

# Generalized Titarenko's algorithm for ring artefacts reduction

E. X. Miqueles,<sup>a\*</sup> J. Rinkel,<sup>a</sup> F. O'Dowd<sup>a</sup> and J. S. V. Bermúdez<sup>b</sup><sup>a</sup>CNPEM/LNLS, Brazilian Synchrotron Light Laboratory, Campinas, São Paulo, Brazil, and<sup>b</sup>Department of Physics, Universidad Nacional de Colombia, Bogotá, Colombia.

\*E-mail: edu.miqueles@gmail.com

A fast algorithm for ring artefact reduction in high-resolution micro-tomography with synchrotron radiation is presented. The new method is a generalization of the one proposed by Titarenko and collaborators, with a complete sinogram restoration prior to reconstruction with classical algorithms. The generalized algorithm can be performed in linear time and is easy to implement. Compared with the original approach, with an explicit solution, this approach is fast through the use of the conjugate gradient method. Also, low/high-resolution sinograms can be restored using higher/lower-order derivatives of the projections. Using different order for the derivative is an advantage over the classical Titarenko's approach. Several numerical results using the proposed method are provided, supporting our claims.

## 1. Introduction

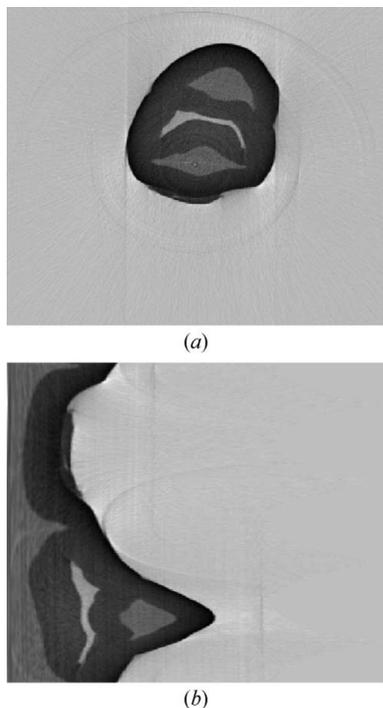
Micro-computed tomography essentially relies on the inversion of the Radon transform, which is a well known operator in inverse problems, especially in image reconstruction. Typically, a three-dimensional volume must be restored from a stacking of two-dimensional reconstructed images, where the  $i$ th feature image satisfies the following principle,

$$\begin{array}{c} \text{input} \\ \text{image} \end{array} \left| \begin{array}{c} \text{sinogram} \\ s \in V \end{array} \right. \longrightarrow \begin{array}{c} \text{output} \\ \text{image} \end{array} \left| \begin{array}{c} \text{feature} \\ f \in U \end{array} \right. \quad (1)$$

The most established method in tomography consists of converting X-rays into visible light with a scintillator and projecting onto a CCD using standard microscope optics. These projections can be used as input for computerized tomographic reconstruction. A common source of degradation in tomographic reconstruction is through the superimposition of so-called 'ring artifacts'. There are several known reasons for the generation of ring artifacts in reconstructed images. Primarily, these are due to defective pixels on the CCD detector or non-linear response of individual detector elements. Defective pixels typically result in a sharp single-pixel artifact unless several consecutive pixels are flawed. Larger artifacts are generally a result of imperfect or dusty scintillator crystals, which can easily exceed a single pixel. Other causes have recently been reported, such as monochromator vibration (see Titarenko *et al.*, 2010a) or thermal processes generating additional electrons in the CCD (see Titarenko *et al.*, 2009). For 180° rotation of the sample, semi-circle artifacts are generated; only for 360° scans will the full ring artifact be generated. As such, these imperfections are

independent of the projection angle and will appear uniformly for all angles in ring format, superimposed on the sample image. Regardless of the source of these artifacts, the resulting degradation of the image quality, obscuring relevant details, is detrimental to the computed tomography process.

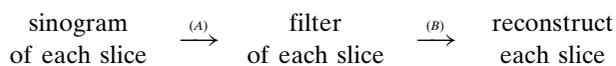
In order to illustrate the artefacts of the reconstructed image using the analytical inversion with filtered back-projection algorithm, we consider the example of Fig. 1, where a human tooth was introduced to a standard tomography using the Medipix detector. From the corrupted sinograms we clearly see the ring artifacts in the reconstructed image. Some authors in the literature used to filter the reconstructed image transforming to polar coordinates in order to identify the ring structures (see Sijbers & Postnov, 2004, and references therein). Indeed, this is because the polar transformation maps circles to straight lines, which can be detected using an appropriate filter. Converting from polar coordinates to Cartesian and *vice versa* can be done with a computational cost of  $O(n^2)$  where  $n$  is the size of the feature image. Unfortunately, changing to polar/Cartesian coordinates introduces additional noise in the reconstructed image. Polar transformation is a fast algorithm compared with the average computational cost of a reconstructed slice, which is  $O(n^2 \log n)$  (see Andersson, 2005). The main disadvantage of polar/Cartesian algorithms is the need to filter the high frequencies with several thresholds on the cut-off frequency, depending on some prior knowledge of the sample under investigation. Another technique for removal of stripe artefacts in the sinogram is the Fourier-Wavelet approach, presented by Münch *et al.* (2009). It is a fast technique, with linear computational cost  $O(n)$  based on a discrete wavelet



**Figure 1** Reconstruction of the feature image  $f$ , using a corrupted sinogram  $s(t, \theta)$ . (a)  $f$  in Cartesian coordinates. (b)  $f$  in polar coordinates. Transforming the reconstructed image to polar coordinates, in order to remove vertical stripes, was a popular approach for removing ring artefacts. See Sijbers & Postnov (2004).

decomposition of the sinogram. This is a more sophisticated technique which is able to remove horizontal and vertical stripes in many imaging problems.

A fast tomographic experiment, as carried out at the Brazilian Synchrotron Light Laboratory, goes through the following scheme,



Step (B) is usually done with a fast reconstruction algorithm, specially designed to handle a huge tomographic data set. In our case we use only analytical reconstruction through ‘pyhst’, a fast hierarchical inversion algorithm implemented for graphics processing unit (gpu) (see Mirone *et al.*, 2013). Step (A) is the main contribution of this manuscript, where a fast filtering of each sinogram has to be carried out, prior to reconstruction.

Titarenko *et al.* (2009, 2010a,b,c) introduced a fast algorithm for filtering the sinogram prior to reconstruction, based on the optimality conditions of a quadratic functional. Their method is direct and with a linear computational cost, *i.e.*  $O(n)$ . This has a great impact on building fast three-dimensional volumes for the tomographic data set. Therefore, even with more sophisticated inversion schemes, such as the expectation maximization algorithm (Helou *et al.*, 2014; De Pierro, 1995) or recent algorithms based on compressed sensing techniques (Candes *et al.*, 2006), the stacking process remains almost unaffected.

In this manuscript we present a generalization of Titarenko’s algorithm, still preserving the computational linear cost. Our method preserves the sinogram structure also introducing some radial smoothness on each projection, a similar approach also presented in their work (Titarenko *et al.*, 2010c). For real tomographic data, measured in the Brazilian Synchrotron Light Laboratory, our method is competitive with Titarenko’s original approach.

The manuscript is organized as follows. §2 presents the theoretical fundamentation of Titarenko’s original work further generalized in §3 and §4. Some numerical experiments are presented in §5, discussion in §6, and §7 gives the final conclusions.

## 2. Generalized variational approach

Theoretically, the pair  $\{s, f\}$  from equation (1) is related through the Radon transform (Deans, 1983), given below,

$$s(t, \theta) \equiv \mathcal{R}f(t, \theta) = \int_{\mathbb{R}^2} f(\mathbf{x}) \delta(t - \mathbf{x} \cdot \xi_\theta) \, d\mathbf{x}, \quad (2)$$

where  $\xi_\theta = (\cos \theta, \sin \theta)$  stands for the direction of the X-ray path,  $\mathbf{x}$  corresponds to a pixel of the feature image  $f \in U$ , and  $\mathbf{y} = (t, \theta)$  corresponds to a pixel of the sinogram  $s \in V$ . Typically,  $U$  and  $V$  are inner product spaces of square integrable functions, *e.g.*  $L^2$  or a Schwarz space. It can be shown that the linear operator  $\mathcal{R}: U \rightarrow V$  is invertible for an appropriate choice of  $U$  and  $V$ . A complete discussion about the action of  $\mathcal{R}$  can be found by Natterer & Wubbeling (2001).

This means that, if we are given  $s = \mathcal{R}f$ , the function  $f$  could be recovered using  $f = \mathcal{R}^{-1}s$  with an appropriate formula for  $\mathcal{R}^{-1}$ . In practice,  $\mathbf{y}$  lies in a discrete mesh of points  $\{t_1, t_2, \dots, t_R\} \times \{\theta_1, \theta_2, \dots, \theta_N\} \in \mathbf{N}$ , where  $N$  and  $R$  are assumed to be large, *e.g.*  $N \simeq 1000$  and  $R \simeq 3000$ . Therefore, a good estimate of the solution  $f$  could be found using the classical filtered backprojection algorithm, which is fast (mainly due to fast Fourier transforms) and approximates the inverse operator  $\mathcal{R}^{-1}$  (Gabor, 2009).

The approach of Titarenko *et al.* (2009, 2010a,b,c) searches for a regularized sinogram that does not contain horizontal stripes. These horizontal stripes are the main source for ring artefacts in the image reconstruction. A regularized sinogram is obtained through the minimization of an appropriate Tikhonov functional (Tikhonov 1963a,b; Yagola *et al.*, 2002). A brief mathematical description of Titarenko’s method is given in Appendix A. In this case, we are looking for a regularized sinogram, the solution of

$$\min_{s \in V} V(s) = \frac{1}{2} \int_{\mathbf{N}} [(\partial_t s)^2 + (\partial_{\theta\theta} s)^2] \, dt \, d\theta + \frac{\lambda}{2} \|s - m\|^2. \quad (3)$$

This was also explored in the work of Twomey (1963) for the solution of integral equations of the first kind. The solution is presented in Appendix A. For completeness, the pseudo-code of the original Titarenko’s ring suppression algorithm is given below,

Titarenko's Algorithm (TA):

*Input:* Corrupted sinogram  $\mathbf{M} \in \mathbb{R}^{R \times N}$

[1] Set  $\mathbf{e}_N = (1) \in \mathbb{R}^N$  and  $\lambda \in \mathbb{R}$

[2] Set matrix  $\mathbf{T}_1$  as in (42)

[3] Define explicitly  $\mathbf{A} \equiv (\mathbf{T}_1 + \lambda \mathbf{I}_d)^{-1}$ ; (see Titarenko *et al.* (2010a))

[4] Define  $\bar{\mathbf{m}} = (1/N) \mathbf{M} \mathbf{e}_N$

[5] Compute  $\mathbf{n}^* = -\mathbf{A} \mathbf{T}_1 \bar{\mathbf{m}}$

*Output:* Restored sinogram

$$\begin{cases} \text{for } (i, j) \in \mathbb{N}^{R \times N} \\ \quad \mathbf{S}_{j,i} = \mathbf{M}_{j,i} + \mathbf{n}_j^* \\ \text{end} \end{cases} \quad (4)$$

### 3. Generalized Titarenko's algorithm without angle dependency

Assuming that the deviation from  $s$  to  $m$  does not depend on the angle, *i.e.*

$$s(\mathbf{y}) = m(\mathbf{y}) + n(t), \quad (5)$$

and changing to the discrete case, we are now searching for  $\mathbf{n}^*$  such that

$$\mathbf{n}^* = \underset{\mathbf{n} \in \mathbb{R}^R}{\operatorname{argmin}} \left\{ V(\mathbf{n}) = \frac{1}{2} \sum_{i,j} \mathbb{F}_2[\mathbf{M}_{j,i} + \mathbf{n}_j]^2 + \lambda \frac{N}{2} \|\mathbf{n}\|^2 \right\}, \quad (6)$$

where  $\mathbb{F}_2$  is a second-order finite difference operator given by

$$\mathbb{F}_2[\mathbf{P}_{j,i}] = \mathbf{P}_{j-1,i} - 2\mathbf{P}_{j,i} + \mathbf{P}_{j+1,i}. \quad (7)$$

Since  $V$  is quadratic, solving  $\nabla V(\mathbf{n}^*) = 0$  guarantees a solution of (6). To make the calculations easier, let  $V(\mathbf{n}) = D(\mathbf{n}) + B(\mathbf{n})$ , with  $D$  related to the finite difference operator. Denoting  $\mathbf{M}_i$  as the  $i$ th column of  $\mathbf{M} \in \mathbb{R}^{R \times N}$ , a straightforward calculation gives us

$$\nabla D(\mathbf{n}) = \sum_i \mathbf{T}_2[\mathbf{M}_i + \mathbf{n}], \quad \nabla B(\mathbf{n}) = \lambda N \mathbf{n}, \quad (8)$$

where  $\mathbf{T}_2$  is a pentadiagonal matrix,

$$\mathbf{T}_2: \mathbf{F} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ -2 & 5 & -4 & 1 & \dots & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & \dots & 0 & 0 & 0 \\ \vdots & & & \ddots & & \vdots & & \\ 0 & 0 & 0 & 0 & \dots & 6 & -4 & 1 \\ 0 & 0 & 0 & 0 & \dots & -4 & 5 & -2 \\ 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 \end{pmatrix}_{R-2 \times R}. \quad (9)$$

Therefore, the optimality criteria gives us the noise vector  $\mathbf{n}^*$  in the same fashion as in (40) (see Appendix A),

$$\begin{aligned} \nabla V(\mathbf{n}^*) = 0 &\Rightarrow \mathbf{T}_2(\sum_i \mathbf{M}_i) + N \mathbf{T}_2 \mathbf{n}^* + \lambda N \mathbf{n}^* = 0 \\ &\Rightarrow (\mathbf{T}_2 + \lambda \mathbf{I}_d) \mathbf{n}^* = -\mathbf{T}_2 \bar{\mathbf{m}}, \end{aligned} \quad (10)$$

with  $\bar{\mathbf{m}}$  the average projection, as defined in (41).

### 3.1. Further generalizations

It is a well known fact, especially in discrete differential equations, that finite difference operators are closely related to tridiagonal or pentadiagonal matrices. The structure of the matrix depends on the type of approximation used for the associated derivatives. In the present case, we are also approximating derivatives and we have exactly the same problem. The finite difference operator  $\mathbb{F}_1$  used by Titarenko *et al.* (2010a,c) (see Appendix A) was easily expanded to a finite difference operator  $\mathbb{F}_2$  when minimizing (3). Of course, this can be expanded even more.

Considering only the discrete case, it is easy to realise that

$$\mathbb{F}[\mathbf{P}_{j,i}] = \mathbf{f}_j^T \mathbf{P}_i, \quad \mathbf{P}_i \in \mathbb{R}^R \quad (11)$$

is the general form of the finite difference operator applied to the  $i$ th column of matrix  $\mathbf{P}$ , and  $\mathbf{f}_j$  is the vector associated with the linear functional  $\mathbb{F}$ , *i.e.*  $\mathbf{f}_j$  is the  $j$ th row of a matrix  $\mathbf{F}$ . In the case of  $\mathbb{F}_1$  and  $\mathbb{F}_2$  we have the following matrices,

$$\mathbf{T}_1: \mathbf{F} = \begin{pmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & & & \ddots & & \\ 0 & 0 & 0 & \dots & -1 & 1 \end{pmatrix}_{R-1 \times R} \quad (12)$$

and

$$\mathbf{T}_2: \mathbf{F} = \begin{pmatrix} 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & & \ddots & & \vdots & & \\ 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 \end{pmatrix}_{R-2 \times R}. \quad (13)$$

The number of rows of matrix  $\mathbf{F}$  strongly depends on the operator  $\mathbb{F}$ ; as a rule, the offset from the main diagonal indicates the number of rows of  $\mathbf{F}$ . As a consequence, one can easily verify that matrices  $\mathbf{T}_1$  and  $\mathbf{T}_2$  satisfy  $\mathbf{T}_1 = \mathbf{F}^T \mathbf{F}$  and  $\mathbf{T}_2 = \mathbf{F}^T \mathbf{F}$ .

Inspired by (3), (37) and (11), we want to solve the following generalized optimization problem,

$$\mathbf{n}^* = \underset{\mathbf{n} \in \mathbb{R}^R}{\operatorname{argmin}} \left\{ V(\mathbf{n}) = \underbrace{\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^R [\mathbf{f}_j^T (\mathbf{M}_i + \mathbf{n})]^2}_{D(\mathbf{n})} + \underbrace{\lambda \frac{N}{2} \|\mathbf{n}\|^2}_{B(\mathbf{n})} \right\}. \quad (14)$$

Using the same notation as in (8), and letting  $\mathbf{P}_i \equiv \mathbf{M}_i + \mathbf{n}$ , it is not difficult to show that

$$\frac{\partial D}{\partial \mathbf{n}_j} = \sum_i (\mathbf{f}_i^T \mathbf{P}_i) \mathbf{f}_{j,1} + \dots + (\mathbf{f}_R^T \mathbf{P}_i) \mathbf{f}_{j,R} \quad (15)$$

$$= \sum_i \mathbf{F}_j^T \begin{pmatrix} \mathbf{f}_1^T \mathbf{P}_i \\ \vdots \\ \mathbf{f}_R^T \mathbf{P}_i \end{pmatrix} = \sum_i \mathbf{F}_j^T \mathbf{F} \mathbf{P}_i \quad (16)$$

$$= \mathbf{F}_j^T \mathbf{F} \sum_i \mathbf{P}_i = \mathbf{F}_j^T \mathbf{F} \sum_i [\mathbf{M}_i + \mathbf{n}] \quad (17)$$

$$= N \mathbf{F}_j^T \mathbf{F} [\bar{\mathbf{m}} + \mathbf{n}] \quad (18)$$

with  $\mathbf{F}_j$  the  $j$ th row of  $\mathbf{F}$ . From (18) and  $\nabla B(\mathbf{n}) = \lambda \mathbf{N} \mathbf{n}$ , we must have

$$\begin{aligned} \nabla V(\mathbf{n}^*) = 0 &\Rightarrow \mathbf{F}^T \mathbf{F}(\bar{\mathbf{m}} + \mathbf{n}^*) + \lambda \mathbf{n}^* = 0 \\ &\Rightarrow (\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d) \mathbf{n}^* = -\mathbf{F}^T \mathbf{F} \bar{\mathbf{m}}. \end{aligned} \quad (19)$$

We refer to equation (19) as a generalization of Titarenko's algorithm, clearly including case (40) and (10). We can now choose a whole family of finite difference operators in order to build a ring suppression algorithm. The generalized approach is given below, where  $\otimes$  stands for the Kronecker product (Horn & Johnson, 1985).

Generalized Titarenko's Algorithm (GTA):

*Input:* Corrupted sinogram  $\mathbf{M} \in \mathbb{R}^{R \times N}$

- [1] Set  $\mathbf{e}_N = (1) \in \mathbb{R}^N$  and  $\lambda \in \mathbb{R}$
- [2] Set finite difference vector/matrix  $\{h, \mathbf{F}\}$
- [3] Define matrix  $\mathbf{A} = \mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d$
- [4] Define  $\bar{\mathbf{m}} = (1/N) \mathbf{M} \mathbf{e}_N$
- [5] Solve  $\mathbf{A} \mathbf{n}^* = -\mathbf{F}^T \mathbf{F} \bar{\mathbf{m}}$

*Output:* Restored sinogram as in (4)

$$\mathbf{S} = \mathbf{M} + \mathbf{n}^* \otimes \mathbf{e}_N^T. \quad (20)$$

### 3.2. Finite difference coefficients

We remark that matrix  $\mathbf{F}$  acting on the operator  $\mathbb{F}$  is a convolution matrix based on the kernel  $h$ . In fact, from (41) we have  $h = (1, -1) \in \mathbb{R}^2$  and  $h = (1, -2, 1) \in \mathbb{R}^3$  in (7). From Bengt (1988) and Mahesh (1998) we have a wide family of backward and forward finite difference coefficients, with different order of accuracy. Some coefficients are depicted in Table 1. The kernel vector is denoted  $h_{kj}$ , with  $k$  standing for the order of derivative and  $j$  for the corresponding accuracy.

### 3.3. Solving the optimality conditions (19)

We want to solve (12) in order to complete step [5] in the Generalized algorithm GTA. Taking  $\mathbf{F}$  as in (12), linear system (19) reduces to (40), which has analytical solution

$$\mathbf{n}^* = (\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d)^{-1} (-\mathbf{F}^T \mathbf{F} \bar{\mathbf{m}}) \quad (21)$$

where  $(\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d)^{-1}$  was explicitly given by Titarenko *et al.* (2010a). In the general case, a similar approach is not trivial. Huckle (1994) presents a Fourier-strategy for solving tridiagonal linear systems arising from elliptic partial differential equations. Although we are not restricted to partial differential equations, the matrix inversion presented in the original work of Titarenko *et al.* is highly connected to the work of Huckle.

It is easy to observe that matrix  $\mathbf{F}$  is rank deficient, *i.e.*  $\text{rank}(\mathbf{F}) = R - r$ , where  $r + 1$  is the length of the Kernel vector  $h$ , as presented in Table 1. For instance, in (40),  $\text{rank}(\mathbf{T}) = R - 1$  whereas  $\text{rank}(\mathbf{T}) = R - 2$  in (10). The relaxation parameter  $\lambda$  ensures that  $\text{rank}(\mathbf{T} + \lambda \mathbf{I}_d) = R$  so that  $V(\mathbf{n})$  attains a minimum.

Let us denote  $\mathbf{A} = \mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d$  and  $\mathbf{f} = -\mathbf{T} \bar{\mathbf{m}}$ . We propose a fast solution of  $\mathbf{A} \mathbf{n} = \mathbf{f}$  through the use of the conjugate

**Table 1**

Finite difference coefficients to compute matrix  $\mathbf{F}$ ; see text for details.

Derivative	Accuracy	Kernel
$\partial_t$	1	$h_{1,1} = (-1, 1)$
	2	$h_{1,2} = (-3/2, 2, -1/2)$
	3	$h_{1,3} = (-11/6, 3, -3/2, 1/3)$
$\partial_{tt}$	6	$h_{1,6} = (-49/20, 6, -15/2, 20/3, -15/4, 6/5, -1/6)$
	1	$h_{2,1} = (1, -2, 1)$
	2	$h_{2,2} = (2, -5, 4, -1)$
$\partial_{ttt}$	6	$h_{2,6} = (469/90, -223/10, 879/20, -949/18, 41, -201/10, 1019/180, -7/10)$
	1	$h_{3,1} = (-1, 3, -3, 1)$
	5	$h_{3,5} = (-967/120, 638/15, -3929/40, 389/3, -2545/24, 268/5, -1849/120, 29/15)$

gradient method (CGM). This is possible since  $\mathbf{A}$  is a symmetric and positive definite matrix.<sup>1</sup> The CGM iterations (see Bazaraa *et al.*, 2007) are particularly interesting for this linear system since  $\mathbf{A}$  is a convolution matrix. Indeed, the CGM strongly depends on matrix-vector product  $\mathbf{A} \mathbf{u}$ , for some  $\mathbf{u}$ , which is computed using fast Fourier transforms. In fact,

$$\mathbf{A} \mathbf{n} = \mathbf{F}^T \mathbf{F} \mathbf{n} + \lambda \mathbf{n} = (\mathbf{n} \star h) \star h + \lambda \mathbf{n}, \quad (22)$$

where  $h$  is the convolution kernel and  $\star$  is a symbol for the circular convolution.

### 3.4. Computational complexity

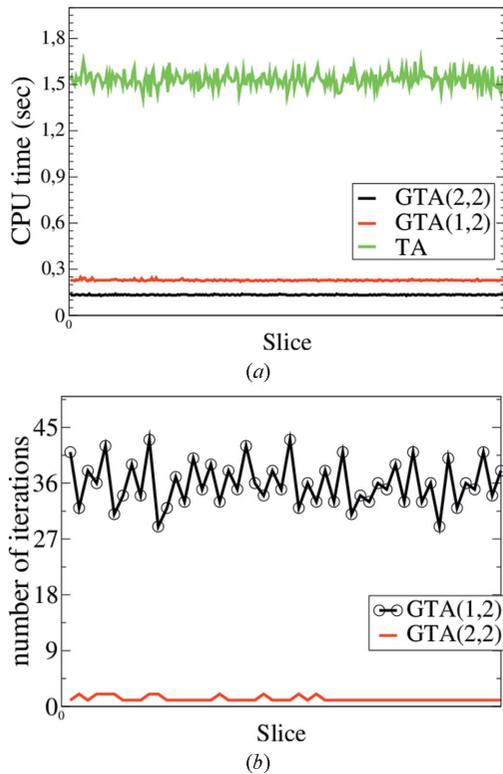
The solution of (19) for Titarenko's case, *i.e.*  $h = (1, -1)$  (see Table 1), could be computed really fast using a tridiagonal matrix algorithm (TDMA) (see Thomas, 1949). The computational complexity of TDMA is  $O(R)$ . If the analytical inverse of  $\mathbf{A}$  is computed previously, we have to perform the matrix-vector product for each slice, which has complexity  $O(R^2)$ . This is not the best programming choice, and it is preferable to use TDMA per iteration. A generic pseudocode for the three-dimensional reconstruction, using Titarenko's method, is given by

$$[\mathbf{3D}] : \begin{cases} \bullet \text{ for each slice}[k], \text{ do:} \\ \quad (a) \text{ Solve: } \mathbf{A} \mathbf{n} = \mathbf{f} \Rightarrow O(R) \\ \quad (b) \text{ Update: } \mathbf{S} = \mathbf{M} + \mathbf{n} \otimes \mathbf{e}_N^T \Rightarrow O(R^2) \\ \quad (c) \text{ Fast reconstruction: } \mathbf{i}_k = \mathbb{P}[\mathbf{S}] \Rightarrow O(R^2 \log R) \\ \bullet \text{ Stack } \{\mathbf{i}_k\} \end{cases}$$

For the generalized case, we still have complexity  $O(R)$  for step (a), using the conjugate gradient method. Therefore, the fast reconstruction strategy remains unaltered. Operator  $\mathbb{P}$  stands for the analytical inversion with filtered backprojection, or an iterative strategy.

Fig. 2(a) depicts the CPU time for each slice, using algorithms TA and GTA in a high-level programming language like Python. The reconstructions for this particular example are presented in more details in §5. Even with an analytical formulation for the inverse matrix  $\mathbf{A} = (\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I})$  in Titarenko's original approach, the CPU time is higher if compared with the solution of  $\mathbf{A} \mathbf{n} = \mathbf{f}$  through the CGM. Indeed,  $\mathbf{A}$  is a symmetric sparse band matrix, and the iterations of CGM

<sup>1</sup> Indeed, for all  $0 \neq \mathbf{u} \in \mathbb{R}^R$  we have  $\mathbf{u}^T \mathbf{A} \mathbf{u} = \|\mathbf{F} \mathbf{u}\|^2 + \lambda \|\mathbf{u}\|^2 > 0$ .



**Figure 2** Complexity of algorithms TA and GTA. (a) Comparison of the CPU time/slice used by Titarenko’s algorithm and his generalization using the conjugate gradient method. (b) Number of iterations used by the conjugate gradient method with order 1 and 2.

converge in at most  $R$  steps, where  $R$  is the number of rays in the sinogram. In fact, for a particular example with  $R = 2048$ , Fig. 2(b) shows the number of iterations for the convergence of the CGM using vector kernel  $h_{1,2}$  and  $h_{2,2}$  for the generalized Titarenko’s algorithm. Even with TA computing just once the dense matrix  $\mathbf{A}^{-1}$ , a matrix-product  $\mathbf{A}^{-1}\mathbf{f}$  has to be performed, for each slice, which has a quadratic computational cost. Using CGM is a more efficient strategy since the matrix-product  $\mathbf{A}\mathbf{n}$  is computed through convolution.

### 3.5. Reduction by blocks

The generalized algorithm (19) could be easily implemented by blocks of columns. This is important whenever the blank scan (the flat field) is corrected in between  $b \in \mathbb{N}$  projections. Also, such a strategy could be used to remove artefacts in the sinogram that are not eliminated using (19).

Let the sinogram be given by

$$\mathbf{S} = (\hat{\mathbf{S}}_1 \quad \hat{\mathbf{S}}_2 \quad \dots \quad \hat{\mathbf{S}}_b), \quad \hat{\mathbf{S}}_k \in \mathbb{R}^{R \times c}, \quad (23)$$

where  $c \in \mathbb{N}$  is such that  $N = cb$ . Each  $\hat{\mathbf{S}}_k$  is restored according to the generalized algorithm, i.e.

$$\hat{\mathbf{S}}_k = \hat{\mathbf{M}}_k + \mathbf{n} \otimes \mathbf{e}_c^T, \quad (24)$$

where  $\mathbf{n}$  is the solution of (19), and  $\hat{\mathbf{m}}$  is the average projection within the block, i.e.  $\hat{\mathbf{m}} = (1/c)\hat{\mathbf{M}}_k \mathbf{e}_c$ . In this case, matrix  $\mathbf{A} = \mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d$  remains the same for each block.

### 3.6. Geometric mean sinogram

Solving (19) through the conjugate gradient method could be made fast using convolution and the fast Fourier transform. Therefore, obtaining two solutions is a simple task, say  $\mathbf{n}_1$  and  $\mathbf{n}_2$ . In this case, we have two approximate restored sinograms  $\mathbf{S}_1$  and  $\mathbf{S}_2$ , respectively. Each one is given by (20). If we take  $\mathbf{n}_1$  as the solution of (19) with kernel  $h_{1,j}$ , we are minimizing (40). On the other hand, if we take  $\mathbf{n}_2$  as the solution of the same linear system, with kernel  $h_{2,j}$ , we are minimizing (41).

Our aim is to combine these two sinograms in order to have a better one. A convex combination of  $\mathbf{S}_{1,j}$  and  $\mathbf{S}_{2,j}$  does not fit to our needs since we have to choose a proper weight to favour first or second derivatives. Instead, we set a new sinogram, denoted by  $\mathbf{G}(\varepsilon)$ , whose entries  $(j, i)$  are defined by

$$\mathbf{G}_{j,i}(\varepsilon) = (\mathbf{N}_{j,i} \mathbf{P}_{j,i} + \varepsilon)^{1/2}, \quad (25)$$

for some previously defined tolerance  $\varepsilon > 0$ . If we take  $\varepsilon \equiv 0$  we have the geometric mean.<sup>2</sup> The choice of  $\varepsilon$  is such that  $\mathbf{G}$  is close to the arithmetic mean.

### 4. Generalized Titarenko’s algorithm with angle dependency

This case was already presented by Titarenko *et al.* (2011), where the corrupted sinogram also depends on the angle of rotation. In this case, hypothesis (5) is replaced by

$$s(t, \theta) = m(t, \theta) + n(t, \theta), \quad (26)$$

taking into account physical properties that depend on the angle of rotation  $\theta$ . After a similar discussion given in Appendix A, a functional is minimized and a linear equation is obtained. Without going into further detail, we claim that a similar analysis of §3 is easily extended for this case, using a finite difference operator  $\mathbb{F}$ . Here, we are looking for a correction matrix  $\mathbf{N} \in \mathbb{R}^{R \times N}$  such that  $\mathbf{S} = \mathbf{M} + \mathbf{N}$  is the updated sinogram, with  $\mathbf{M}$  the measured one.

In this case, the rows of the matrix  $\mathbf{N}$  are written as a linear combination of a previously known basis, i.e.  $\mathbf{N} = \mathbf{C}\mathbf{D}$  with  $\mathbf{C} \in \mathbb{R}^{R \times S}$  the unknown coefficients and  $\mathbf{D} \in \mathbb{R}^{S \times N}$  the basis. After computing the optimality conditions, the following linear system is obtained,

$$(\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d) \mathbf{C}_k = \mathbf{G}_k, \quad k = 1, 2, \dots, N, \quad (27)$$

where  $\{\mathbf{C}_k, \mathbf{G}_k\}$  are the  $k$ th columns of matrices  $\mathbf{C}$  and  $\mathbf{G}$ , respectively. Here, matrix  $\mathbf{G}$  is given by  $\mathbf{G} = \mathbf{B}\mathbf{D}^T \in \mathbb{R}^{R \times S}$  with  $\mathbf{B} = \mathbf{F}^T \mathbf{F} \mathbf{M} \in \mathbb{R}^{R \times N}$ . In a matricial form, the correction matrix  $\mathbf{N}$  is the one that satisfies  $(\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d) \mathbf{N} = \mathbf{B}\mathbf{D}^T \mathbf{D}$ . Generalization comes from the fact that we can choose a kernel vector  $h$ , see Table 1, in order to define the finite difference matrix  $\mathbf{F}$ . Taking  $h = (1, -1) \in \mathbb{R}^2$ , matrix  $(\mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d)$  becomes exactly the one proposed by Titarenko *et al.* (2010a), with an analytic inverse. Further details of this algorithm can be found in the work of Titarenko *et al.* (2011).

<sup>2</sup> This is always lower than or equal to the arithmetic mean.

Generalized Titarenko's algorithm with angle dependency  
[GTA( $\theta$ )]

*Input:* Corrupted sinogram  $\mathbf{M} \in \mathbb{R}^{R \times N}$ ,  $0 < S \in \mathbb{Z}$

[1] Define  $\mathbf{D} \in \mathbb{R}^{S \times N}$  orthonormal by rows

[2] Set  $\mathbf{B} = (\mathbf{F}^T \mathbf{F}) \mathbf{M}$

[3] Define matrix  $\mathbf{A} = \mathbf{F}^T \mathbf{F} + \lambda \mathbf{I}_d$

[4] Define matrix  $\mathbf{G} = \mathbf{B} \mathbf{D}^T \mathbf{D}$

[5] Solve  $\mathbf{A} \mathbf{N}_k = \mathbf{G}_k$ , for each  $k = 1, 2, \dots, N$

*Output:* Restored sinogram  $\mathbf{S} = \mathbf{M} + \mathbf{N}$

Compared with the GTA algorithm, GTA( $\theta$ ) demands a high computational cost. Loop [5] could be eliminated if the inverse of matrix  $\mathbf{A}$  is computed *a priori*, although this is only true for kernel vector  $h = (1, -1)$ ; in this case,  $\mathbf{N} = \mathbf{A}^{-1} \mathbf{B} \mathbf{D}^T \mathbf{D}$ . For different kernels, the explicit inverse matrix is no longer available, and the conjugate gradient method is used.

## 5. Numerical experiments

In 2011, work began in earnest on the design of an imaging beamline at the Brazilian Synchrotron Light Source (<http://lnls.br>). IMX, which has an electron source size of  $391 \mu\text{m} \times 97 \mu\text{m}$  and beam divergence of  $808 \mu\text{rad} \times 26 \mu\text{rad}$ , was built on a 2 T bending magnet of the 1.37 GeV storage. This beamline can operate in either white beam or monochromatic beam. The white-beam energy spectrum ranges from 4 keV to 15 keV, with a photon flux at the sample position of approximate  $10^{15}$  photons  $\text{s}^{-1}$ . Several scintillators are available at the beamline, the most utilized being the YAG:Ce (scintillator crystal), which has an emission wavelength of 550 nm and a photon output per 8 keV. The scintillator is mounted on a commercially available microscope system from Optique Peter<sup>3</sup> (Douissard *et al.*, 2012) coupled with a PCO.2000 CCD camera (Germany). The PCO.2000 is a high-resolution 14-bit CCD cooled camera with a quantum efficiency of 55% at 500 nm and has a full well capacity of 40000. The resulting array size of this detector system is  $2048 \times 2048$ , with pixel size from  $0.37 \mu\text{m}$  to  $3.7 \mu\text{m}$ . For a typical tomographic scan, 1000 images were acquired over  $180^\circ$ . A schematic of the experimental system for the acquisition of projected data is shown in Fig. 3.

### 5.1. Medipix detector

Hybrid silicon photon-counting detectors can meet the requirements of very high dynamic range and high detection efficiency specially needed for tomographic applications using synchrotron radiation. Compared with other families of hybrid pixel detectors, such as Pilatus (see Kraft *et al.*, 2009) and XPAD (see Pangaud *et al.*, 2008), the Medipix3RX readout chip enables a higher spatial resolution with a pixel size of  $55 \mu\text{m}$  by  $55 \mu\text{m}$ . Furthermore, this chip takes advantage of the advanced CMOS (complementary metal oxide semiconductor) technology to allow a high level of functionality in each pixel, enabling reconstruction of the charge generated by a photon detected on several neighbour pixels (charge



**Figure 3**  
Schematic of the imaging beamline IMX at LNLS.

summing mode) or allowing up to eight energy thresholds per cluster of four pixels (spectroscopic mode). Unfortunately, some characteristics of analog and digital circuits of the ASICs (application specific integrated circuit) are responsible for a non-linear relationship between measurement and incident flux. One of these characteristics is the shaping time directly related to the dead time of the system, which is responsible for pile-up effects (see Rinkel *et al.*, 2011). The pixel-to-pixel dispersion of these non-linear parameters cannot be corrected by simple flat-field normalization. In tomography, these dispersions are responsible for ring artefacts, in addition to global false estimation of the reconstructed attenuation coefficients.

To assess the robustness of the proposed algorithms for ring artefact correction, tomographic measurements were performed on the tomographic beamline of the Brazilian Synchrotron (LNLS) with a Medipix complementary detector. A Medipix3RX ASIC bump bonded to a  $300 \mu\text{m}$  silicon sensor was used in conventional single-pixel mode configured with single 24-bit counters and high-gain mode (see Gimenez *et al.*, 2011). The detector was read out using Medipix3 USB interfaces and Pixelman software (see Vykydal *et al.*, 2006).

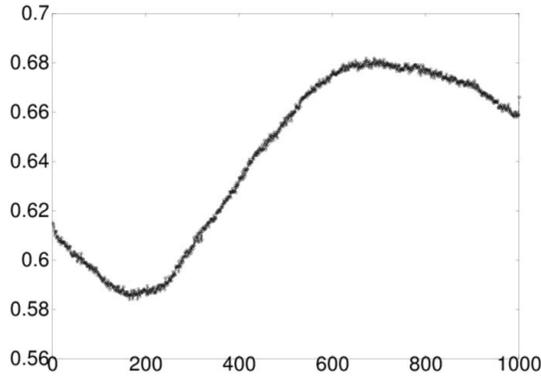
### 5.2. Examples

In the absence of noise, the theoretical sinogram  $s(t, \theta)$  satisfies the following mathematical relation (Helgason, 1980),

$$\begin{aligned} \frac{1}{2} \int_{\mathbb{R}} s(t, \theta) dt &= z(\theta) = \text{constant} \\ &= \frac{1}{2} \int_{\mathbb{R}^2} f(\mathbf{x}) d\mathbf{x} \quad \forall \theta. \end{aligned} \quad (28)$$

We refer to constant  $z$  as the *mass* of the sinogram, *i.e.* the *mean* remains constant for all possible values of  $\theta$ . For noisy data, (28) is no longer true. For the LNLS synchrotron data, as the beam current decays in time, function  $z(\theta)$  is far from being constant, with a monotonic behaviour.

<sup>3</sup> <http://www.optiquepeter.com/en/applications-synchrotron.php>.



**Figure 4** Function  $w = w(\theta)$ , as defined in equation (29), is the standard deviation of each measured projection, for a fixed sinogram. In this particular example the sinogram is presented in Fig. 5(a).

Since  $z(\theta)$  in (28) is not constant for real measured data  $m(\cdot, \theta)$ , the standard deviation (std) of each projection  $m(\cdot, \theta) \equiv m_\theta(\cdot)$  is a function  $w(\theta)$ , say

$$w(\theta) = \text{std}(m_\theta) \Rightarrow \mathbf{w} = [w(\theta_k)] \in \mathbb{R}^N. \quad (29)$$

Function  $w = w(\theta)$  is shown in Fig. 4 for some measured sinogram at the angle set  $\{\theta_1, \dots, \theta_{1000}\}$ . We take the regularization parameter  $\lambda$  [see (14)] as the standard deviation of  $\mathbf{w}$ , *i.e.*

$$\lambda = \text{std}(\mathbf{w}). \quad (30)$$

As is clear from (30), the value of  $\lambda$  depends on the measured sinogram  $\mathbf{M} \in \mathbb{R}^{R \times N}$ . As  $\mathbf{z} = [z(\theta_k)]$  is closer to a constant vector,  $\lambda$  is lower as we are dealing with a well behaved sinogram (at least in theory). The opposite remains true, *i.e.* the more distant  $\mathbf{z}$  is from a constant vector then the higher will be the penalty since we have an ill-behaved sinogram.

We use the following notation:

$\mathbf{S}_{k,j} \in \mathbb{R}^{R \times N}$  is the restored sinogram  $\mathbf{S}_{k,j} = \mathbf{M} + \mathbf{q} \otimes \mathbf{e}_N^T$ ; where  $\mathbf{q} \in \mathbb{R}^R$  is the solution of (19) with kernel  $h_{k,j}$  given in Table 1;

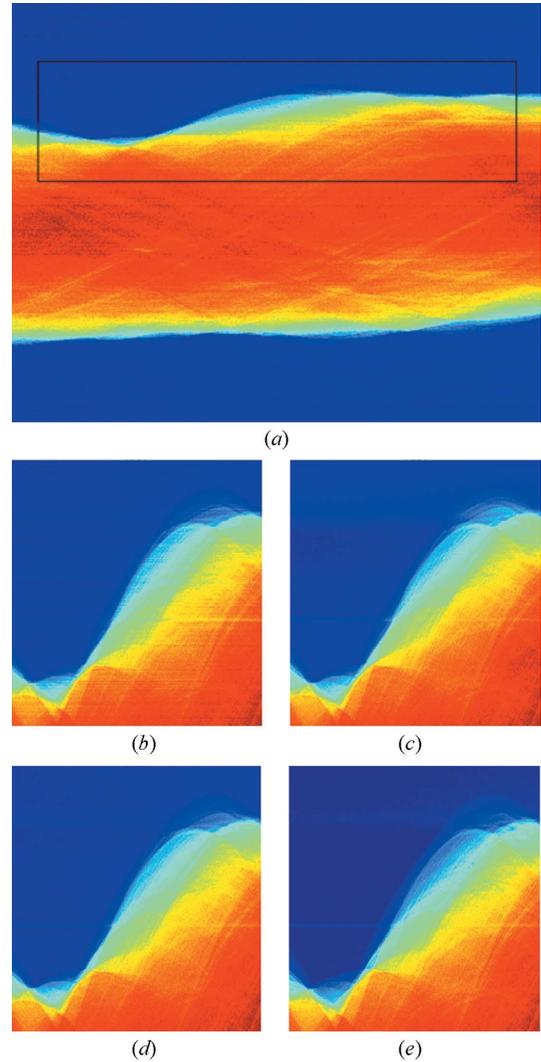
$\mathbf{G} = \mathbf{G}(\lambda) \in \mathbb{R}^{R \times N}$  is the geometric mean sinogram, with parameter  $\lambda$  defined in (30), using  $\mathbf{S}_{1,3}$  and  $\mathbf{S}_{2,2}$  as references [see (25)];

$\mathbf{M}$  is the measured sinogram, usually corrupted as in (5).

A measured sinogram at the LNLS,  $\mathbf{M}$ , is shown in Fig. 5(a). The region within the rectangle is presented in Fig. 5(b), where several horizontal stripes are visible. Each stripe generates a circle in the reconstruction technique.

The generalized Titarenko's suppression algorithm, with kernel  $h_{1,3}$ , see Table 1, gives the result  $\mathbf{S}_{1,3}$  in Fig. 5(c). Using a high-order finite difference, the horizontal stripes are clearly reduced. However, as can be noted in Fig. 5(c), there are two severe horizontal stripes remaining in the restored sinogram. The one on the top is due to the average projection [see vector  $\bar{\mathbf{m}}$  in (41)]. The second is probably due to dead/damaged pixels in the CCD camera.

Fig. 5(d) is the restored sinogram  $\mathbf{S}_{2,2}$  using the generalized algorithm with kernel  $h_{2,2}$ . The artefact corresponding to the first stripe is strongly reduced, although the second stripe still



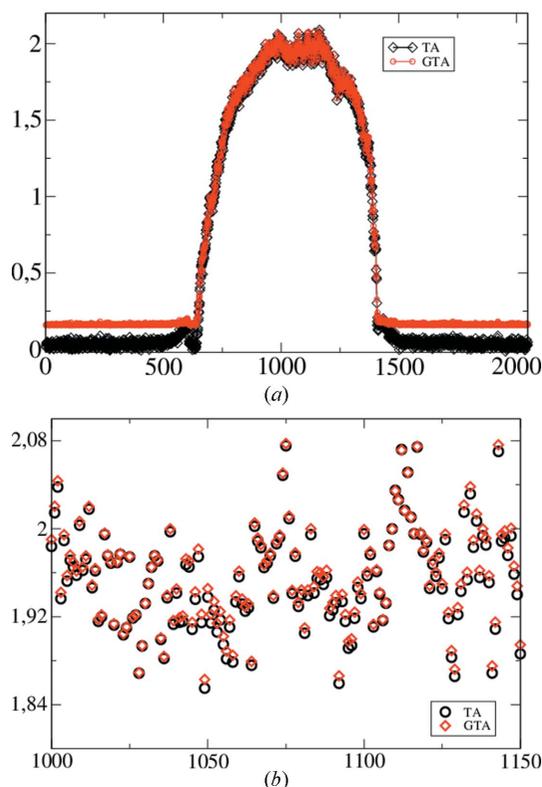
**Figure 5** Sinogram I (see text for details). (a) Original sinogram from the LNLS beamline. (b) Zoomed region from the original sinogram. (c) Sinogram  $\mathbf{S}_{1,3}$ . (d) Smoothed sinogram  $\mathbf{S}_{2,2}$ . (e) Geometric mean sinogram  $\mathbf{G}$ .

remains, as it is a major corruption in the data. Fig. 5(e) depicts the geometric sinogram  $\mathbf{G}$  using the combination of Figs. 5(d) and 5(c).

The projection at the angle  $\theta_{400} = \pi/10$  for the restored sinogram  $\mathbf{S}_{1,3}$  is presented in Fig. 6(a), compared with the same projection of the geometric sinogram  $\mathbf{G}$ . As expected, the geometric map overcomes the sinogram  $\mathbf{S}_{1,3}$ , as shown in the rectangle, zoomed in Fig. 6(b).

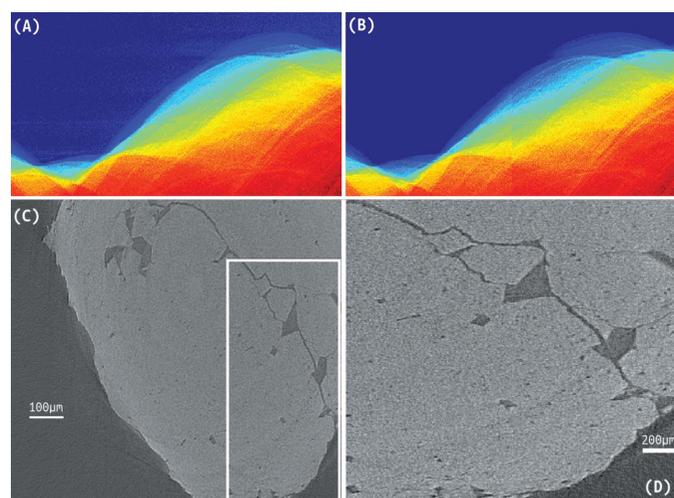
To remove the second horizontal stripe (see Fig. 5e), we adopt the block strategy presented in (24), using only two blocks. The restored sinograms [only the critical region of Fig. 5(a)], are shown in Figs. 7(a) and 7(b). The first was obtained using kernel  $h_{2,2}$  at each block, whereas the second was obtained using the geometric mean sinogram of each block, taking  $h_{2,2}$  and  $h_{1,3}$  as appropriate convolution kernels.

The first horizontal stripe present in the restored sinogram (see Fig. 5c) leads to the circular shadow outside the sample. The second horizontal stripe, uncorrected by first-order finite

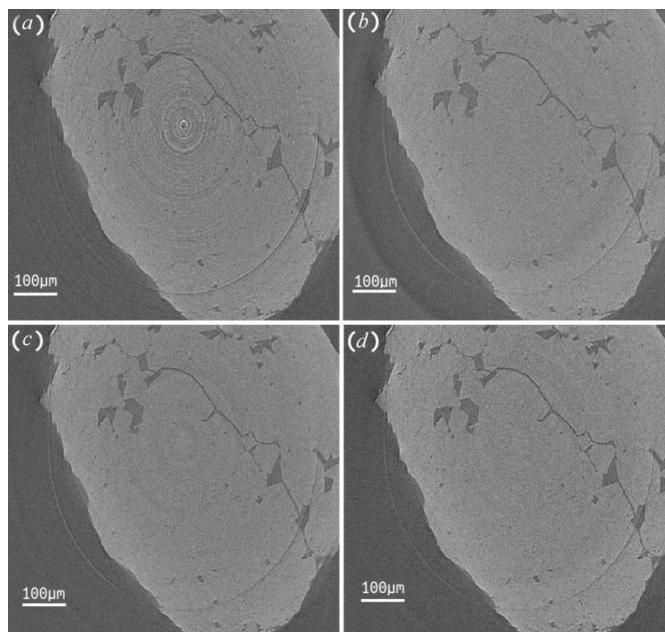


**Figure 6** Projections at  $\theta_{400} = \pi/1000$ . (a) Projection  $S_{1,3}(\cdot, 400)$  versus projection  $G(\cdot, 400)$ . (b) Zoomed region of part (a).

differences, leads to only one strong ring artefact in the feature image. Using second-order finite differences, *i.e.* using  $S_{2,2}$ , we obtain the reconstruction in Fig. 8(c). Since the first horizontal stripe was reduced, the shadow outside the sample, as in Fig. 8(b), was dramatically reduced. Nevertheless, since the second stripe still remains in the restored sinogram, the



**Figure 7** (a) Restoration using two blocks [see equation (24)] with kernel  $h_{2,2}$  at each block. (b) Geometric mean sinogram using two blocks. (c) Reconstruction with the sinogram of (b). (d) Magnified part of (c).

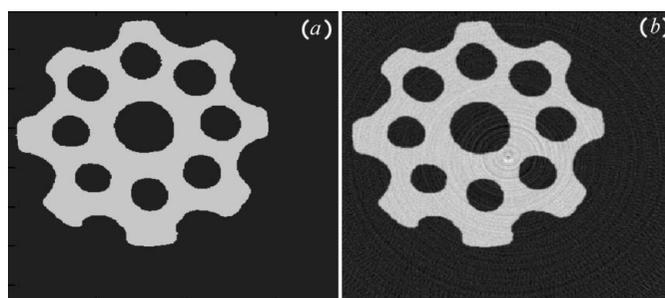


**Figure 8** FOV of reconstruction from sinograms presented in Fig. 5. See text for details.

reconstructed feature image presents one strong ring artefact. Through the geometric mean sinogram  $G$  (see Fig. 5e) we obtain the reconstruction in Fig. 8(d).

To reduce the leading ring artefact, still present in the reconstructed images [see Figs. 8(b), 8(c) and 8(d)], we use the filtered backprojection on the sinogram of Fig. 7(b), which is the geometric combination of  $S_{2,2}$  (by blocks) and  $S_{1,3}$  (by blocks). As depicted in Fig. 7(c), the leading ring artefact was completely eliminated, while still preserving the aspects of the image; as seen in Fig. 7(d) [zoomed part of Fig. 7(c)].

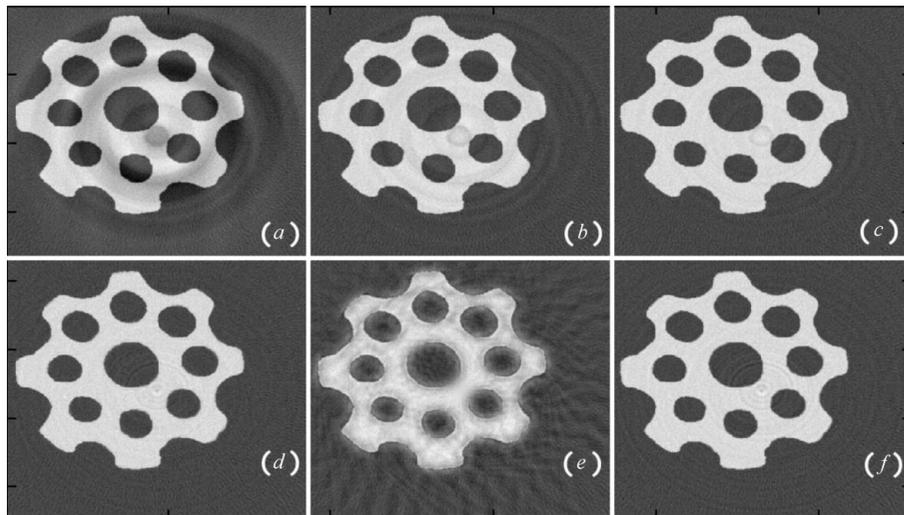
In order to illustrate the action of the derivative kernel  $h_{j,k}$ , see Table 1, we present a low-resolution simulated example. A  $327 \times 327$  image, representing a simulated micro-gear, is shown in Fig. 9(a). After adding noise to the  $527 \times 180$  sinogram (*i.e.* horizontal stripes), the filtered backprojection is applied to the corrupted sinogram, originating the reconstructed Fig. 9(b), where the ring artefacts are clearly visible.



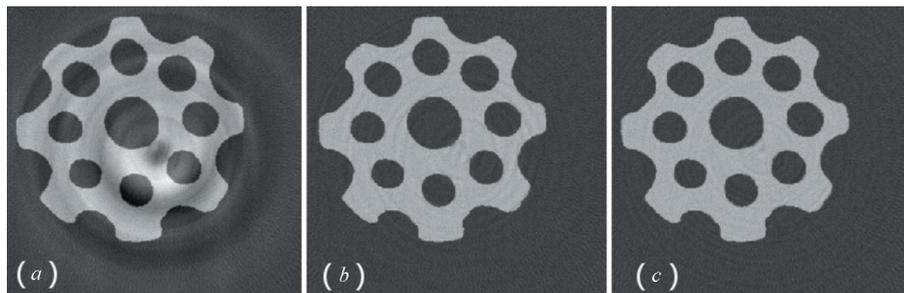
**Figure 9** Simulated example. (a) Image with resolution  $327 \times 327$ , representing an ideal micro-gear. (b) Reconstructed image (with strong ring artefacts) after adding horizontal stripes to the  $527 \times 180$  sinogram.

The generalized Titarenko’s algorithm (see §2) with different orders for the derivative was applied to the corrupted simulated gear data. The results are depicted in Fig. 10. Part (a) presents the result obtained using the original Titarenko’s algorithm (see §2 for an algorithmic description of TA). In this case the strong ring artefacts were reduced, but new artefacts are introduced, mainly because the sinogram has low resolution in the pixel axis. We remark that Titarenko’s original algorithm is obtained with the generalized algorithm and kernel vector  $h_{1,1}$ . Using the second-order derivative, we reduce the stripe effects through smoothness in the radial direction of the sinogram. The reconstructed image using second derivatives, *i.e.* using kernel vector  $h_{2,1}$  and  $h_{2,2}$ , are presented in part (b) and (c), respectively. Although the rings still remain in the reconstructed images, they were reduced if compared with (a). Using the block strategy and different kernel vectors, we obtain Figs. 10(d), 10(e) and 10(f).

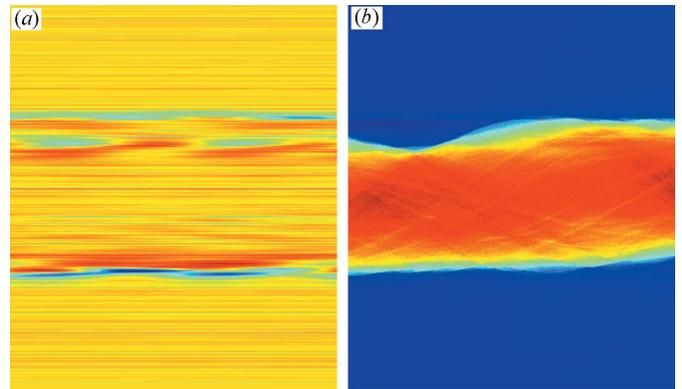
The Wavelet-FFT approach for ring removal, presented by Münch *et al.* (2009), is based on a discrete wavelet decomposition of the sinogram. Without going into further details, the algorithm to recover the sinogram  $\mathbf{S}$  is given by



**Figure 10** Simulated example, for TA and GTA. (a) TA for corrupted data. (b) GTA with  $h_{1,2}$ . (c) GTA with  $h_{2,2}$ . (d) GTA with  $h_{1,2}$  and 6 blocks. (e) GTA  $h_{2,2}$  and 60 blocks. (f) GTA with  $h_{3,1}$  and 6 blocks. See text for details.



**Figure 11** Reconstruction of the simulated micro-gear using the Wavelet-FFT correction and Daubechies wavelet DB25 (Münch *et al.*, 2009). (a) Parameters  $L = 7$ ,  $\sigma = 2$ . (b) Parameters  $L = 3$ ,  $\sigma = 3$ . (c) Parameters  $L = 2$ ,  $\sigma = 7$ .



**Figure 12** Sinogram restoration using  $\text{GTA}(\theta)$  with kernel  $h = (1, -2, 1)$  and  $S = 5$ . (a) Correction matrix  $\mathbf{N} \in \mathbb{R}^{R \times N}$ . (b) Updated sinogram  $\mathbf{S}$ . See §4 for details.

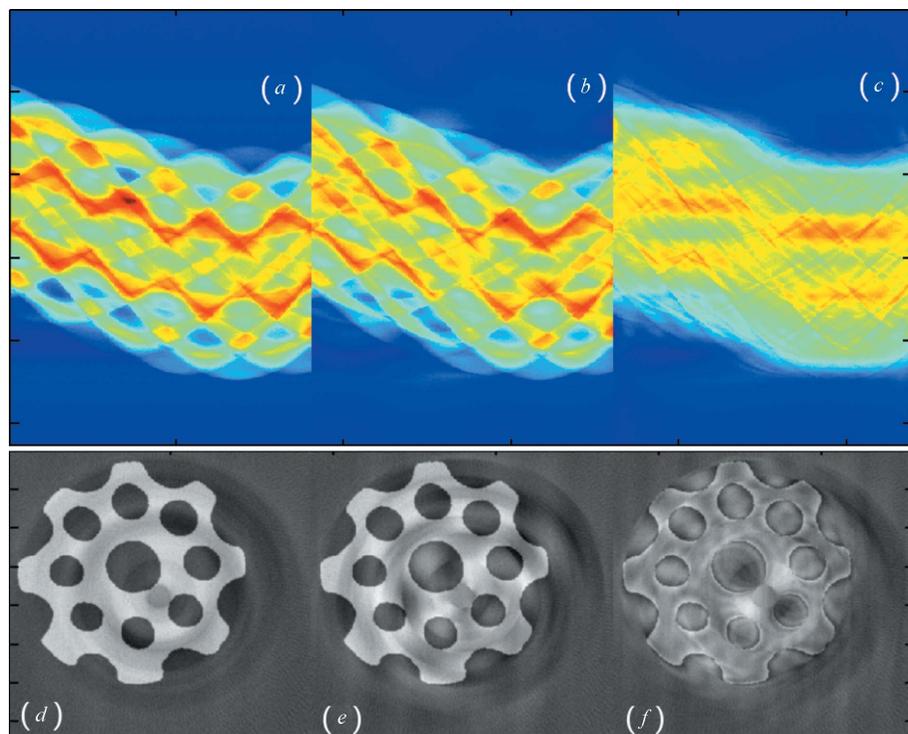
$$\mathbf{S} = \mathcal{W}[\mathbf{M}, L, \omega, \sigma], \quad (31)$$

where  $\mathbf{M}$  is the measured sinogram,  $L$  is the highest decomposition level using the wavelet type  $\omega$ , and  $\sigma$  is a damping factor, fixed for each decomposition level.

For example,  $\omega$  could represent a signature to Daubechies wavelets (*e.g.* DB25 or DB42) or to a Haar wavelet (DB1). Fig. 11 presents some reconstructions using DB25 with three different choices for  $(L, \sigma)$ . In comparison with the generalized Titarenko’s algorithm, we note that that the method is highly sensitive to the choice of  $(L, \sigma)$ .

The block strategy was applied to the corrupted gear data, originating the reconstructions in Figs. 10(d), 10(e) and 10(f). The restored sinograms were recovered using the generalized Titarenko’s approach using  $b = 6$  blocks [see (23)], *i.e.* using GTA in between 30 angles.

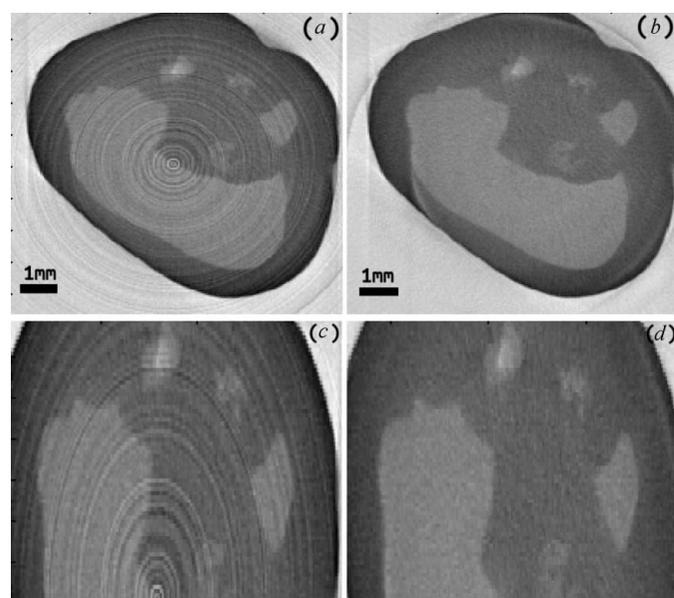
The sinogram of the rock, presented in Fig. 5, was restored using the generalized Titarenko’s algorithm with angle dependency [see §4]. Using kernel vector  $h = (1, -2, 1)$  from Table 1 we obtain the results in Fig. 12. The restored sinogram presented in Fig. 12(b) is similar to the one obtained using the classical Titarenko’s approach. The same strong stripe artefact remains, even with a smoothing kernel. In this example,  $\text{GTA}(\theta)$  is competitive with GTA, but with a higher computational cost. Fig. 12(a) shows the correction matrix  $\mathbf{N}$ , which is non-constant through columns. Fig. 13 depicts  $\text{GTA}(\theta)$  in the simulated micro-gear example, with different values of



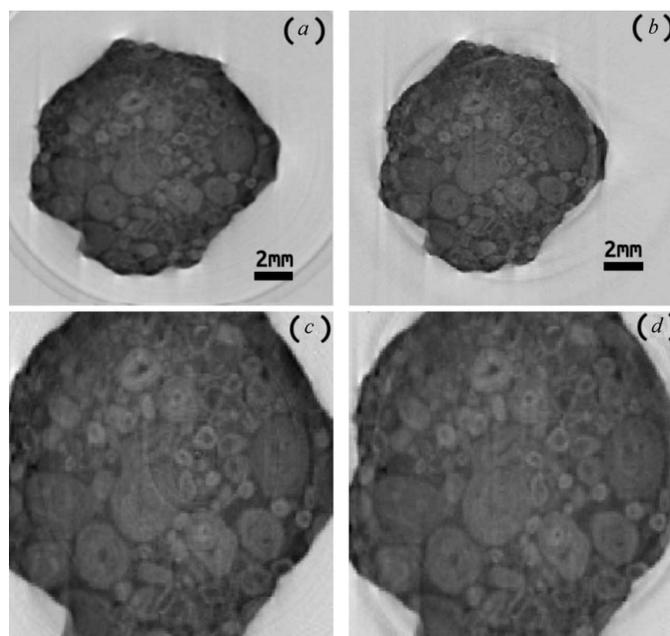
**Figure 13**  
Sinogram restoration using  $GTA(\theta)$  with kernel  $h = (1, -2, 1)$  and different values of  $S \in \mathbb{Z}$ . (a), (b), (c)  $S = 2, S = 5$  and  $S = 20$ , respectively. (d), (e), (f) Reconstruction of sinograms (a), (b) and (c), respectively.

parameter  $S$ , and with fixed parameter  $\lambda$ , as in (30).

Finally, some image reconstructions, with corrupted sinograms obtained with the Medipix detector, are shown in Figs. 14 and 15. As Medipix are low-resolution images, the same discussion of the micro-gear example of Fig. 9 applies



**Figure 14**  
Reconstruction of the human tooth of Fig. 1 with the Medipix detector. (a) Using sinogram  $S_{1,1}$ . (b) Using the geometric sinogram. (c) Enlarged image of (a). (d) Enlarged image of (b).



**Figure 15**  
Reconstruction of a rock sample with the Medipix detector. (a) Using the corrupted sinogram. (b) Using the geometric sinogram. (c) Enlarged image of (a). (d) Enlarged image of (b).

here. Noisy and low-resolution sinograms are better restored with a higher-order derivative, and using a block strategy.

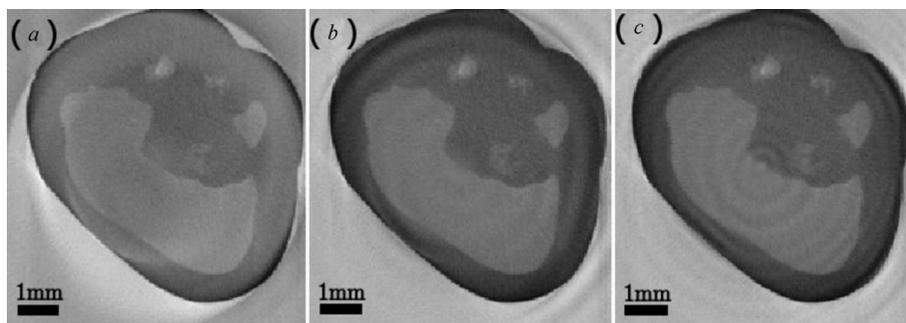
## 6. Discussion

As mentioned in the *Introduction*, the generation of ring artifacts in reconstructed images are usually related to non-linearity of the pixels' response. The model of sinogram corruption given by (5) accounts for the presence of strips. It assumes that the deviation from  $s$  to  $m$  does not depend on the angle. In other words, this deviation is supposed to be independent of the sample inhomogeneity.

Consider a pixel receiving a signal (intensity or number of photons) denoted by  $I_0$  without sample and  $I$  in the presence of the sample. The associated measurements are denoted  $Im_0$  and  $Im$ , respectively. The measured sinogram  $m$  and the original one  $s$  of (5) are obtained by calculating the measured and real X-ray attenuations of the object, respectively,

$$m(y) = \log(Im_0) - \log(Im), \quad (32)$$

$$s(y) = \log(I_0) - \log(I). \quad (33)$$



**Figure 16**  
Reconstruction of the human tooth of Fig. 1 with the Medipix detector, using the Wavelet-FFT correction and Daubechies wavelet DB25 (Münch *et al.*, 2009). (a) Parameters  $L = 7$ ,  $\sigma = 2$ . (b) Parameters  $L = 3$ ,  $\sigma = 3$ . (c) Parameters  $L = 2$ ,  $\sigma = 7$ .

We can deduce the value of the deviation term,

$$n(y) = \log(I_0/I_{m0}) - \log(I_m/I). \quad (34)$$

The second term of the last equation,  $-\log(I_m/I)$ , depends on the object composition and in this way on the projection angle for each pixel. Most non-linear effects, such as the pile-up phenomenon, tend to increase with the incident flux. This term is expected to become smaller with higher flux, resulting in better artefact correction for high-attenuation objects. This interpretation could explain the behaviour of the ring suppression algorithm on the images obtained with Medipix (see Figs. 14, 15 and 16). The ring suppression within the samples is really efficient for the tooth (Fig. 14) and for the rock (Fig. 15). This could be related to their high attenuations with maximum attenuation values over the whole sinograms of 8.3 and 6.2, respectively, for these samples. For the last sample, the ring suppression within the sample is only partial, which could be explained by this low attenuation (maximum attenuation value of 1.8 over the whole sinogram).

We are following the approach of Prince & Willsky (1990), where a quadratic functional is minimized. This was also the approach adopted by the classical Titarenko's approach. In fact, using a quadratic norm, we obtain an easy linear system to solve. This is one of the main goals of Titarenko's algorithm. Using a total variation norm (TV) penalty should be more appropriate, but this would introduce another type of algorithm, more difficult and not as fast as the one proposed. One of the most important aspects of imaging beamlines at a synchrotron is computational time, where GTA is appropriate.

## 7. Conclusions

We have presented a fast algorithm for ring artefact reduction in tomography by generalizing the approach proposed by Titarenko and collaborators. We provided several numerical and simulated results using the proposed method. The algorithm was tested on experimental data acquired with two different detectors: a scintillator mounted on a microscope system coupled with a CCD camera, and a counting detector based on a 300  $\mu\text{m}$  Si sensor connected to a Medipix3RX readout ASIC. For a Medipix detector (with low resolution),

we obtain good results using the generalized approach with high accuracy on the finite difference operator.

The main limitation of the method relies on the precision of the model of data corruption. Further work is focused on Titarenko's algorithm allowing the dependency of the correction term with the projection angle, which has already been carried out in their work of 2011 (Titarenko *et al.*, 2011), *i.e.* by considering  $s(t, \theta) = m(t, \theta) + n(t, \theta)$  instead of (5). We have applied such a technique in this manuscript, but the computational cost is

high. We have also compared our strategy, without the angle dependency, with the Wavelet-FFT algorithm presented by Münch *et al.* (2009). The wavelet approach is fast, with a linear computational cost  $O(N)$ . Although it is a very competitive method, it depends on three parameters,  $\{L, \omega, \sigma\}$ , making it difficult to choose the best configuration. In contrast with the present method, which depends on only one parameter, this is clearly a disadvantage of the wavelet algorithm.

Applying a TV penalty to the corrupted sinogram could be a good choice to restore the sinogram, penalizing smaller perturbation. This is an excellent starting point for a new study on rings artefacts reduction.

## APPENDIX A Titarenko's Algorithm

In real experiments, the measured sinogram  $m(\mathbf{y})$  is a corrupt version of the original one, that we denote by  $s(\mathbf{y})$ , *i.e.*

$$s(\mathbf{y}) = m(\mathbf{y}) + n(\mathbf{y}), \quad (35)$$

where  $n \in V$  is a sinogram to be determined, obeying some mathematical criteria (see Titarenko *et al.*, 2011). Following the classical approach of Rudin *et al.* (1992), the restored sinogram  $s \in V$  is the one that minimizes the TV, subject to constraints of mean, say  $\mathcal{M}$ , and standard deviation, given by  $\mathcal{S}$ ,

$$\left. \begin{array}{l} \min \frac{1}{2} \text{TV}(s) \\ \text{s.t. } \mathcal{M}(s) = \mathcal{M}(m) \\ \mathcal{S}(s) = \sigma \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \min \frac{1}{2} \text{TV}(s) + \frac{\lambda}{2} \|s - m\|^2 \\ \text{s.t. } s \in V \end{array} \right. \quad (36)$$

Here, the total variation norm is defined by  $\text{TV}(s) = \int_N \|\nabla s(\mathbf{y})\| \, d\mathbf{y}$ . The equivalence of the constrained and unconstrained optimization problems in (36) was previously shown by Chambolle (2004). The choice of the regularization parameter  $\lambda$  depends on the nature of the problem and could be estimated with an appropriate method (see Yagola *et al.*, 2002). A TV approach will penalize smaller perturbations on the corrupted sinogram, but in this case the optimality conditions are more difficult to solve.

The approach adopted by Titarenko *et al.* (2010a,b,c) searches for a solution that minimizes the quadratic functional

$$(1/2) \int_{\mathbf{N}} \|\nabla s(\mathbf{y})\|^2 d\mathbf{y} + (\lambda/2) \|s - m\|^2. \quad (37)$$

This approach was also explored in the work of Prince & Willsky (1990, 2002), where the sinogram is restored through the minimization of the above functional, subject to the Ludwig–Helgason consistency conditions (Ludwig, 1966; Helgason, 1980).

Assuming that the deviation from  $s$  to  $m$  does not depend on the angle, *i.e.*

$$s(\mathbf{y}) = m(\mathbf{y}) + n(t), \quad (38)$$

we arrive at the minimization, in terms of the radial (or noise) function  $n = n(t)$ , *i.e.*

$$V(n) = (1/2) \int_{\mathbf{N}} (\partial_t m + \partial_t n)^2 dt d\theta + (\lambda/2) \|n\|^2. \quad (39)$$

Assumption (38) is motivated by the stripes along the  $\theta$ -axis on each sinogram.

In a discrete sense, the sinogram  $m$  is represented by matrix  $\mathbf{M} \in \mathbb{R}^{R \times N}$ , and we are looking for a noise vector  $\mathbf{n} \in \mathbb{R}^R$  such that  $V = V(\mathbf{n})$  is minimum, with the integral replaced by the sum and the derivative replaced by finite differences. Since we are minimizing a quadratic, the function attains a minimum where the gradient is zero, *i.e.*  $\nabla V(\mathbf{n}^*) = 0$ . The first-order conditions and a tridiagonal system must be solved to find  $\mathbf{n}^*$ . Titarenko’s algorithm now follows,

$$\mathbf{n}^* = \underset{\mathbf{n} \in \mathbb{R}^R}{\operatorname{argmin}} \left[ V(\mathbf{n}) = \frac{1}{2} \sum_{i,j} \mathbb{F}_1(\mathbf{M}_{j,i} + \mathbf{n}_j)^2 + \lambda \frac{N}{2} \sum_j \mathbf{n}_j^2 \right] \\ \Rightarrow (\mathbf{T}_1 + \lambda \mathbf{I}_d) \mathbf{n}^* = -\mathbf{T}_1 \bar{\mathbf{m}}, \quad (40)$$

with  $\mathbf{T}_1$  being a constant tridiagonal matrix,  $\mathbb{F}_1$  a first-order finite difference and  $\bar{\mathbf{m}}$  the average of the projections, *i.e.*

$$\mathbb{F}_1[\mathbf{P}_{j,i}] = \mathbf{P}_{j,i} - \mathbf{P}_{j+1,i}, \\ \bar{\mathbf{m}} = (1/N) \mathbf{M} \mathbf{e}_N \in \mathbb{R}^R, \quad (41) \\ \mathbf{e}_N = (1) \in \mathbb{R}^N.$$

Titarenko *et al.* (2010a) presented the exact analytical inverse of  $\mathbf{T}_1 + \lambda \mathbf{I}$  using properties of hyperbolic trigonometric functions. For completeness, matrix  $\mathbf{T}_1$  is a tridiagonal matrix, whose diagonal entries are given by

$$\mathbf{T}_1: \mathbf{F} = \begin{pmatrix} 1 & -1 & \dots & 0 & 0 \\ -1 & 2 & \dots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & 2 & -1 \\ 0 & 0 & \dots & -1 & 1 \end{pmatrix}_{R-1 \times R}. \quad (42)$$

## APPENDIX B Octave codes

The codes for ring suppression, as presented in this manuscript, are presented below, in Octave open-source programming language. Function `ring()` performs the correction by columns, while function `ring_blocks()` corrects by blocks of columns; see (20), (23) and (24). Auxiliary function `ringMatXvec()` performs the matrix-vector product (22) using the convolution function `conv()` implemented by Octave. Finally, function `ringCGM()` iterates using the Conjugate gradient method in order to solve linear system (21). Functions `kron()` (Kronecker product), `ones()` and `std()` are also implemented in Octave (see Eaton *et al.*, 2009).

By changing vector  $h$  at line 6 of functions `ring()` and `ring_blocks()`, we modify the order of the derivative, according to Table 1. Transcription of these codes to a high-level language such as Python could be easily done. We emphasize that the original Titarenko’s algorithm is obtained using `h = [1 -1]` at line 6 of function `ring()`. The same applies to function `ring_blocks()`.

Function `ring_a()` stands for the algorithm  $\text{GTA}(\theta)$  (see §4) with  $S$  as the input parameter; see Titarenko *et al.* (2011) for further details on matrices  $\mathbf{D}$  and value  $S \in \mathbb{Z}$ .

```
function new = ring(old);

alpha = 2 * std(std(old));
pp = sum(old') / size(old, 2);
N = size(old, 2);
h = [-1 2 -1];
f = -ringMatXvec(h, pp);
n = ringCGM(h, alpha, f);
new = old + kron(n, ones(1, N));
```

```
function new = ring_blocks(old, step);
```

```
alpha = 2 * std(std(old));
R = size(old, 1);
N = size(old, 2);
h = [-1 2 -1];
blocks = fix(N/step);
```

```
for k = 1: blocks,
    col = (k - 1) * step + 1: k * step;
    sino_block = old(:, col);
    pp = sum(sino_block') / size(sino_block, 2);
    f = -ringMatXvec(h, pp);
    q = ringCGM(h, alpha, f);
    new(:, col) = sino_block + diag(q) * ones(R, step);
end
```

```

function y = ringMatXvec(h, x);

s = conv(x, fliplr(h));
u = s(length(h):length(x));
y = conv(u, h);

function x = ringCGM(h, alpha, f)

x0 = zeros(size(f));
r = f - [ringMatXvec(h, x0) + alpha * x0];
w = -r;
z = ringMatXvec(h, w) + alpha * w;
a = (r' * w)/(w' * z);
x = x0 + a * w;
B = 0;

for i = 1:10^6
r = r - a * z;
    if (norm(r)<1e - 6)
        break;
    endif
    B = (r' * z)/(w' * z);
    w = -r + B * w;
    z = ringMatXvec(h, w) + alpha * w;
    a = (r' * w)/(w' * z);
    x = x + a * w;
end

function new = ring_a(old, S);

R = size(old, 1);
N = size(old, 2);
F = zeros(S, N);
i = 1: N;

for k = 1: fix(S/2),
    D(2 * k, :) = sqrt(2/N) * cos(pi * i * 2 * k/N);
    D(2 * k - 1, :) = sqrt(2/N) * sin(pi * i * 2 * k/N);
end
D(1, :) = sqrt(1/N) * ones(1, N);
h = [-1 2 -1];

for i = 1: N, B(:, i) = -ringMatXvec(h, old(:, i)); end

G = B * D' * D;
alpha = 2 * std(std(old));

for w = 1: N,
    new(:, w) = old(:, w) + ringCGM(h, alpha, G(:, w));
end

```

The authors gratefully acknowledge the comments by an anonymous referee. The authors are also grateful to Hugo H.

Slepicka for helping at the integration of the ring artefact code with Python programming language at the LNLS imaging beamline IMX.

## References

- Andersson, F. (2005). *SIAM J. Appl. Math.* **65**, 818–837.
- Bazaraa, M. S., Sherati, H. D. & Shetty, C. M. (2007). *Nonlinear Programming, Theory and Algorithms*, 3rd ed. New York: Wiley.
- Bengt, F. (1988). *Math. Comput.* **51**, 699–706.
- Candes, E. J., Romberg, J. & Tao, T. (2006). *IEEE Trans. Inf. Theory*, **52**, 489–509.
- Chambolle, A. (2004). *J. Math. Imaging Vis.* **20**, 89–97.
- Deans, S. (1983). *The Radon Transform and Some of Its Applications Deans*. New York: John Wiley and Sons.
- De Pierro, A. R. (1995). *IEEE Trans. Med. Imaging*, **14**, 132–137.
- Douissard, P. A., Cecilia, A., Rochet, X., Chapel, X., Martin, T., van de Kamp, T., Helfen, L., Baumbach, T., Luquot, L., Xiao, X., Meinhardt, J. & Rack, A. (2012). *J. Instrum.* **7**, P09016.
- Eaton, J. W., Bateman, D. & Hauberg, S. (2009). *GNU Octave, a high-level interactive language for numerical computations*, Version 3.0.1. CreateSpace Independent Publishing Platform.
- Gabor, G. T. (2009). *Fundamentals of Computerized Tomography: Image Reconstruction from Projections, Advances in Pattern Recognition*. Berlin: Springer.
- Gimenez, E. N., Ballabriga, R., Campbell, M., Horswell, I. & Llopart, X. (2011). *IEEE Trans. Nucl. Sci.* **58**, 323–332.
- Helgason, S. (1980). *The Radon Transform*. Boston: Birkhauser.
- Helou, E. S., Censor, Y., Chen, T. B., Chern, I. L., De Pierro, A. R., Jiang, M. & Lu, H. H. S. (2014). *Inverse Probl.* **30**, 055003.
- Horn, R. A. & Johnson, C. R. (1985). *Matrix Analysis*. Cambridge University Press.
- Huckle, T. (1994). *BIT Numer. Math.* **34**, 99–112.
- Kraft, P., Bergamaschi, A., Broennimann, C., Dinapoli, R., Eikenberry, E. F., Graafsma, H., Henrich, B., Johnson, I., Kobas, M., Mozzanica, A., Schlepzt, C. M. & Schmitt, B. (2009). *IEEE Trans. Nucl. Sci.* **56**, 758–764.
- Ludwig, D. (1966). *Commun. Pure Appl. Math.* **19**, 49–81.
- Mahesh, K. (1998). *J. Comput. Phys.* **145**, 332–358.
- Mirone, A., Gouillart, E., Brun, E., Tafforeau, P. & Kieffer, J. (2013). *PyHST2: a hybrid distributed code for high speed tomographic reconstruction with iterative reconstruction and a priori knowledge capabilities*, <http://arxiv.org/abs/1306.1392>.
- Münch, B., Trtik, P., Marone, F. & Stampanoni, M. (2009). *Opt. Express*, **17**, 8567–8591.
- Natterer, F. & Wubbeling, F. (2001). *Mathematical Methods in Image Reconstruction*. Philadelphia: Society for Industrial and Applied Mathematics.
- Pangaud, P., Basolo, S., Boudet, N., Berar, J.-F., Chantepie, B., Clemens, J.-C., Delpierre, P., Dinkespiler, B., Medjoubi, K., Hustache, S., Menouni, M. & Morel, C. (2008). *Nucl. Instrum. Methods Phys. Res. A*, **591**, 159–162.
- Prince, J. L. & Willsky, A. S. (1990). *Opt. Eng.* **29**, 535–544.
- Prince, J. L. & Willsky, A. S. (2002). *IEEE Trans. Image Process.* **2**, 401–416.
- Rinkel, J., Brambilla, A., Dinten, J.-M. & Mougél, F. (2011). Patent WO2011064381.
- Rudin, L. I., Osher, S. & Fatemi, E. (1992). *Physica D*, **60**, 259–268.
- Sijbers, J. & Postnov, A. (2004). *Phys. Med. Biol.* **49**, 247–253.
- Thomas, L. H. (1949). *Elliptic Problems in Linear Differential Equations over a Network*, Watson Science Computer Laboratory Report, Columbia University, New York, USA.
- Tikhonov, A. N. (1963a). *Sov. Math. Dokl.* **4**, 1624–1627.
- Tikhonov, A. N. (1963b). *Sov. Math. Dokl.* **5**, 1035–1038.

- Titarenko, S., Titarenko, V., Kyrieleis, A. & Withers, P. J. (2009). *Appl. Phys. Lett.* **95**, 071113.
- Titarenko, S., Titarenko, V., Kyrieleis, A. & Withers, P. J. (2010c). *J. Synchrotron Rad.* **17**, 540–549.
- Titarenko, S., Titarenko, V., Kyrieleis, A., Withers, P. & De Carlo, F. (2011). *J. Synchrotron Rad.* **18**, 427–435.
- Titarenko, S., Withers, P. J. & Yagola, A. (2010a). *Appl. Math. Lett.* **23**, 1489–1495.
- Titarenko, S. & Yagola, A. (2009). *Mosc. Univ. Phys. Bull.* **65**, 65–67.
- Titarenko, V., Bradley, R., Martin, C., Withers, P. J. & Titarenko, S. (2010b). *Proc. SPIE*, **7804**, 78040Z.
- Twomey, S. (1963). *J. ACM*, **10**, 97–101.
- Vykydal, Z., Jakubek, J. & Pospisil, S. (2006). *Nucl. Instrum. Methods Phys. Res. A*, **563**, 112–115.
- Yagola, A. G., Leonov, A. S. & Titarenko, V. N. (2002). *Inverse Probl. Eng.* **10**, 117–129.