixels in width. Included are ten samples for bright- and dark-fields, which are averaged to obtain a single bright-field and dark-field. The denoising results of a single sinogram are shown in Figure 14, and the corresponding tomograms in Figure 15.

4.1. Implementation details

We use the BM3D implementation for Python (available from the PyPI package `bm3d`) with the ‘refilter’ profile and input PSD estimated as described in Section 3.3.2.

For the multiscale denoising procedure, we performed vertical binning with $m_v = \lceil m/64 \rceil \approx 64$ pixels; this value, being slightly larger than the height of the BM3D search neighborhood of Section 2.2.1 (39 × 39 pixels), allows our method to attenuate streaks which change slowly across the angle – a larger value of $m_v$ might be used to deal with streaks featuring faster variation. The number of horizontal scales $K$ for each sinogram was set as $K = \lceil \log_2(n/40) \rceil$, which gives $K=3$ for Fly and the Shepp–Logan phantom, and $K=6$ for 00076; these values offer a compromise between denoising wide streaks versus preserving low-frequency signal components – larger values of $K$ not only result in $Z_K$ narrower than the BM3D search neighborhood but also in extremely coarse scales that naturally feature a very high SNR that may lead to overestimating $\hat{Z}$ and hence to oversmoothing. For the localized processing of each scale (Section 3.3.3), we estimate $\hat{\kappa}$, $K \geq k \geq 0$ and apply BM3D denoising on 39-pixel wide segments, following the width of BM3D search neighborhood. For the attenuation of extreme streaks, we used a 19 × 19 pixels window.

To consider the computational cost, we note that the full denoising process of Fly (181 × 512 × 512 pixels) run single-threaded on an AMD Ryzen 7 1700 processor takes about one hour, mostly due to the BM3D denoising in CPU. A highly parallel GPU-based implementation is expected to reduce this run time to the scale of seconds (Davy & Ehret, 2020). The complexity of BM3D is linear with the number of pixels in the

Figure 10
Left: comparison of sinograms after different denoising procedures on the Shepp–Logan phantom with noise as in (14) with std$(\eta) = 0.02$ and signal peak 2560. Top to bottom: $Y = \ln[A + \pi/(1 + \eta)]$; $Z$: Münch et al. (2009); Vo et al. (2018); proposed procedure based on BM3D denoising. Right: corresponding estimation errors. Note how both of the comparison methods create strong artifacts around the areas with the highest contrast, as pointed by the arrows. These artifacts are not present in either the noisy image $Z$ or the BM3D-based estimate.
noisy image. Thus, the computation time is directly proportional to the sinogram height after vertical binning. As each iteration of the horizontal multiscale denoising halves the number of pixels, the computational cost of BM3D for an extra iteration $k > 0$ is then a $1/2^k$th of the cost of $k = 0$, the total for all $k = 0; \ldots; K$ being at most twice that of single scale denoising of $Z_0$.

The correlation kernels $g_c$ and $g_B$ do not depend on the input or scale, and can thus be pre-computed. To compute $g_B$, we use directly the definition (23) through a Monte Carlo simulation of sample standard deviation in the Fourier domain. We note that, as the kernel is vertically constant, it suffices to perform this simulation in 1-D and repeat the kernel $m_v$ times in the vertical dimension.

For Münch et al. (2009) and Vo et al. (2018), we use implementations remove_stripe_fw and remove_all_stripe provided by the tomopy Python library of Gürsoy et al. (2014). The tomograms of each experiment are reconstructed using the xpack library (Marchesini et al., 2020).

5. Discussion and conclusions
We have presented a model for streak noise in the sinogram domain as locally stationary correlated noise additive in the logarithmic scale. Based on this model, we have described a BM3D-based multiscale denoising procedure removing streak noise, and, consequently, the tomogram ring artifacts. The use of the recently proposed variant (Mäkinen et al., 2020) of BM3D is crucial for this work, as we deal with long-range noise correlation which earlier BM3D designs could not handle satisfactorily.

Tested on both synthetic and real data, our denoising procedure achieves state-of-the-art performance in streak noise removal.
removal. Compared with the two popular streak removal algorithms (Münch et al., 2009; Vo et al., 2018), our procedure achieves superior results both visually and quantitatively in terms of signal-to-noise ratio. Although all tested algorithms manage to successfully remove most streak noise, both Münch et al. (2009) and Vo et al. (2018) tend to create large distortions especially where the intensities of the underlying sinogram columns vary significantly. This type of artifact hinders interpretation of the results and subsequent analysis such as segmentation. In comparison, the proposed algorithm offers 

Figure 13
Comparison of resulting tomograms after different denoising procedures on the second sinogram of Fly shown in Figure 12. Top to bottom: noisy reconstructed object; Münch et al. (2009); Vo et al. (2018); proposed procedure based on BM3D denoising. Although all methods achieve good results in removing the streaks, both Münch et al. (2009) and Vo et al. (2018) introduce strong shadows absent from the proposed estimate, as indicated by the arrows.

Figure 14
Comparison of sinograms after different denoising procedures on 00076. Top to bottom: noisy sinogram Z; Münch et al. (2009); Vo et al. (2018); proposed procedure based on BM3D denoising. Note how in the zoom-in the proposed method manages to remove streak noise without creating additional artifacts. Münch et al. (2009) creates a horizontal streak-like artifacts as seen in the middle of the zoom-in, not present in the noisy sinogram; Vo et al. (2018) does not fully denoise the sinogram.
similar or better streak removal without further distorting the sinogram.

The proposed multiscale framework, here using basic pixel binning and debinning, could be extended to more sophisticated scale decompositions, such as steerable pyramids, contourlets, or dual-tree wavelets (Kovacevic & Chebira, 2008). In this work, we have considered each sinogram as a separate input for BM3D for simplicity, but the same mechanism can be used for simultaneous filtering of the entire 3-D stack of sinograms. In this way, the similarities between consecutive sinograms of the stack could be utilized within the collaborative filter.

As the sinograms are commonly processed after a logarithmic transform, we have not discussed inversion of the logarithm needed for denoising. Although the exponential function is naturally the inverse of the logarithm, the non-linearity of the logarithm causes bias in a denoised sinogram if inverted this way. Hence, if the sinogram should be reverted back from the logarithmic domain after denoising, an exact unbiased inverse (Mäkitalo & Foi, 2010; Mäkitalo et al., 2010) should be used instead.

Acknowledgements

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

Funding information

Funding for this research was provided by: Academy of Finland (grant No. 310779); Sigray Inc. 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. (2016). IEEE Signal Process. Lett. 23, 1086–1090.
Boas, F. E. & Fleischmann, D. (2012). Imaging Med. 4, 229–240.
Croton, L. C., Ruben, G., Morgan, K. S., Paganin, D. M. & Kitchen, M. J. (2019). Opt. Express, 27, 14231–14245.
Dabov, K., Foi, A., Katkovnik, V. & Egiazarian, K. (2008). Proc. SPIE, 6812, 681207.
Davison, D., Fröjdth, C., O’Shea, V., Nilsson, H.-E. & Rahman, M. (2003). Nucl. Instrum. Methods Phys. Res. A, 509, 146–150.
Davy, A. & Ehret, T. (2020). J. Real-Time Image Process. https://doi.org/10.1007/s11554-020-00945-4.
De Carlo, F., Gürsöy, 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.
Gürsöy, 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, Monographs on the Physics and Chemistry of Materials, pp. 141–160.
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.
Kovacevic, J. & Chebira, A. (2008). An Introduction to Frames. Now Publishers Inc.
Mäkinnen, Y., Azzari, L. & Foi, A. (2020). IEEE Trans. Image Process. 29, 8339–8354.
Mäkitalo, M. & Foi, A. (2011). IEEE Trans. Image Process. 20, 99–109.
Mäkitalo, M., Foi, A., Fevralev, D. & Lukin, V. (2010). *Proceedings of the 2010 International Conference on Mathematical Methods in Electromagnetic Theory (MMET 2010)*, 6–8 September 2010, Kyiv, Ukraine, pp. 53–56.

Marchesini, S., Trivedi, A., Enfedaque, P., Perciano, T. & Parkinson, D. (2020). *Proceedings of the 20th International Conference on Computational Science (ICCS 2020)*, edited by V. V. Krzhizhanovskaya, G. Zavodszy, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos and J. Teixeira, 3–5 June 2020, Amsterdam, The Netherlands, pp. 248–261. Cham: Springer International Publishing (https://doi.org/10.1007/978-3-030-50371-0_18).

Münch, B., Trtik, P., Marone, F. & Stampanoni, M. (2009). *Opt. Express*, **17**, 8567–8591.

Paleo, P. & Mironè, 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.

Swinehart, D. F. (1962). *J. Chem. Educ.* **39**, 333.

Vidal, F. P., Letang, 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.