A Backprojection Slice Theorem for Tomographic Reconstruction

Eduardo Miqueles, Nikolay Koshev, and Elias S. Helou

***Abstract*— Fast image reconstruction techniques are becoming important with the increasing number of scientific cases in high resolution micro and nano tomography. The processing of the large scale 3D data demands new mathematical tools for the tomographic reconstruction. Due to the high computational complexity of most current algorithms, big data sizes demands powerful hardware and more sophisticated numerical techniques. Several reconstruction algorithms are dependent on a mathematical tool called backprojection (a transposition process). A conventional implementation of the backprojection operator has cubic computational complexity. In the present manuscript we propose a new fast backprojection operator for the processing of tomographic data, providing a low-cost algorithm for this task. We compare our formula against other fast transposition techniques, using real and simulated large data sets.**

***Index Terms*— Imaging, tomography, reconstruction algorithms.**

I. Introduction

TOMOGRAPHIC imaging is a very powerful instrument of non-destructive research and control of the internal structure of non-opaque objects. An important branch of tomographic techniques is *transmission tomography*, which can be used at nano, micro and macro resolution levels. Physically, tomography is based on registering the energy loss or/and intensity loss of the incoming electromagnetic wave (X-rays for instance) or even the elastic scattering after passing through the object under investigation also referred to here as *sample*. In our case, we consider that X-rays generated from a synchrotron light source hit the object under investigation determining a projection image (also referred as *frame*) at a CCD (charge coupled device) camera. A typical dataset is shown in Figure 1.A, where a high-resolution frame P gathered using the X-rays source is shown, with dimensions 2048×2048. After half-rotation of the sample on the rotation axis (indicated by the angle θ), we obtain a cubic dataset as shown in Figure 1.B. Each slice of this dataset give us an image, which is called *sinogram*, and that will be used as input to an appropriate inversion algorithm in order to reconstruct the slice of the sample. The classical inversion problem described by the aforementioned techniques is modelled by the Radon transform $\mathscr{R}$, which will be defined later, after we pose our problem. A simple tomographic image reconstruction scheme is presented in Figure 2. One of the most celebrated reconstruction algorithms, the *filtered backprojection*, depends explicitly on the action of two operators - say $\mathcal{P}$ and $\mathscr{B}$ - on the measured data g, so that the output image can be stored as $f = \mathscr{B}\{\mathcal{P}[g]\}$ for all possible pixels. Usually, $\mathcal{P}$ is a pre-processing step including removal of ring artifacts, low-pass filtering or even phase-retrieval techniques. The nature of the measured data g - say emission tomography, transmission tomography, missing wedge acquisition, truncated data, attenuated data - usually determines the choice of the best reconstruction algorithm. The majority of these algorithms are

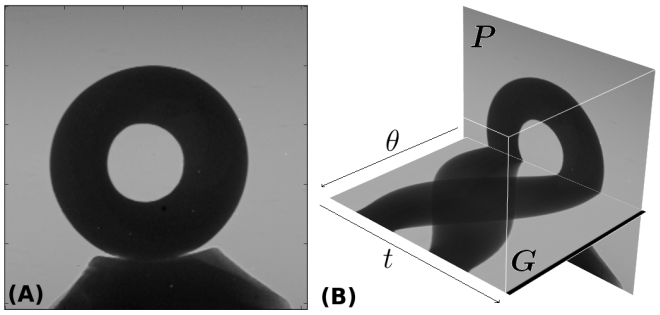

Fig. 1. (A) Projection (or *frame*) for a cylindrical sample obtained with a CCD camera (B) 3D representation of the measured data: P is the measured frame and G is the sinogram image at a given row of the area detector.

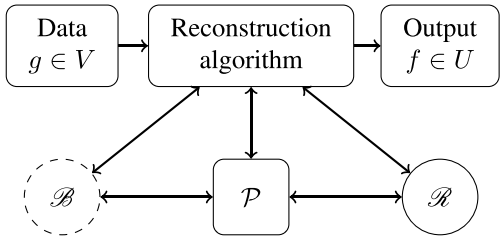

Fig. 2. Fluxogram for a tomographic reconstruction. Most of the existing algorithms depends on the operators $\{\mathscr{B}, \mathscr{R}, \mathcal{P}\}$. A new approach for $\mathscr{B}$ is presented in this manuscript; see the text for further details.

Manuscript received August 2, 2016; revised February 16, 2017, May 25, 2017, August 11, 2017, and October 4, 2017; accepted October 20, 2017. Date of publication October 26, 2017; date of current version November 28, 2017. The work of E. Miqueles was supported by CNPq under Grant 442000/2014-6. The work of N. Koshev was supported by Fapesp under Grant 2014/22040-7. The work of E. S. Helou was supported in part by Fapesp under Grant 2013/07375-0 and Grant 2016/24286-9 and in part by CNPq under Grant 311476/2014-7. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Abd-Krim K. Seghouane. *(Corresponding author: Eduardo Miqueles.)*

E. Miqueles is with the Brazilian Synchrotron Light Laboratory, Brazilian Center for Research in Energy and Materials, São Paulo 13083-970, Brazil (e-mail: eduardo.miqueles@lnls.br).

N. Koshev and E. S. Helou are with the Instituto de Ciências Matemáticas e de Computação, Universty of São Paulo, São Carlos 13566-590, Brazil.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2766785

explicitly dependent on the action of the operators $\mathscr{B}$, $\mathscr{R}$ and eventually $\mathcal{P}$.

The computation of the operator $\mathscr{R}$ can be made very fast using, for example, the Fourier-slice Theorem (FST) [1], which relates the Fourier transform of the sinogram data g with the Fourier transform of the reconstructed image f. The evaluation of $\mathscr{R}f$ is a different algorithmic problem, which will not be covered in this manuscript [2]. There exists, nowadays, a class of sophisticated algorithms specially dedicated to the inversion of g using the Fourier-slice theorem, called gridding methods [3]–[6]. These are fast algorithms mainly due to the use of Fast Fourier transform algorithms in the polar domain.

Even though the main concern of the imaging community is inversion, as suggested by the fluxogram of Figure 2, our goal in this manuscript is not the solution f. Instead, we look for a fast computation of $\mathscr{B}$. Indeed, as pointed by the literature in the last twenty years, most of the reconstruction algorithms, and for many different kind of imaging problems related to the Radon transform, make an intense use of the operator $\mathscr{B}$. We justify the introduction of this new technique in several ways throughout the manuscript. First, numerical experimentation indicates that it may be faster than other available methods. It is also very simple to implement compared to many of the currently existing alternatives. Further, it presents images with less artifacts than some of the other approaches for this problem. Very importantly, it enables a natural and very fast implementation of Tikhonov regularization for the Radon inversion problem, which does not seem to be the case for the other techniques in the literature. Finally, it presents a dual nature with the important Fourier Slice Theorem, a feature that is very satisfying and that could provide future insights in the theory of tomographic image reconstruction.

For concreteness, some reconstruction algorithms that use the backprojection operator are listed below:

- **a)** *Analytical*: The filtered backprojection, or the filter of the backprojection, are well known 2D analytical inversion formulas, given respectively as $f = \mathscr{B}[\mathcal{P}_1 g]$ and $f = \mathcal{P}_2[\mathscr{B}g]$, where $\mathcal{P}_1$ and $\mathcal{P}_2$ are 1D and 2D convolution filters respectively - see [7], [8];
- **b)** *Expectation Maximization*: Based on the maximization of the log-likelihood function, there is a difference in the emission and the transmission case. Both are multiplicative algorithms given as fixed points of the operators $\phi(f) = f\mathscr{B}[g/\mathscr{R}f] \,/\, \mathscr{B}o$ and $\phi(f) = f\mathscr{B}\left[de^{-\mathscr{R}f}\right] \,/\, \mathscr{B}[h]$, respectively for emission and transmission. Here, where $\{h, d\}$ are the photon and blank scan counts respectively; and o is a sinogram of ones;
- **c)** *Generalized Transforms*: The generalized attenuated Radon transform problem $g = \mathscr{R}_\omega f$ can be properly inverted provided that the weight function w obeys a decaying criteria [9]. A particular inversion scheme - using the inverse of the Radon transform - is obtained through the resolvent formalism. The details can be found in [9]–[11] and we use fixed point iterations of the functional by $\phi(f) = f + \mathscr{B}\mathcal{P}[g - \mathscr{R}_\omega f]/a$, with a an appropriate function depending on the weight ω. Several emission modalities take advantage of this formula, like PET, SPECT and XFCT [12]–[14]. In particular, the case $a \equiv 1$ and $w \equiv 1$ provide a simple iterative refinement for the non-attenuated case $g = \mathscr{R}f$;
- **d)** *Regularization approach*: As is easy to note, reconstruction techniques based on the minimization of the functional $\|\mathscr{R}f - g\|_{L_2}^2 + H(f)$ will use, per iteration, the computation of both operators $\{\mathscr{R}, \mathscr{B}\}$. Here, $H(f)$ is an operator imposing some property on the final reconstructed image f. This family of inversion algorithms has gained some attention recently, for example when H is the total variation operator;
- **e)** *Fan/Cone Beam inversion*: A classic result by [15] establishes the fan-beam reconstruction formula as a series, where the backprojection is the zeroth order term. Apart from being an interesting result, the fan-beam inversion depends explicitly on the computation of $\mathscr{B}$, which also plays an important role in the Feldkamp-Davis-Kress [16] inversion algorithm for 3D data;

Describing the above small class of simple reconstruction algorithms from (a) to (e), we note that we do not always need an explicit evaluation of $\mathscr{B}g$ with $g = \mathscr{R}f$. In fact, this is a trivial problem with an easy solution: $\mathscr{B}\mathscr{R}f = f \star \frac{1}{\|x\|}$, with $\star$ standing for a 2D convolution, [8], [17]. Of course, this fact is only true in the ideal case, without noise and with an infinite number of projections. We pose our problem, in the following manner:

Problem: Given any sinogram data $g \in V$, can we find a new and low-complexity algorithm to compute $\mathscr{B}[g] \in U$?

The computation of the backprojection operator bears the major computational cost of $O(N^3)$ for images with N^2 pixels and reconstructed from $O(N)$ angular projections. For a high-resolution tomographic synchrotron experiment, the amount of data at a micro-tomography setup is considerably large for today's computational standards, mainly because of this asymptotic floating point operations (flops) bottleneck. Indeed, at the Brazilian National Synchrotron Light Source (LNLS) one wishes to obtain 2048 reconstructions images with 2048×2048 pixels from datasets having 3200×2048 points, or possibly more. Therefore, implementation of $\mathscr{B}$ represents the main bottleneck of the reconstruction process. If certain useful mathematical properties of $\mathscr{B}$ are exploited, the computational effort can be significantly reduced to $O(N^2 \log N)$ flops [3], [4].

Several techniques were developed aiming at a reduction to $O(N^2 \log N)$ flops for computing $\mathscr{B}$. One approach - due to Andersson - was established in [18], where the computation of $\mathscr{B}g$ is performed after a change from cartesian to log-polar coordinates in the data. This approach leads to a convolution, which is computable through Fast Fourier Transform (FFT) algorithms. Although elegant, the method suffer from the ill-conditioning of the Log-polar transform at the "fovea". Nevertheless, it is possible to translate the fovea to different regions of the cartesian plane, in order to enclose the reconstruction region. This leads to the concept of *partial-backprojection* which can be easily implemented in a parallel form. Other methods for fast computation of $\mathscr{B}$ were presented in [19]–[22], using a divide and conquer strategy based on hierarchical decompositions of the full

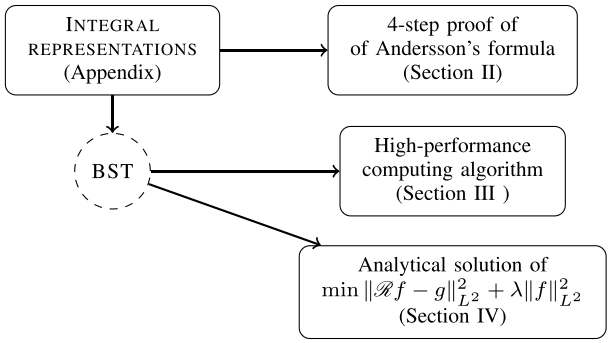

Fig. 3. Fluxogram of the manuscript.

backprojection which are simpler than the full backprojection. Hierarchical decompositions can be created both in the image [21] or in the data [19], [20]. Yet another approach is based on Non-uniform Fast Fourier Transform (NFFT) algorithms (see [21], [23], [24]) and the so-called Fourier Slice Theorem ([3], [5], [8], [25]. Linogram methods [26], [27] are also based on a change of variables producing low cost $O(N^2 \log N)$ methods, but are also focused on reconstruction from projections, and not on the computation of the backprojection.

Our goal in this manuscript is to present a fast method for the computation of the backprojection image $b \in U$ for a given sinogram $g \in V$. We claim [22] that the backprojection of $g \in V$ can be easily done by filtering the lines of the $\tilde{g}$ one by one, where $\tilde{g}$ is the polar representation of g in $S_+ = \mathbb{R} \times [0, 2\pi]$. The methodology developed is called *backprojection-slice theorem* (BST) since it has an obvious relation to the Fourier-slice theorem. During the derivation of the BST formula we have rediscovered a 4-step proof of Andersson's equation which is simpler than the original one presented in his manuscript [18]. Since Andersson's algorithm has the same convolution nature as ours, we present a fair comparison of both techniques throughout the manuscript.

The main contributions of the present paper are presented in the diagram of Figure 3. The integral representations needed to derive our formula are presented in the Appendix. Section II presents a discussion of low-complexity algorithms for the computation of $\mathscr{B}$ and also a 4-step re-derivation of Andersson's formula. Our low-complexity formula is presented in Section III. Further comparison of all algorithms is presented in Section V and a discussion of the results is presented in Section VI.

Mathematical Definitions

We introduce the cartesian coordinate system in the plane of a given slice of the object. Let the function $f(\boldsymbol{x}) \in U$ be the *feature function*, i.e., a function which depends on the internal structure of the object in the plane of the slice and which defines the linear absorption coefficient of the sample. Set U, referred to here as the *feature space*, is a Schwartz space $\mathcal{S}(\mathbb{R}^2)$ (i.e., a space of rapidly decreasing functions on $\mathbb{R}^2$).

A given frame (see Figure 1.A) represents the integral of $f(\boldsymbol{x})$ over straight lines passing through the sample and perpendicular to the detector's plane. One row of each of these

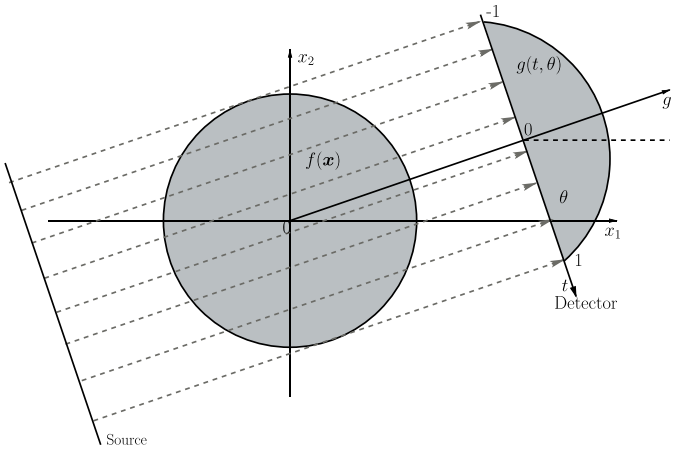

Fig. 4. Geometry of incoming x-rays for parallel tomography.

frame images contains the integrals relevant to a slice of the object, and orthogonal to the rotation axis. Let us introduce an axis t over the detector's row. It is clear that for each angle θ (see Figure 4), such a row is determined by

$$g(\theta, t) = \int\limits_{L(\theta,t)} f(\boldsymbol{x}) \mathrm{d}s = \int\limits_{\mathbb{R}^2} f(\boldsymbol{x}) \delta(\boldsymbol{x} \cdot \boldsymbol{\xi}_\theta - t) \mathrm{d}\boldsymbol{x}, \quad \text{(I.1)}$$

where $L(\theta, t)$ is a straight line defining the x-ray path,

$$L(\theta, t) = \left\{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \cdot \boldsymbol{\xi}_\theta = t\right\}, \quad \boldsymbol{\xi}_\theta = (\cos\theta, \sin\theta)^T. \quad \text{(I.2)}$$

From (I.1) we have a linear operator acting on the feature function f, i.e., $\mathscr{R}\colon f \in U \mapsto g \in V$, which is called the *Radon transform*. Space V is the Schwartz space $\mathcal{S}(\mathbb{R}_+ \times [0, \pi])$. The operator $\mathscr{B}\colon V \to U$ defined as

$$b(\boldsymbol{x}) = \mathscr{B}g(\boldsymbol{x}) = \int_{[0,\pi]} g(\boldsymbol{x} \cdot \boldsymbol{\xi}_\theta, \theta) \mathrm{d}\theta, \quad \text{(I.3)}$$

is defined as the *backprojection* operator, and is the adjoint of $\mathscr{R}$ in the following sense

$$\int\limits_{\mathbb{R}_+ \times [0,\pi]} \mathscr{R}f(t, \theta) g(t, \theta) \mathrm{d}t \mathrm{d}\theta = \int\limits_{\mathbb{R}^2} f(\boldsymbol{x}) \mathscr{B}g(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \quad \text{(I.4)}$$

More about the theory of the integral operators $\{\mathscr{R}, \mathscr{B}\}$ can be found on [17], [25], [28], [29].

At this point, it is convenient to introduce some notations. We first introduce the notations for the representation of feature function $f : \mathbb{R}^2 \to \mathbb{R}$ in different coordinate systems, and their respective jacobians:

a) *Prüfer coordinates (see [30])*: Using

$$\boldsymbol{x} = p(\mu)\boldsymbol{\xi}_\theta, \quad p\colon \mathbb{R}_+ \to \mathbb{R}, \quad \text{(I.5)}$$

we have $\mathrm{d}\boldsymbol{x} = |p'(\mu)p(\mu)| \mathrm{d}\mu \mathrm{d}\theta$. The representation is denoted by $[f]_{\mathsf{Pr}}(\mu, \theta)$. Function p will always be well defined within the context by special notation as follows;

b) *Log-polar coordinates*: particular case of Prüfer coordinates when $p(\mu) = e^\mu$. Here, $\mathrm{d}\boldsymbol{x} = e^{2\mu} \mathrm{d}\mu \mathrm{d}\theta$. The representation is denoted by $[f]_{\mathsf{L}}(\mu, \theta)$;

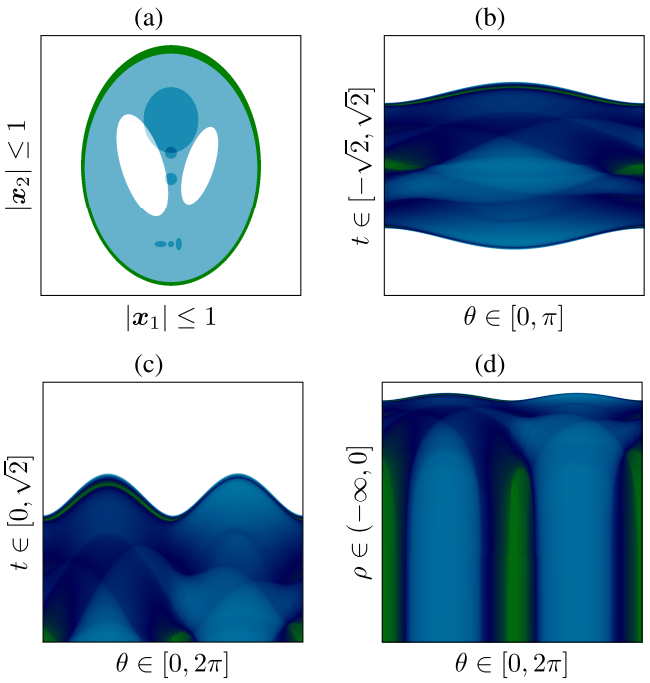

Fig. 5. (a) Shepp-Logan feature function $f(\boldsymbol{x})$ and his associated sinograms in different coordinate systems: (b) Semi-polar coordinates $[g]_{\mathsf{S}}$, (c) Polar coordinates $[g]_{\mathsf{P}}$ and (d) Log-polar coordinates $[g]_{\mathsf{L}}$.

c) *Semi-polar coordinates*: particular case of Prüfer coordinates when $p(\mu) = \mu$. Here, $\mathrm{d}\boldsymbol{x} = \mu \mathrm{d}\mu \mathrm{d}\theta$. The representation is denoted by $[f]_{\mathsf{P}}(\mu, \theta)$.

d) *Sinogram* coordinates are similar to the semi-polar coordinates and, in fact, can be obtained by flipping the angles $\theta \in [\pi, 2\pi)$ to the negative part of the t-axis, so that $t \in [-1, 1]$.

Using the above notation, function g in (I.1) can be written as $[g]_{\mathsf{P}}(t, \theta)$ in order to indicate semi-polar coordinates. An example, using the well-known *Shepp-Logan phantom* [31] is presented in Figure 5. The sinogram of the Shepp-logan feature function f is presented in the sinogram coordinate system mentioned above.

Remark: In this manuscript we use the the integral operator, sometimes with $\mathrm{d}x$ placed before the integrand, as it is more convenient to make explicit the variables being considered. Whenever the integrand is short, we adopt the classic notation $\int f(x)\mathrm{d}x$.

Notation: Some of the mathematical symbols used throughout the manuscript are summarized in Table I.

II. Class of Algorithms for the Backprojection

Let $g \in V$ be a given sinogram. A naïve implementation of the typical backprojection formula (I.4) has to be done using nested loops. Indeed, for each pixel $\boldsymbol{x}$ lying on a predefined meshgrid within the square $\|\boldsymbol{x}\|_\infty \leq 1$, the approximation of $b(\boldsymbol{x}) = \mathscr{B}g(\boldsymbol{x})$ is given by

$$b(\boldsymbol{x}) \approx \Delta\theta \sum_{k=1}^{N_\theta} g(\boldsymbol{x} \cdot \boldsymbol{\xi}_{\theta_k}, \theta_k). \tag{II.1}$$

TABLE I

List of Mathematical Symbols Used in This Manuscript

Symbol	Action
U	Feature space containing functions like $f = f(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^2$
V	Sinogram space containing functions like $g = g(t, \theta)$
$\mathscr{R}$	Radon transform, operating from U to V
$\mathscr{B}$	Backprojection transform, operating from V to U
$\hat{\cdot}$	1D or 2D Fourier transform operator, acting as $\hat{f}$
$\star$	1D or 2D convolution
$\mathcal{I}$	Identity operator
$\|\cdot\|_*$	Norms in $\mathbb{R}^2$, $* \in \{2, \infty\}$
$O(\tau)$	Big-O notation
$[f]_*$	Prüfer, log-polar, polar and cartesian coordinates; $* \in \{\mathsf{Pr}, \mathsf{L}, \mathsf{P}, \mathsf{C}\}$
$\boldsymbol{x}_{\mu,\theta}$	General Prüfer coordinates for point $\boldsymbol{x} \in \mathbb{R}^2$ - see Eq.(I.5)

It is easy to realize that the above approximation has a computational cost of $O(N_\theta)$ for each pixel $\boldsymbol{x}$, where N_θ is the total number of sampled angles. For a high resolution frame (see Figure 1.A), a linear interpolation for $\boldsymbol{x} \cdot \boldsymbol{\xi}_\theta$ on the grid of $-1 \leq t \leq 1$ is usually precise enough. Assuming that b is represented by a square image of order $N \times N$, the total cost for computing the final backprojected image b is $O(N^2 N_\theta)$. In practice, N_θ has almost the same magnitude of N, and thus we can state that the asymptotic cost to obtain b is $O(N^3)$. Such an algorithm is impractical for high-resolution images.

There are at least four other types of backprojection algorithms which can dramatically reduce the computing time of the backprojected image b, for large datasets:

i) A fast slant-stack based approach [32] was proposed by Averbuch *et al.* Although this is an elegant and fast approach, it will not be covered in this manuscript;

ii) *Hierarchical decomposition* [19]–[21]: Two different approaches that apply the divide-and conquer paradigm to the backprojection computation, splitting it recursively into smaller and simpler subproblems;

iii) *NFFT* [33]: The Fourier Slice Theorem sets the Fourier Transform as a bridge between the Radon Transform $\mathcal{R}f(\theta, t)$ and the original image f. However, tomographic data does not provide an evenly distributed sampling of the Fourier space, as required by traditional FFT techniques (see [34]). Use of this Fourier approach was enabled by research on NFFT algorithms (see [21], [23], [24]);

iv) *Andersson's formula* [18]: Such a formula is obtained with an appropriate change of variables on the classical equation of the backprojection formula (I.4). The main idea is to convolve the sinogram in log-polar coordinates with an ideal kernel using FFT algorithms. In this manuscript we refer to LP (log-polar) as the algorithm using Andersson's formula.

In this paper, we focus mainly on the comparison between BST and algorithm (iv).

A. 4-Step Proof of Andersson's Formula

The approach of Andersson is based on a representation of the Backprojection/Radon transform as a convolution, by casting the computation in a log-polar coordinate system. In this section we propose a different proof for his formula. His formula is a consequence of the integral representation of

the backprojection as a summation over circles, as presented in the Appendix A. Using the coordinate system notation of the previous section, where $[\cdot]_{\mathsf{L}}$ denotes the log-polar representation of some function, the main formula of the log-polar backprojection is written as

$$[\mathscr{B}g]_{\mathsf{L}}(\rho,\theta) = [g]_{\mathsf{L}} \star [K]_{\mathsf{L}}\ (\rho,\theta), \tag{II.2}$$

where $\star$ stands for the 2D convolution, and K is the convolution Kernel

$$[K]_{\mathsf{L}}(\rho,\theta) = \delta(1 - e^{\rho}\cos\theta). \tag{II.3}$$

We provide an alternative proof for the equation (II.2), assuming that $g \in \mathcal{S}(\mathbb{R}_+ \times [-\pi,\pi])$.

Proof: We start with the integral representation of the backprojection operator, given in (A.10) (See Appendix A). Now, formula (II.2) is derived as follows:

a) Changing the integral (A.10) from cartesian coordinates $\boldsymbol{y} \in \mathbb{R}^2$ to Prüfer coordinates, i.e., $\boldsymbol{y} \equiv \boldsymbol{y}_{\mu,\theta} = p(\mu)\boldsymbol{\xi}_{\phi}$ we get $\mathrm{d}\boldsymbol{y} = |p'(\mu)p(\mu)|\mathrm{d}\mu\mathrm{d}\theta$ and

$$\mathscr{B}g(\boldsymbol{x}_{\rho,\theta}) = \int_{S_+} g(\boldsymbol{y}_{\mu,\phi})\delta\left(\kappa_{\boldsymbol{x}_{\rho,\theta}}(\boldsymbol{y}_{\mu,\phi})\right) \times |p'(\mu)p(\mu)|\mathrm{d}\mu\mathrm{d}\phi \tag{II.4}$$

b) The argument of the Delta distribution in (II.4) is

$$\begin{aligned}\kappa_{\boldsymbol{x}_{\rho,\theta}}(\boldsymbol{y}_{\mu,\phi}) &= p(\mu)^2\left[1 - \frac{p(\rho)}{p(\mu)}\boldsymbol{\xi}_{\phi}\cdot\boldsymbol{\xi}_{\theta}\right] \\ &= p(\mu)^2\left[1 - \frac{p(\rho)}{p(\mu)}\cos(\phi-\theta)\right]\end{aligned} \tag{II.5}$$

c) Let $[\cdot]_{\mathsf{Pr}}$ be the representation in Prüfer coordinates. From (II.5) and (II.4) we arrive at

$$\begin{aligned}&[\mathscr{B}g]_{\mathsf{Pr}}(\rho,\theta) \\ &= \int_{S_+} [g]_{\mathsf{G}}(\mu,\phi)\delta\left(p(\mu)^2\left[1 - \frac{p(\rho)}{p(\mu)}\cos(\phi-\theta)\right]\right) \\ &\quad\times|p'(\mu)p(\mu)|\mathrm{d}\mu\mathrm{d}\phi \\ &= \int_{S_+} [g]_{\mathsf{G}}(\mu,\phi)\delta\left(1 - \frac{p(\rho)}{p(\mu)}\cos(\phi-\theta)\right) \\ &\quad\times\frac{|p'(\mu)p(\mu)|}{p(\mu)^2}\mathrm{d}\mu\mathrm{d}\phi\end{aligned} \tag{II.6}$$

where $S_+ = \mathbb{R}_+ \times [-\pi,\pi]$

d) A convolution is obtained in (II.6) only if p is such that $p(\rho) = p(\mu)p(\rho-\mu)$, which in turn implies that p is an exponential function. Hence, Prüfer coordinates reduce to log-polar coordinates, which we denote by $[\cdot]_{\mathsf{L}}$. Finally, we obtain

$$[\mathscr{B}g]_{\mathsf{L}}(\rho,\theta) = \int_{S_+} [g]_{\mathsf{L}}(\mu,\phi) \times\delta\left(1 - e^{\rho-\mu}\cos(\phi-\theta)\right)\mathrm{d}\mu\mathrm{d}\phi \tag{II.7}$$

which is the final convolution formula. □

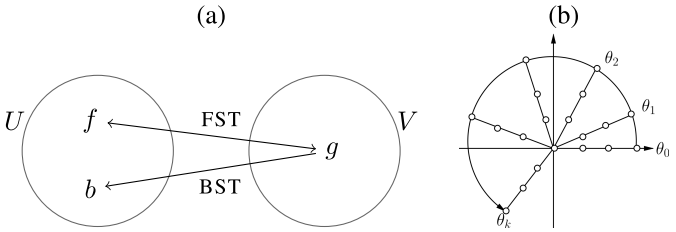

Fig. 6. (a) Action of BST versus FST (Fourier-slice theorem) (b) Action of BST on a polar mesh.

III. Back-Projection Slice Theorem

Although Andersson's approach is asymptotically fast, it has a few drawbacks. Firstly, the gain of speed using Fourier transforms to compute the convolution is reduced with forward/backward log-polar transformations. Also, these interpolations can produce numerical errors, especially near the origin, due to a strong non-uniformity of the log-polar mesh in that region. To avoid these factors, another approach for the calculation of the backprojection operator can be used. This approach is based on the following theorem:

Theorem (Backprojection Slice Theorem (BST)): Let $g = g(t,\theta) \in V$ be a given image in the sinogram space and $\hat{\cdot}$ denote the Fourier transform operation. It follows that the backprojection $\mathscr{B}$ satisfies

$$\widehat{\mathscr{B}g}(\sigma\boldsymbol{\xi}_{\theta}) = \frac{\hat{g}_{\theta}(\sigma)}{\sigma} \tag{III.1}$$

with $\sigma > 0 \in \mathbb{R}$ and $\theta \in [0, 2\pi]$.

Before we proceed with a formal proof, we remark that, the backprojection-slice theorem (BST) is the dual of the Fourier-slice theorem (FST), in the sense that $g = \mathscr{R}f$ and $b = \mathscr{B}g$ are related in the reciprocal space within a polar grid. As a well known fact, the composition $\mathscr{B}\mathscr{R}$ is a convolution operator with point spread function given by $1/\|\boldsymbol{x}\|_2$, [8], [17], [28]. If $g = \mathscr{R}f$ the computation of $b = \mathscr{R}g$ follows easily with the aforementioned 2D convolution. Also, FST provide a direct link between $\{f, g\}$ and $\{b, g\}$. If $g \neq \mathscr{R}f$ the link between $\{f, g\}$ is no longer valid through FST but we can still provide a powerful link between $\{b, g\}$. This is exactly the goal of the backprojection slice theorem. We remark that, physically, the backprojection corresponds to an average 'smear' of all projections passing through a single pixel $\boldsymbol{x}$. Since an average is described as a convolution, it is natural to regard formula (III.1) as an average in the frequency domain. In fact, a convolution with a singular kernel like $1/\sigma$ provides the usual *blurring*, typical from the backprojection operator. The explicit average of all rays $\boldsymbol{x}\cdot\boldsymbol{\xi}_{\theta} = t$ passing through a pixel $\boldsymbol{x}$ is clear in the definition of $\mathscr{B}$, see (I.3). Finally, as shown in diagram (b), it is possible to estimate $b = \mathscr{B}g$ at each radial line $\{(\sigma,\theta_k)\colon \sigma > 0, \theta_k \in [0, 2\pi]\}$, over the polar domain, as presented in Fig. 6.

The following proof provides the calculation of $\mathscr{B}g$ assuming that g is arbitrary in the sinogram space V.

Proof: Using the sifting property of the δ-distribution, the backprojection (I.3) can be presented in the following form

$$\begin{aligned}\mathscr{B}g(\boldsymbol{x}) &= \int_0^{\pi} g(\boldsymbol{x}\cdot\boldsymbol{\xi}_{\theta},\theta)\mathrm{d}\theta \\ &= \int_0^{\pi}\int_{\mathbb{R}} g(t,\theta)\delta(t - \boldsymbol{x}\cdot\boldsymbol{\xi}_{\theta})\mathrm{d}t\mathrm{d}\theta\end{aligned} \tag{III.2}$$

Considering $\widehat{\mathscr{B}g}$ the 2D Fourier transform of $\mathscr{B}g$, we use representation (A.10) (see Appendix) to obtain,

$$\begin{aligned}\widehat{\mathscr{B}g}(\boldsymbol{\omega}) &= \int_{\mathbb{R}^2} \mathscr{B}g(\boldsymbol{x}) e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}} \mathrm{d}\boldsymbol{x} \\ &= \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{x} \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{y}\ [g]_{\mathsf{C}}(\boldsymbol{y}) \delta(\boldsymbol{y}\cdot(\boldsymbol{y}-\boldsymbol{x}))\, e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}} \\ &= \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{y}\ [g]_{\mathsf{C}}(\boldsymbol{y}) \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{x}\ \delta(\boldsymbol{y}\cdot(\boldsymbol{y}-\boldsymbol{x}))\, e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}} \\ &\equiv \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{y}\ [g]_{\mathsf{C}}(\boldsymbol{y}) \mathcal{T}(\boldsymbol{y},\boldsymbol{\omega})\end{aligned}$$

where $\boldsymbol{y}, \boldsymbol{\omega} \in \mathbb{R}^2$ and

$$\mathcal{T}(\boldsymbol{y},\boldsymbol{\omega}) = \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{x}\ \delta\left(h_{\boldsymbol{y}}(\boldsymbol{x})\right) e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}}, \tag{III.3}$$

with $h_{\boldsymbol{y}}$ defined as

$$h_{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{y}\cdot(\boldsymbol{y}-\boldsymbol{x}) \tag{III.4}$$

Since the distribution (III.3) is supported in the set $h_{\boldsymbol{y}}^{-1}(0) = \{\boldsymbol{x} \in \mathbb{R}^2 \colon h_{\boldsymbol{y}}(\boldsymbol{x}) = 0\}$, it follows from (A.1) (See Appendix A) and $\nabla h_{\boldsymbol{y}} = -\boldsymbol{y}$ that

$$\begin{aligned}\mathcal{T}(\boldsymbol{y},\boldsymbol{\omega}) &= \frac{1}{\|\boldsymbol{y}\|_2} \int_{h_{\boldsymbol{y}}^{-1}(0)} e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}} \mathrm{d}s(\boldsymbol{x}) \\ &= \int_{h_{\boldsymbol{y}}^{-1}(0)} e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}(q)} \mathrm{d}q\end{aligned}$$

The set $h_{\boldsymbol{y}}^{-1}(0)$ determines a straight line passing through $\boldsymbol{y}$ and with normal vector $\boldsymbol{y}$. Thus, $h_{\boldsymbol{y}}^{-1}(0) = \boldsymbol{y} + \mathrm{span}\{S\boldsymbol{y}\}$, being $S\boldsymbol{y} \perp \boldsymbol{y}$ and S a $\frac{\pi}{2}$-rotation matrix. Therefore, $\boldsymbol{x}(q) \in h_{\boldsymbol{x}}^{-1}(0)$ is on the form $\boldsymbol{x}(q) = \boldsymbol{y} + qS\boldsymbol{y}$ and the integral in (III.5) can be written as:

$$\mathcal{T}(\boldsymbol{y},\boldsymbol{\omega}) = \int_{\mathbb{R}} e^{-i\boldsymbol{\omega}\cdot[\boldsymbol{y}+qS\boldsymbol{y}]} \mathrm{d}q \tag{III.5}$$

$$= e^{-i\boldsymbol{\omega}\cdot\boldsymbol{y}} \int_{\mathbb{R}} e^{-iq\boldsymbol{\omega}\cdot(S\boldsymbol{y})} \mathrm{d}q \tag{III.6}$$

$$= e^{-i\boldsymbol{\omega}\cdot\boldsymbol{y}} \delta(\boldsymbol{\omega}\cdot S\boldsymbol{y}) \tag{III.7}$$

Hence, the Fourier transform of $\mathscr{B}g$ becomes

$$\widehat{\mathscr{B}g}(\boldsymbol{\omega}) = \int_{\mathbb{R}^2} [g]_{\mathsf{C}}(\boldsymbol{y}) \delta(\boldsymbol{\omega}\cdot S\boldsymbol{y})\, e^{-i\boldsymbol{\omega}\cdot\boldsymbol{y}}\ \mathrm{d}\boldsymbol{y} \tag{III.8}$$

For $\boldsymbol{\omega}$ fixed, $\{\boldsymbol{y} \in \mathbb{R}^2 \colon \boldsymbol{\omega}\cdot(S\boldsymbol{y}) = 0\} = \mathrm{span}\{\boldsymbol{\omega}\}$, with S a $\frac{\pi}{2}$-rotation matrix. Indeed, since $S\boldsymbol{y} \perp \boldsymbol{w}$ and $S\boldsymbol{y} \perp \boldsymbol{y}$, it follows $\boldsymbol{\omega} \parallel \boldsymbol{y}$. Once again, using the representation (A.1) (see Appendix A) for (III.8) we arrive at

$$\widehat{\mathscr{B}g}(\boldsymbol{\omega}) = \int_{\mathbb{R}} \frac{[g]_{\mathsf{C}}(q\boldsymbol{\omega})}{\|S\boldsymbol{\omega}\|_2} e^{-i\boldsymbol{\omega}\cdot(q\boldsymbol{\omega})}\ \mathrm{d}s(\boldsymbol{\omega}) \tag{III.9}$$

Since $\|S\boldsymbol{\omega}\|_2 = \|\boldsymbol{\omega}\|_2$ and $\mathrm{d}s(\boldsymbol{\omega}) = \|\boldsymbol{\omega}\|_2 dq$, we finally obtain

$$\widehat{\mathscr{B}g}(\boldsymbol{\omega}) = \int_{\mathbb{R}} [g]_{\mathsf{C}}(q\boldsymbol{\omega}) e^{-iq\|\boldsymbol{\omega}\|_2^2}\ \mathrm{d}q \tag{III.10}$$

From the above equation, the backprojection is a polar convolution. Indeed, switching the frequency domain to polar coordinates, i.e., $\boldsymbol{\omega} = \sigma\boldsymbol{\xi}_\theta$ (with $\sigma \in \mathbb{R}_+$ and $\theta \in [0, 2\pi]$) we get

$$\widehat{\mathscr{B}g}(\sigma\boldsymbol{\xi}_\theta) = \int_{\mathbb{R}} [g]_{\mathsf{C}}(q\sigma\boldsymbol{\xi}_\theta) e^{-iq\|\sigma\boldsymbol{\xi}_\theta\|_2^2} \mathrm{d}q \tag{III.11}$$

$$= \int_{\mathbb{R}} \frac{[g]_{\mathsf{C}}(u\boldsymbol{\xi}_\theta)}{\sigma} e^{-iu\sigma}\, \mathrm{d}u. \tag{III.12}$$

Observe that $[g]_{\mathsf{C}}(u\boldsymbol{\xi}_\theta) = g(u,\theta)$ is the input sinogram $g(u,\theta)$. From (III.10) and (III.12), using polar coordinates

$$[\widehat{\mathscr{B}g}]_{\mathsf{P}}(\sigma,\theta) = \widehat{\mathscr{B}g}(\sigma\boldsymbol{\xi}_\theta) = \frac{1}{\sigma}\int_{\mathbb{R}} g(u,\theta) e^{-iu\sigma}\, \mathrm{d}u \tag{III.13}$$

Identity (III.13) is our backprojection-slice Theorem (III.1) for computing the operator $\mathscr{B}$. ■

Indeed, at each radial line θ in the frequency domain, the 2D Fourier transform of $\mathscr{B}$ equals the 1D radial Fourier transform of the projection $g(t,\theta)$ multiplied by the kernel $1/\sigma$ for $\sigma > 0$.

Remark 1: The mathematical proof outlined above provides a *direct* formula for the computation of a backprojected image, i.e., given a sinogram g, the explicit steps to compute the backprojection in the frequency polar coordinates results in formula (III.1). In practice, there are several iterative methods that depends explicitly on the computation of the backprojection of *any* sinogram. In the other hand, analytical formulas usually handle with the backprojection of a filtered sinogram, from where standard formulas like the *filtered backprojection* or the *filter of the backprojection* are established. To validate our backprojection result we remark the following items:

(i) It is a well known fact [17], [25], [28] that, for a given feature function $f \in U$, the following property holds

$$\mathscr{B}\mathscr{R}f(\boldsymbol{x}) = (f \star h)(\boldsymbol{x}), \quad h(\boldsymbol{x}) = \frac{1}{\|\boldsymbol{x}\|_2} \tag{III.14}$$

which, in the frequency domain, is written as (cartesian and polar representation, respectively)

$$\widehat{\mathscr{B}\mathscr{R}f}(\boldsymbol{w}) = \hat{f}(\boldsymbol{w})\frac{1}{\|\boldsymbol{w}\|_2} \ \Leftrightarrow\ \widehat{\mathscr{B}\mathscr{R}f}(\sigma\boldsymbol{\xi}_\theta) = \hat{f}(\sigma\boldsymbol{\xi}_\theta)\frac{1}{\sigma} \tag{III.15}$$

due to the fact that $\mathcal{F}\colon \frac{1}{\|\boldsymbol{x}\|_2} \mapsto \frac{1}{\|\boldsymbol{w}\|_2}$. Now, replacing the backprojection slice theorem (III.1) into (III.15), we obtain

$$\widehat{\mathscr{R}f}(\sigma\boldsymbol{\xi}_\theta)\frac{1}{\sigma} = \hat{f}(\sigma\boldsymbol{\xi}_\theta)\frac{1}{\sigma} \ \Rightarrow\ \widehat{\mathscr{R}f}(\sigma\boldsymbol{\xi}_\theta) = \hat{f}(\sigma\boldsymbol{\xi}_\theta) \tag{III.16}$$

which is the celebrated Fourier Slice-Theorem [29].

(ii) From the classical inversion of the Radon transform, i.e., the filtered-backprojection algorithm, it is true that

$$\mathscr{B}Fg(\boldsymbol{x}) = f(\boldsymbol{x}), \quad g = \mathscr{R}f \tag{III.17}$$

where F is a low-pass filtering operator, that is $\widehat{Fg}(\nu,\theta) = \hat{g}(\nu,\theta)|\nu|$, for $\nu \in \mathbb{R}$. In the polar frequency domain, (III.17) reads $\widehat{\mathscr{B}Fg}(\sigma\boldsymbol{\xi}_\theta) = \hat{f}(\sigma\boldsymbol{\xi}_\theta)$. From the backprojection slice Theorem (III.1), such equation becomes

$$\frac{1}{\sigma}\widehat{Fg}(\sigma\boldsymbol{\xi}_\theta) = \hat{f}(\sigma\boldsymbol{\xi}_\theta) \ \Rightarrow\ \frac{1}{\sigma}\hat{g}(\sigma\boldsymbol{\xi}_\theta)\sigma = \hat{f}(\sigma\boldsymbol{\xi}_\theta), \quad \sigma \in \mathbb{R}_+ \tag{III.18}$$

Once again, the above equation yields the Fourier Slice-Theorem.

Remark 2: The DC-component of the Backprojection of some function g lying in the sinogram space is defined by

$$\widehat{\mathscr{B}g}(\mathbf{0}) = \int_{\mathbb{R}^2} \mathscr{B}g(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{III.19}$$

$$= \int_{\mathbb{R}^2} \mathrm{d}\boldsymbol{x} \int_0^{\pi} \mathrm{d}\theta \; g(\boldsymbol{x}\cdot\boldsymbol{\xi}_\theta, \theta) \tag{III.20}$$

$$= \int_0^{\pi} \mathrm{d}\theta \int_{\mathbb{R}} \mathrm{d}t \int_{\mathbb{R}} \mathrm{d}s \; g(t,\theta) \equiv M \tag{III.21}$$

where we have used $\mathrm{d}\boldsymbol{x} = \mathrm{d}t\mathrm{d}s$ to make explicit the change of variables from $\boldsymbol{x}$ to (t, s), being s the variable along the direction $\boldsymbol{\xi}_\theta^{\perp}$. The DC of an arbitrary $g \in V$ provide[1] $M = \infty$. In this sense, $\widehat{\mathscr{B}g}$ behaves like a tempered distribution since $\mathscr{B}g$ lies in a Schwartz space, where the Fourier transform is an automorphism. Also, it is easy to note that

$$\frac{\hat{g}(\sigma,\theta)}{\sigma} = i\hat{h}(\sigma,\theta), \quad h(t) = \int_{\infty}^{t} g(t,\theta)\mathrm{d}t \tag{III.22}$$

i.e., h is a primitive of g. Hence, using (III.22) as $\sigma \to 0$, the limit of the ratio $\hat{g}(\sigma,\theta)/\sigma$ diverges in $\sigma = 0$. Still, BST formula can be easily applied for some $g \in V$ with a nonzero DC-component. In fact, setting $p(t,\theta) = g(t,\theta) - \hat{g}(0,\theta)$, it is true that $\hat{p}(0,\theta) = 0$ and the backprojection of g follows with $\mathscr{B}g(\boldsymbol{x}) = \mathscr{B}p(\boldsymbol{x}) + \hat{g}(0,\theta)$.

IV. Regularized FBP: an Application of BST

The BST formula (III.1) can be used to obtain an analytical solution of the standard Tikhonov regularization problem in the feature space $(U, \|\cdot\|_{L^2})$

$$\underset{f\in U}{\text{minimize}} \; \|\mathscr{R}f - g\|_{L_2}^2 + \lambda\|f\|_{L_2}^2 \tag{IV.1}$$

In fact, the Euler-Lagrange equations provide the optimality condition for the above optimization problem, i.e., f minimizes (IV.1) if and only if [35]

$$(\mathscr{R}^*\mathscr{R} + \lambda\mathcal{I})f(\boldsymbol{x}) = \mathscr{R}^*g(\boldsymbol{x}) \tag{IV.2}$$

with $\mathscr{R}^*$ standing for the adjoint operator of the Radon transform and $\mathcal{I}$ the identity operator in U. In fact, (IV.2) are the so-called normal equations in the Hilbert spaces U and V. Since $\mathscr{R}^* = \mathscr{B}$ in the usual inner-product for L_2, the above equation becomes

$$(\mathscr{B}\mathscr{R} + \lambda\mathcal{I})f(\boldsymbol{x}) = \mathscr{B}g(\boldsymbol{x}) \tag{IV.3}$$

Applying the Fourier transformation on (IV.3) and using property (III.14), we obtain the following standard result

$$\hat{f}(\boldsymbol{\omega})\frac{1}{\|\boldsymbol{\omega}\|_2} + \lambda\hat{f}(\boldsymbol{\omega}) = \widehat{\mathscr{B}g}(\boldsymbol{\omega})$$

$$\iff \hat{f}(\boldsymbol{\omega})\left(\frac{1+\lambda\|\boldsymbol{\omega}\|_2}{\|\boldsymbol{\omega}\|_2}\right) = \widehat{\mathscr{B}g}(\boldsymbol{\omega}) \tag{IV.4}$$

[1]Even if g is the sinogram of a compactly supported function feature function on the unit disk $\|x\|_2 \le 1$, we have $\hat{g}(0,\theta) =$ constant, although with $M = \infty$.

From (IV.4) is easy to obtain f as a convolution of $\mathscr{B}g$ with an specific 2D filter. If $\lambda = 0$ the analytical formula obtained is exactly the 'rho-filter layergram' proposed in [7] consisting in a post-processing of the backprojection (also mentioned earlier in this manuscript as *filter of the backprojection*).

The novelty here is that, if we change (IV.4) to polar coordinates, we can immediately apply the BST formula (III.1). Indeed, since $\boldsymbol{\omega} = \sigma\boldsymbol{\xi}_\theta$, the pointwise product becomes

$$\hat{f}(\sigma\boldsymbol{\xi}_\theta)\left(\frac{1+\lambda\sigma}{\sigma}\right) = \widehat{\mathscr{B}g}(\sigma\boldsymbol{\xi}_\theta) \tag{IV.5}$$

which is essentially the same as

$$\hat{f}(\sigma\boldsymbol{\xi}_\theta) = \left(\frac{1}{1+\lambda\sigma}\right)\hat{g}(\sigma,\theta) \tag{IV.6}$$

The above equation is a regularized version of the Fourier-Slice-Theorem and can be used to obtain f explicitly through any gridding strategy [3], [5]. In fact, after the filter of g with $1/(1+\lambda\sigma)$ we can use the Direct Fourier inversion method (i.e., a gridding method) described in [4] and [36] to obtain f.

Applying (IV.6) in the change of variables of the Fourier representation of $f(\boldsymbol{x})$ we finally obtain a new representation for the reconstructed image f,

$$f(\boldsymbol{x}) = \int_{\mathbb{R}^2} \hat{f}(\boldsymbol{\omega})e^{i\boldsymbol{\omega}\cdot\boldsymbol{x}}\mathrm{d}\boldsymbol{\omega}$$

$$= \int_{\mathbb{R}} \mathrm{d}\sigma \int_0^{\pi} \mathrm{d}\theta f(\sigma\boldsymbol{\xi}_\theta)|\sigma|e^{i\sigma\boldsymbol{x}\cdot\boldsymbol{\xi}_\theta}$$

$$= \int_{\mathbb{R}} \mathrm{d}\sigma \int_0^{\pi} \mathrm{d}\theta \left(\frac{1}{1+\lambda|\sigma|}\right)\hat{g}(\sigma,\theta)|\sigma|e^{i\sigma\boldsymbol{x}\cdot\boldsymbol{\xi}_\theta} \tag{IV.7}$$

Equation (IV.7) provides exactly the same reconstruction pattern as a typical filtered backprojection reconstruction algorithm, but with a different filter. In fact, we can generalize our regularized strategy in the following representation

$$f_\lambda(\boldsymbol{x}) = \mathscr{B}F_\lambda g(\boldsymbol{x}) \tag{IV.8}$$

Now, $\{f_\lambda\}$ is a family of solutions of the optimization problem (IV.1), depending on the regularization parameter λ. The filter function F_λ, in the frequency domain reads

$$\widehat{F_\lambda}(\sigma) = \frac{|\sigma|}{1+\lambda|\sigma|} \tag{IV.9}$$

Our regularized solution (IV.8) depends explicitly on the computation of the Backprojection operator $\mathscr{B}$, and either the BST, Andersson's formula, Slant-Stack or other approach can be used to obtain the final inversion f.

V. Numerical Results

All the algorithms were implemented using the fast Fourier framework FFTW3 [37]. We validate our approach using four datasets: one real sinogram and three simulated. The experimental sinograms (a slice from a wood-fiber) was obtained at the imaging beamline of the Brazilian Synchrotron light source and are high-resolution images with 2048×1000 (rays $\times$ angles). Therefore, the feature images (either backprojected or filtered-backprojected) were restored with 2048×2048 pixels in order to test the efficiency of the algorithms. The simulated

data are: i) the classical shepp-logan phantom depicted in Figure 5, ii) the circular function and iii) the following linear combination given respectively by

$$f(\boldsymbol{x}) = \begin{cases} 1, & \|\boldsymbol{x}\|_2 \leq \dfrac{1}{2} \\ 0, & \text{otherwise} \end{cases} \qquad f(\boldsymbol{x}) = \sum_{j=1}^{1000} \delta(\boldsymbol{x} - \boldsymbol{a}_j) \tag{V.1}$$

where $\{\boldsymbol{a}_j\}$ are points randomly spanned over the domain $[-0.3, 0.3] \times [-0.3, 0.3]$, We remember that the analytical backprojection is easily obtained from the convolution of the original image f with the point spread function $1/\|\boldsymbol{x}\|_2$.

The BST strategy will be compared against Andersson's algorithm (LP standing for log-polar) since they have a similar convolution nature. Image comparisons between BST, using Non-equispaced FFT's and the ray-based Bresenham's algorithm (BRES) [38] are also presented in order to illustrate the quality of the final backprojected image.

In this section we use the following symbols to indicate mesh properties:

- $\{N_s, N_\theta, \Delta s\}$: number of points in the polar domain $s \geq 0$, number of points in the angular domain $\theta \in [0, 2\pi]$, and the associate stepsize in the s-axis;
- $\{N, \Delta_{\boldsymbol{x}}\}$: number of points in the square domain $[-1, 1]^2$ and the associate stepsize (equal in both directions);
- N_ρ: number of points in the log-polar domain $\rho \in (\rho_0, 0]$. Here, we use ρ_0 as a large negative number in order to approximate $s = 0$ in the polar domain.

A. Time Speed: BST × LP

The straight usage of the Fourier transform for the implementation of (III.1) - produces rather big artifacts near the origin (boundary effect), caused by the fact that the values on sinogram on the line $s = 0$ are not equal to 0. This problem can be solved with usage of short-time Fourier transform near the origin, with window w. In this work we use as the Kaiser-Bessel function [39], defined on the polar mesh through:

$$w(s) = \left| I_0\left(s\sqrt{1 - \left(\frac{2i - N_s + 1}{N_s - 1}\right)^2}\right)\right| / |I_0(s)|, \tag{V.2}$$

where $I_0(\cdot)$ is a modified Bessel function of the zeroth order.

The BST strategy applied over a sinogram image g is obtained after 7 processing stages, each one executed within an elapsed time τ_k. The final backprojected image b is obtained as a composition of algorithms $\{P_k\}$, i.e.,

$$b = P_6 P_5 P_4 P_3 P_2 P_1 P_0(g) \tag{V.3}$$

Each processing step P_k was implemented in a parallel form with m threads. Step P_0 indicates an interpolation to polar coordinates with the multiplication by the Kaiser-Bessel window function (V.2). This is an easy process, computed with complexity $O(N_\theta N_s)$. Step P_1 is the zero padding of the polar sinogram - equivalent to an oversampling in the frequency domain - with the same complexity of P_0. Even though P_0 and P_1 can be merged into one single step, they were considered disjoint operations in our customized implementation. Step P_2 is the convolution of the polar sinogram with kernel $1/\sigma$. This part was divided in m parallel FFT's, each computed at an individual thread with complexity $O(N_s \log N_s)$. Step P_3 is an interpolation from polar to cartesian coordinates in the frequency domain. The bigger the zero padding at step P_1, the better this part will behave, preventing aliasing artifacts. Step P_4 is a 2D inverse Fourier transform of the data from step P_3. This is an operation with low computational complexity and obtained with order $O(N_s^2 \log N_s)$. Step P_5 is the FFT shifting from previous step and P_6 the final (and optional) 2D interpolation of the resulting image to the correct feature domain $[-1, 1] \times [-1, 1]$.

We have performed a total of 10 executions, for a fixed number of threads m, yielding a sequence of 10 elapsed times $\{\tau_k^{(1)}, \tau_k^{(2)}, \ldots, \tau_k^{(10)}\}$ with summation $\{\tau^{(1)}, \ldots, \tau^{(10)}\}$ respectively, corresponding to the final execution time of a backprojection with BST. The average times (in seconds) for each step P_k are presented in Table II. The standard deviation of the execution is indicated in the table with the symbol `std`. As expected, most of the computational complexity comes from step P_2, the 1D convolution with kernel $1/\sigma$.

The execution time using the reconstruction software Tomopy [40] was also included in Table II as a reference. Using the same number of cores, BST is able to backproject a sinogram image of 1024 rays and 2048 angles in 0.565 seconds, while the reference software builds a reconstruction with gridding strategies in 0.762 seconds. For a sinogram

TABLE II

EXECUTION TIME FOR BST (IN SECONDS) USING A INTEL XEON 3.4 GHz AND DIFFERENT NUMBER m OF THREADS. THE SIZE OF BACKPROJECTED IMAGES IS EQUAL TO THE NUMBER OF RAYS. TOMOPY PARALLELIZATION IS ON THE SLICE LEVEL, USING A SINGLE THREAD FOR ONLY ONE SLICE [40]; WHICH IS A DIFFERENT APPROACH ADOPTED IN THE IMPLEMENTATION OF BST FOR THIS MANUSCRIPT

Geometry: 1024 rays × 2048 angles							*Tomopy*: 0.762 sec	
m	τ	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
1	0.754	0.027	0.020	0.551	0.106	0.029	0.012	0.009
std	0.011	0.006	0.002	0.005	0.004	0.002	0.000	0.001
2	0.385	0.014	0.011	0.280	0.053	0.015	0.007	0.005
std	0.010	0.002	0.001	0.009	0.002	0.000	0.000	0.000
4	0.244	0.009	0.006	0.180	0.033	0.010	0.003	0.002
std	0.013	0.002	0.000	0.012	0.001	0.001	0.000	0.000
8	0.182	0.008	0.004	0.125	0.020	0.022	0.002	0.001
std	0.005	0.002	0.000	0.003	0.000	0.001	0.000	0.000
16	0.134	0.007	0.004	0.097	0.012	0.010	0.002	0.002
std	0.005	0.002	0.000	0.004	0.000	0.001	0.000	0.000
32	0.131	0.006	0.003	0.088	0.010	0.018	0.003	0.003
std	0.002	0.002	0.000	0.001	0.000	0.001	0.000	0.000
Geometry: 2048 rays × 4096 angles							*Tomopy*: 2.761 sec	
m	τ	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
1	2.653	0.311	0.287	1.147	0.504	0.188	0.151	0.065
std	0.080	0.006	0.002	0.039	0.053	0.021	0.026	0.007
2	1.515	0.172	0.172	0.662	0.277	0.092	0.106	0.034
std	0.095	0.025	0.030	0.081	0.018	0.008	0.009	0.001
4	1.010	0.116	0.109	0.469	0.178	0.058	0.061	0.019
std	0.050	0.013	0.015	0.046	0.009	0.002	0.002	0.002
8	0.845	0.100	0.101	0.396	0.107	0.071	0.057	0.013
std	0.023	0.007	0.010	0.024	0.002	0.002	0.004	0.001
16	0.737	0.096	0.095	0.337	0.082	0.060	0.056	0.011
std	0.012	0.006	0.007	0.008	0.002	0.002	0.002	0.001
32	0.683	0.094	0.088	0.323	0.068	0.044	0.052	0.012
std	0.008	0.001	0.001	0.005	0.006	0.002	0.001	0.000

with dimensions (2048, 4096) BST is also comparable with Tomopy, processing the final image in 0.18 seconds against 2.761 seconds of the reference software. We have not included time for other reconstruction softwares because BST relies on the same gridding strategy of Tomopy's algorithm. We remark that 2.761 seconds is the time for processing only one slice using the gridding algorithm from Tomopy; to process a block of $S > 1$ sinograms, the average time obtained is 0.17 seconds per slice, which is close to the one obtained using BST. It is also important to note that parallelization of Tomopy is on the slice level, using a single thread to process a single slice. This is a different approach from the one adopted in the manuscript, where we parallelize the process of a single slice. A comparison of BST and Tomopy, for a block of sinograms is beyond the scope of this work, as we are mainly interested in the process of one slice through the new backprojection formulation.

The complexity of reconstruction, based on BST formula depends on the size of the final image, which we are using as the number of given rays. Much of computational complexity falls on the 1D Fourier transforms, whose size is equal to the number of rays in the sinogram, and to final 2D Fourier transform in cartesian coordinates. Hence, the complexity is of order $O\left(N_\theta N_s \log_2(N_s) + 2N^2 \log_2 N\right) + \Omega_{\mathsf{P}}$ where Ω_{P} denotes the total amount of complexity for polar interpolations (polar to cartesian, usually $\Omega_{\mathsf{P}} = O(N_s^2)$) and N_s is the number of rays in the polar representation.

B. Image Comparisons

According to the log-polar representation, $s \to 0$ in the polar domain causes $\rho \to -\infty$ in the log-polar domain. Of course, in real calculations it is not possible to get a proper interpolation to this grid. This practical problem can be solved in two ways. The first approach is clearly mathematical, and was proposed by Andersson [18], called *partial backprojection method*. This method is based on moving the origin outside the region of interest, which allows us to make a clear interpolation of all points of the sinogram with non-zero values. The second approach is to select a proper ρ_0, which adapts properly to the resulting cartesian grid. For clear interpolation we have to use ρ_0 as a very large negative number, from which we start the approximation to the log-polar mesh. But, in fact, this number is connected to the mesh which is chosen for cartesian representation of the result. Assume the cartesian mesh has the direction step $\Delta_{\boldsymbol{x}}$. In this case, to avoid the loss of information near the origin, we have to set $\rho_0 < \ln \Delta_{\boldsymbol{x}}$. The performance of Log-polar backprojection depends on the desired resolution of the resulting image. The oversampling of log-polar mesh grows up fast as the number of pixel increases in the cartesian grid, as it shown below in this section. To obtain a good reconstruction, it is easy to obtain the number of points N_ρ in the axial log-polar domain using $N_\rho \approx \ln(1/N)/\ln(1-\Delta s)$). An example of the log-polar backprojection using an adptative ρ_0 selection is presented in Figure 7, for the classical Shepp-logan function. Images (c) and (d) make a local comparison of BST and LP near the origin, where we see the effect of the Kaiser-Bessel function.

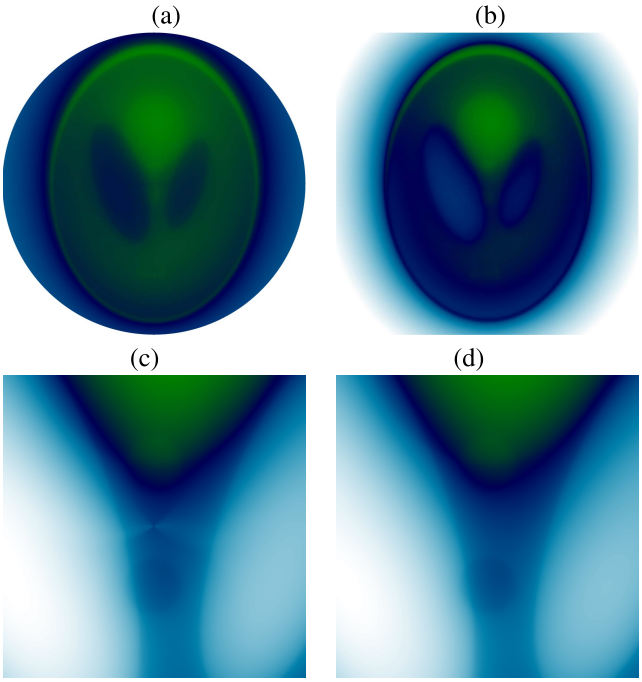

Fig. 7. Backprojection applied to the Shepp-Logan function and their comparison near the origin. (a) LP. (b) BST. (c) LP near $\boldsymbol{x} = \mathbf{0}$. (d) BST near $\boldsymbol{x} = \mathbf{0}$.

TABLE III

EXECUTION TIME OF ANDERSSON'S ALGORITHM (IN SECONDS) USING A INTEL XEON 3.4GHz AND DIFFERENT NUMBER m OF THREADS. THE SIZE OF BACKPROJECTED IMAGES IS EQUAL TO THE NUMBER OF RAYS. TOMOPY PARALLELIZATION IS ON THE SLICE LEVEL, USING A SINGLE THREAD FOR ONLY ONE SLICE [40]; WHICH IS A DIFFERENT APPROACH ADOPTED IN THE IMPLEMENTATION OF BST FOR THIS MANUSCRIPT

Geometry: 1024 rays × 2048 angles			*Tomopy*: 0.762 sec		
m	τ	τ_0	τ_1	τ_2	τ_3
1	4.189	0.028	2.898	1.154	0.109
std	0.279	0.008	0.276	0.018	0.006
2	2.408	0.011	1.677	0.662	0.058
std	0.162	0.001	0.155	0.023	0.003
4	1.388	0.007	0.949	0.396	0.036
std	0.036	0.001	0.038	0.007	0.001
8	0.779	0.005	0.489	0.263	0.021
std	0.025	0.001	0.006	0.023	0.001
16	0.514	0.004	0.277	0.220	0.012
std	0.011	0.001	0.005	0.011	0.000
32	0.406	0.006	0.199	0.193	0.008
std	0.016	0.002	0.013	0.004	0.000
Geometry: 2048 rays × 4096 angles			*Tomopy*: 2.763 sec		
m	τ	τ_0	τ_1	τ_2	τ_3
1	17.345	0.220	11.535	5.108	0.482
std	0.658	0.044	0.529	0.304	0.073
2	10.318	0.144	6.971	2.957	0.246
std	0.677	0.023	0.737	0.156	0.012
4	5.875	0.114	3.854	1.745	0.163
std	0.280	0.011	0.163	0.179	0.016
8	3.449	0.102	2.125	1.123	0.099
σ	0.118	0.009	0.051	0.097	0.003
16	2.308	0.095	1.238	0.915	0.059
std	0.041	0.007	0.018	0.026	0.005
32	3.154	0.091	0.840	2.179	0.044
std	0.057	0.014	0.018	0.058	0.002

The log-polar strategy applied to a sinogram image g is obtained after 4 processing stages $\{P_0, P_1, P_2, P_3\}$, each one executed within an elapsed time τ_k, $k \in \{0, 1, 2, 3\}$.

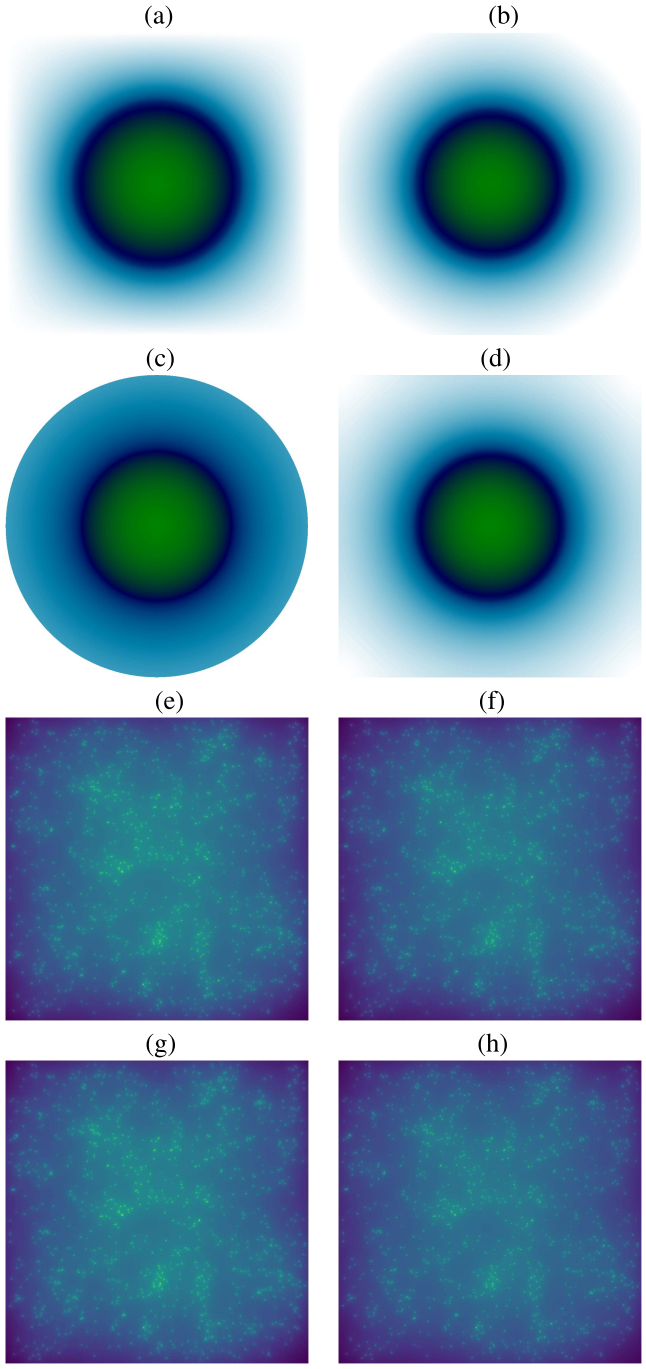

Fig. 8. Comparison of four backprojection algorithms; see text for details. (a) Analytical. (b) BST. (c) LP. (d) NFFT. (e) Analytical. (f) BST. (g) LP. (h) Ray-Based.

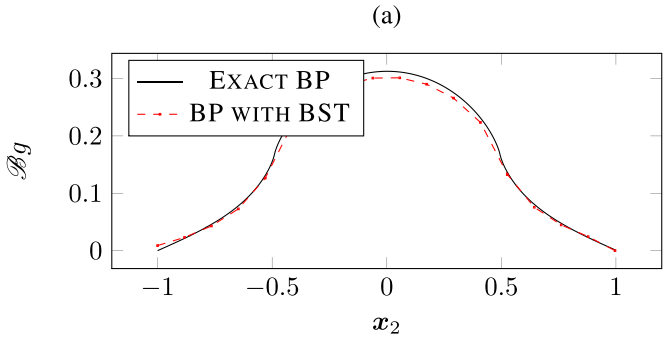

Fig. 9. Exact and numerical backprojection with BST for the circular function.

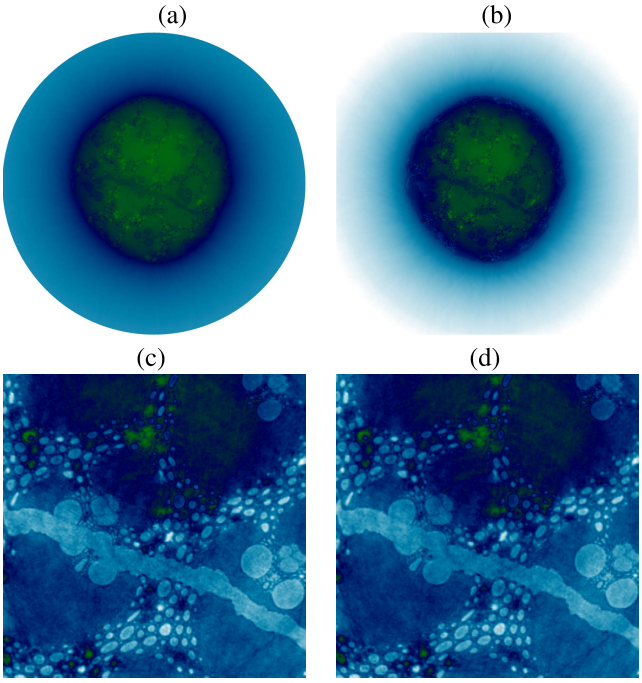

Fig. 10. Backprojection applied over noisy sinograms. Comparison between the log-polar (LP) approach and the backprojection slice theorem (BST). (a) LP. (b) BST. (c) LP near $\boldsymbol{x} = \mathbf{0}$. (d) BST near $\boldsymbol{x} = \mathbf{0}$.

The final backprojected image b is obtained as a composition of algorithms $\{P_k\}$, i.e.,

$$b = P_3 P_2 P_1 P_0(g) \tag{V.4}$$

The processing steps indicated above are similar to BST. Step P_0 indicates an interpolation to polar coordinates; step P_1 is the interpolation from polar to log-polar coordinates with zero padding; step P_2 is the 2D convolution with Andersson's kernel; step P_3 is an interpolation back to cartesian coordinates in the feature domain. Using the same ideas for timing algorithm speed, we present in Table III the execution times of Andersson's algorithm using different number of threads, also compared with a reference software like Tomopy [40]. Our implementation was based on FFT's and using the same parallel strategy of BST.

The computational complexity of Andersson's algorithm is similar to the complexity of the 2D convolution, i.e.

$$\Omega_{\mathsf{LP}} = O\left(N_\theta N_\rho (\log_2 N_\rho + \log_2 N_\theta)\right) + \Omega_{\mathsf{L}}, \tag{V.5}$$

where Ω_{L} is the summarized complexity of all log-polar interpolations (sinogram to log-polar and log-polar to cartesian). Here we assume that the Fourier transform of the kernel was pre-calculated (numerically, or analytically, like in [18]) and we do not take it into account for the final complexity. In case of adaptive ρ_0 selection a new N_ρ is selected leading to an oversampling of the log-polar sinogram. For most common sizes of the sinogram (i.e. $512 < N_s, N < 16000$) this over-

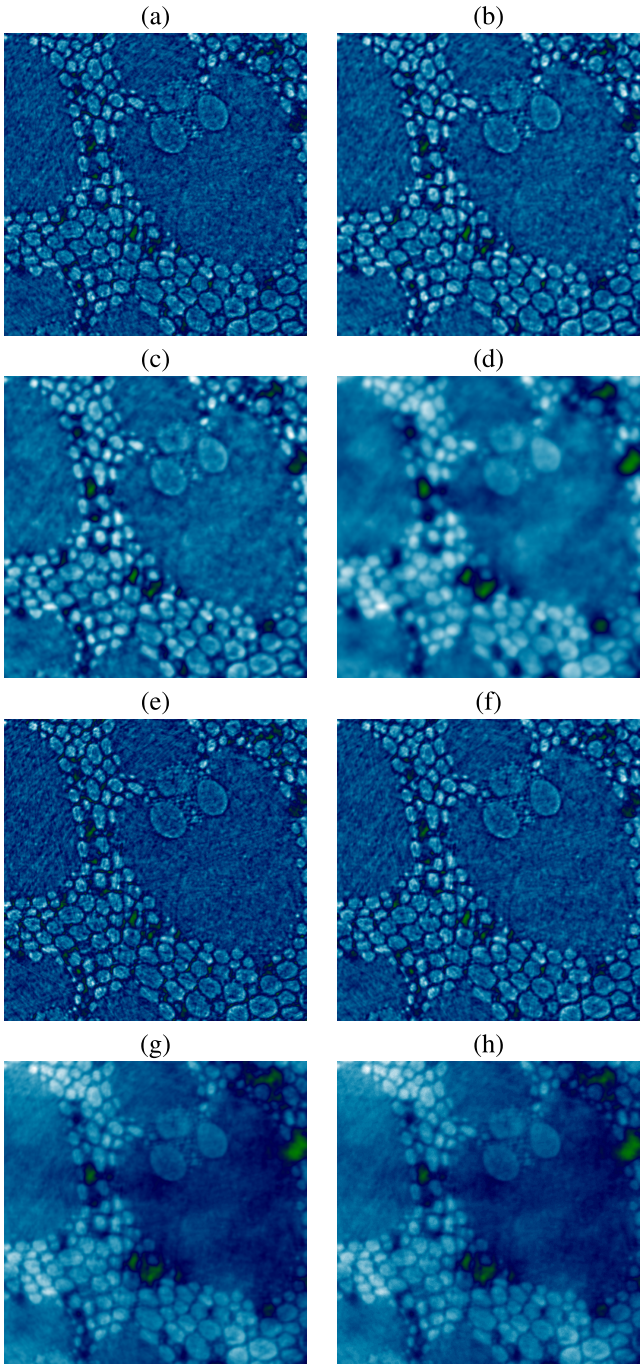

Fig. 11. (a)-(d): Regularized reconstructions using BST as a backprojection with $\lambda = 0$, $\lambda = 0.0001$, $\lambda = 0.001$ and $\lambda = 0.01$ respectively; (e)-(h): Reconstructions using different algorithms from Tomopy [40], `gridrec`, `art`, `mlem` and `pml_quad` respectively.

sampling is not very big ($N_\rho \approx 4N_s$), and the complexity of log-polar reconstruction can be estimated as $8N^2(\log_2 N + 1)$.

On Figure 8 we present the comparison of four different backprojection methods with the analytical backprojected image obtained from the sinogram of the function f. A plot comparing the BST image with the analytical backprojection at the line $\boldsymbol{x}_1 = 0$ is presented in Figure 9. A comparison of backprojected images from the wood-fiber sample, using BST and LP, is presented in Figure 10.

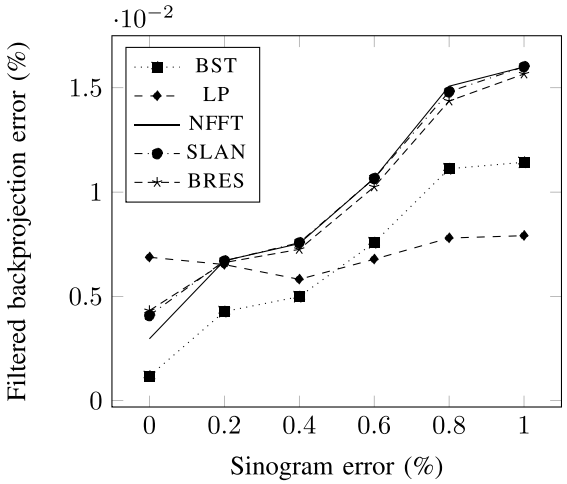

Fig. 12. Mean square error (MSE) of the result, obtained using different algorithms, in dependence on the MSE of the input sinogram.

The regularized filtered-backprojection algorithm described in Section IV was applied to the noisy data gathered for the wood-fiber. The results are shown in Figure 11 for the wood-fiber sample using only three values for the regularization parameter λ. In fact, an algorithm for the selection of the optimal parameter is beyond the scope of this manuscript. Regularized filtered-backprojected images were obtained using $\lambda \in \{0, 0.0001, 0.001, 0.01\}$. As it is known from Tikhonov regularization schemes, the bigger is λ, the smoother the resulting image will be. This is clearly visible in Figure 11, what indicates that such an approach could be used to increase the constrast in the final reconstructed image. A comparison of the regularized reconstructions with those obtained from software Tomopy [36], [40], using four different algorithms, is also presented in Figure 11. The images reconstructed by Tomopy used default filter, i.e., Shepp-Logan, for `gridrec` and one single iteration for the iterative methods `art`, `mlem` and `pml_quad`. Notice that the simultaneous methods `mlem` and `pml_quad` provide over-smoothed images in the first iteration, but the row-action method `art` is already sufficiently accurate. Nonetheless, because our intent is not to compare analytical algorithms with iterative methods, we have not tried to select the best iteration number for each iterative method.

Finally, Figure 12 presents some benchmarks of accuracy for the developed algorithm compared with the other three techniques. More precisely, we present the resulting mean squared error (MSE) *versus* the error in the input data (i.e., the sinogram). Calculations were done for the Shepp-Logan phantom with addition of Poisson noise to the analytical sinogram. From Figure 12 one can note that for weak noises BST shows the best accuracies, while Log-Polar reconstruction is very stable to strong noise.

VI. Conclusion

In this manuscript we have proposed a new backprojection technique (BST) and compared it against three other already established algorithms, the log-polar (LP) approach from

Andersson [18], with the use of Non-Uniform Fast Fourier Transforms (NFFT) from [33] and with ray-based Bresenham's [38], [41] algorithm. With the increasing size of input data, measured at imaging beamlines from synchrotron facilities, the need for fast post-processing of the data becomes an immediate demand. Either conventional reconstruction techniques like filtered backprojection or more robust iterative methods can benefit from a fast backprojection algorithm. Finally, in order to demonstrate this possibility in practice, we have provided an application where the Tikhonov image is computed by means of the BST formula in a short amount of time, thereby enabling the practical application of interesting regularization schemes to very large datasets. The BST formula is also very appealing for iterative reconstruction methods in tomography where the backprojection computation is needed per iteration.

The BST formula is very appealing for a GPU implementation because the convolution is just in one dimension. The Fourier transform applied over a block of sinograms, followed by the pointwise multiplication by kernel $1/\sigma$ and interpolation to cartesian coordinates can be made extremely fast using GPU advanced strategies. On the other hand, we have to compute the entire block of images in just one operation, which may be a limitation of BST for large datasets. In this cases, ray based methods could be attractive even though with a higher computational complexity.

Appendix A
Integral Representations

We use the following standard representation for path integrals, the proof can be found in [17]: *for a continuously differentiable* $z\colon \mathbb{R}^n \to \mathbb{R}$ *such that* $\|\nabla z\|_2 \neq \mathbf{0}$ *it is true that*:

$$\int_{\mathbb{R}^n} h(\boldsymbol{y})\delta(z(\boldsymbol{y}))\mathrm{d}\boldsymbol{y} = \int_{C^{-1}(0)} \frac{h(\boldsymbol{y}(s))}{\|\nabla z(\boldsymbol{y}(s))\|_2}\mathrm{d}s(\boldsymbol{x}) \tag{A.1}$$

where $\mathrm{d}s(\boldsymbol{x})$ is the arclength measure along curve $C^{-1}(0)$. Assuming that $g \in V$ is a given sinogram and $\boldsymbol{x}$ a pixel in the reconstruction region. The backprojection (I.4) of g is defined as the contribution of all possible straight lines, parameterized by the angle θ, and passing through $\boldsymbol{x}$. Using the sifting property of the Delta distribution, we have

$$\mathscr{B}g(\boldsymbol{x}) = \int_0^{\pi} g(\boldsymbol{x}\cdot\boldsymbol{\xi}_\theta, \theta)\mathrm{d}\theta \tag{A.2}$$

$$= \int_0^{\pi}\int_{\mathbb{R}} g(t,\theta)\delta(t - \boldsymbol{x}\cdot\boldsymbol{\xi}_\theta)\mathrm{d}t\mathrm{d}\theta \tag{A.3}$$

Switching the above integral from (t,θ) coordinates to cartesian coordinates $\boldsymbol{y} \in \mathbb{R}^2$ we have $|t|\mathrm{d}t\mathrm{d}\theta = \mathrm{d}\boldsymbol{y}$; where $\mathscr{B}g$ now becomes

$$\mathscr{B}g(\boldsymbol{x}) = \int_{\mathbb{R}^2} [g]_{\mathrm{c}}(\boldsymbol{y})\delta(z(\boldsymbol{y}))\frac{1}{\|\boldsymbol{y}\|_2}\mathrm{d}\boldsymbol{y} \tag{A.4}$$

with $[g]_{\mathrm{c}}(\boldsymbol{y}) = g(t(\boldsymbol{y}), \theta(\boldsymbol{y}))$ refering to the sinogram g in cartesian coordinates. In fact, $|t| = \|\boldsymbol{y}\|_2$ is the unsigned distance to the origin and $\theta = \arctan(\frac{y_2}{y_1}) \in [0, \pi]$ is the angle with respect to the $\boldsymbol{y}_1$-axis. Function z reads

$$z(\boldsymbol{y}) = t - \boldsymbol{x}\cdot\boldsymbol{\xi}_\theta \tag{A.5}$$

$$= \|\boldsymbol{y}\|_2 - \boldsymbol{x}_1\cos\theta(\boldsymbol{y}) - \boldsymbol{x}_2\sin\theta(\boldsymbol{y}) \tag{A.6}$$

$$= \|\boldsymbol{y}\|_2 - \boldsymbol{x}_1\frac{\boldsymbol{y}_1}{\|\boldsymbol{y}\|_2} - \boldsymbol{x}_2\frac{\boldsymbol{y}_2}{\|\boldsymbol{y}\|_2} \tag{A.7}$$

$$= \|\boldsymbol{y}\|_2 - \frac{(\boldsymbol{x}_1\boldsymbol{y}_1 + \boldsymbol{x}_2\boldsymbol{y}_2)}{\|\boldsymbol{y}\|_2} \tag{A.8}$$

$$= \frac{\boldsymbol{y}\cdot(\boldsymbol{y}-\boldsymbol{x})}{\|\boldsymbol{y}\|_2} \tag{A.9}$$

From (A.9), (A.4) and the property $\delta(au) = \frac{1}{|a|}\delta(u)$ for all $u \in \mathbb{R}$, the backprojection now follows:

$$\mathscr{B}g(\boldsymbol{x}) = \int_{\mathbb{R}^2} [g]_{\mathrm{c}}(\boldsymbol{y})\delta(\kappa_{\boldsymbol{x}}(\boldsymbol{y}))\mathrm{d}\boldsymbol{y}, \quad \kappa_{\boldsymbol{x}}(\boldsymbol{y}) = \boldsymbol{y}\cdot(\boldsymbol{y}-\boldsymbol{x}) \tag{A.10}$$

It should be noted that, for a fixed $\boldsymbol{x} \in \mathbb{R}^2$, the set $\kappa_{\boldsymbol{x}}^{-1}(0) = \{\boldsymbol{y} \in \mathbb{R}^2 : \kappa_{\boldsymbol{x}}(\boldsymbol{y}) = 0\}$ is defined as a circle in the plane. Indeed, since $\boldsymbol{y}\cdot(\boldsymbol{y}-\boldsymbol{x}) = \boldsymbol{y}\cdot\boldsymbol{y} - 2\boldsymbol{y}\cdot\left(\frac{\boldsymbol{x}}{2}\right) = \left\|\boldsymbol{y}-\frac{\boldsymbol{x}}{2}\right\|_2^2 - \left\|\frac{\boldsymbol{x}}{2}\right\|_2^2$, it follows that $\kappa_{\boldsymbol{x}}^{-1}(0)$ is a circle passing through the origin $\boldsymbol{y} = \mathbf{0}$, centered at $\frac{1}{2}\boldsymbol{x}$ and with radius $\frac{1}{2}\|\boldsymbol{x}\|_2$. Since $\kappa_{\boldsymbol{x}}^{-1}(0) = \{\frac{1}{2}\boldsymbol{x} + r\boldsymbol{\xi}_\theta : \theta \in [0, 2\pi],\ r = \frac{1}{2}\|\boldsymbol{x}\|_2\}$ is a parametric representation of the circle, the backprojection operator also reads, in an alternative form: *$\mathscr{B}$ is a stacking operator through circles* $\kappa_{\boldsymbol{x}}^{-1}(0)$*:*

$$\mathscr{B}g(\boldsymbol{x}) = \int_{\kappa_{\boldsymbol{x}}^{-1}(0)} \frac{[g]_{\mathrm{c}}(\boldsymbol{y})}{\|2\boldsymbol{y}-\boldsymbol{x}\|_2}\mathrm{d}s \tag{A.11}$$

$$= \frac{1}{2}\int_0^{2\pi} [g]_{\mathrm{c}}\left(\frac{1}{2}\boldsymbol{x} + \frac{1}{2}\|\boldsymbol{x}\|_2\boldsymbol{\xi}_\theta\right)\mathrm{d}\theta \tag{A.12}$$

The above representation follows from $\mathrm{d}s = \frac{1}{2}\|x\|_2\mathrm{d}\theta$, (A.10) and (A.1) with $\nabla\kappa_{\boldsymbol{x}}(\boldsymbol{y}) = 2\boldsymbol{y} - \boldsymbol{x}$. Last equality comes from $\boldsymbol{y} = \frac{1}{2}\boldsymbol{x} + \frac{1}{2}\|\boldsymbol{x}\|_2\xi_\theta \in \kappa_{\boldsymbol{x}}^{-1}(0)$ for some θ. Therefore, in cartesian coordinates, the backprojection contribution for a ball $\{\boldsymbol{u} \in \mathbb{R}^2 : \|\boldsymbol{u}-\boldsymbol{x}\|_2 \leq \epsilon\}$ comes from a family of circles passing through the ball and the origin, this fact is related to the *comet-tail region* mentioned in [42].

Acknowledgment

The authors would like to thank the reviewers for many valuable suggestions. The authors would also like thank to Gilberto Martinez Jr. for helping with the implementation of BST in C/C++ programming language.

References

[1] R. N. Bracewell, "Strip integration in radio astronomy," *Austral. J. Phys.*, vol. 9, no. 2, pp. 198–217, 1956.

[2] F. Arcadu, M. Nilchian, A. Studer, M. Stampanoni, and F. Marone, "A forward regridding method with minimal oversampling for accurate and efficient iterative tomographic algorithms," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1207–1218, Mar. 2016.

[3] F. Marone and M. Stampanoni, "Regridding reconstruction algorithm for real-time tomographic imaging," *J. Synchrotron Radiat.*, vol. 19, no. 6, pp. 1029–1037, 2012.

[4] J. Waldén, "Analysis of the direct Fourier method for computer tomography," *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 211–222, Mar. 2000.

[5] A. Voropaev, A. Myagotin, L. Helfen, and T. Baumbach, "Direct Fourier inversion reconstruction algorithm for computed laminography," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2368–2378, May 2016.

[6] A. Myagotin, A. Voropaev, L. Helfen, D. Hänschke, and T. Baumbach, "Efficient volume reconstruction for parallel-beam computed laminography by filtered backprojection on multi-core clusters," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5348–5361, Dec. 2013.

[7] R. Gordon, G. T. Herman, and S. A. Johnson, "Image reconstruction from projections," *Sci. Amer.*, vol. 233, no. 4, pp. 56–71, 1975.

[8] F. Natterer, *The Mathematics of Computerized Tomography*, vol. 32. Philadelphia, PA, USA: SIAM, 1986.

[9] E. X. Miqueles and A. R. De Pierro, "On the iterative inversion of generalized attenuated radon transforms," *J. Inverse Ill-Posed Problems*, vol. 21, no. 5, pp. 695–712, 2013.

[10] L. A. Kunyansky, "Generalized and attenuated radon transforms: Restorative approach to the numerical inversion," *Inverse Problems*, vol. 8, no. 5, p. 809, 1992.

[11] G. Rigaud and A. Lakhal, "Approximate inverse and Sobolev estimates for the attenuated radon transform," *Inverse Problems*, vol. 31, no. 10, p. 105010, 2015.

[12] F. Natterer, "Inverting the attenuated vectorial radon transform," *J. Inverse Ill-Posed Problems*, vol. 13, no. 1, pp. 93–101, 2005.

[13] O. Tretiak and C. Metz, "The exponential radon transform," *SIAM J. Appl. Math.*, vol. 39, no. 2, pp. 341–354, 1980.

[14] E. X. Miqueles and A. R. De Pierro, "Exact analytic reconstruction in X-ray fluorescence ct and approximated versions," *Phys. Med. Biol.*, vol. 55, no. 4, p. 1007, 2010.

[15] Y. Wei, G. Wang, and J. Hsieh, "Relation between the filtered backprojection algorithm and the backprojection algorithm in CT," *IEEE Signal Process. Lett.*, vol. 12, no. 9, pp. 633–636, Sep. 2005.

[16] L. Feldkamp, L. Davis, and J. Kress, "Practical cone-beam algorithm," *J. Opt. Soc. Amer. A*, vol. 1, no. 6, pp. 612–619, 1984.

[17] S. Helgason, *The Radon Transform on* $\mathbb{R}^n$. New York, NY, USA: Springer, 2011.

[18] F. Andersson, "Fast inversion of the radon transform using log-polar coordinates and partial back-projections," *SIAM J. Appl. Math.*, vol. 65, no. 3, pp. 818–837, 2005.

[19] A. Brandt, J. Mann, M. Brodski, and M. Galun, "A fast and accurate multilevel inversion of the radon transform," *SIAM J. Appl. Math.*, vol. 60, no. 2, pp. 437–462, 2000.

[20] A. George and Y. Bresler, "Fast Tomographic Reconstruction via Rotation-Based Hierarchical Backprojection," *SIAM J. Appl. Math.*, vol. 68, no. 2, pp. 574–597, 2007.

[21] S. Basu and Y. Bresler, "O(N(2)log(2)N) filtered backprojection reconstruction algorithm for tomography," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1760–1773, Oct. 2000.

[22] E. X. Miqueles and E. S. Helou, "Fast backprojection operator for synchrotron tomographic data," in *Proc. Eur. Conf. Math. Ind.*, 2014, pp. 243–252.

[23] D. Potts and G. Steidl, "New Fourier reconstruction algorithms for computerized tomography," in *Proc. Int. Symp. Opt. Sci. Technol.*, 2000, pp. 13–23.

[24] K. Fourmont, "Non-equispaced fast Fourier transforms with applications to tomography," *J. Fourier Anal. Appl.*, vol. 9, no. 5, pp. 431–450, 2003.

[25] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. Piscataway, NJ, USA: IEEE Press, 1988.

[26] D. Potts and G. Steidl, "A new linogram algorithm for computerized tomography," *IMA J. Numer. Anal.*, vol. 21, no. 3, pp. 769–782, 2001.

[27] P. R. Edholm, *The Linogram Algorithm and Direct Fourier Method With Linograms*. Linköping, Sweden: Linköping Univ. Electron. Press, 1991.

[28] S. R. Deans, *The Radon Transform and Some of Its Applications*. North Chelmsford, MA, USA: Courier Corp., 2007.

[29] F. Wubbeling and F. Natterer, *Mathematical Methods in Image Reconstruction*, vol. 8. Philadelphia, PA, USA: SIAM, 2001, p. 16.

[30] H. Prüfer, "Neue Herleitung Der Sturm-Liouvilleschen Reihenentwicklung Stetiger Funktionen," *Math. Ann.*, vol. 95, no. 1, pp. 499–518, 1926.

[31] L. A. Shepp and B. F. Logan, "The Fourier reconstruction of a head section," *IEEE Trans. Nucl. Sci.*, vol. NC-21, no. 3, pp. 21–43, Jun. 1974.

[32] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, and J. Walden, "Fast slant stack: A notion of radon transform for data in a Cartesian grid which is rapidly computible, algebraically exact, geometrically faithful and invertible," *SIAM Sci. Comput.*, 2001.

[33] S. Kunis and D. Potts, "Time and memory requirements of the nonequispaced FFT," *Sampling Theory Signal Image Procoss.*, vol. 7, no. 1, pp. 77–100, Jan. 2008.

[34] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, 1965.

[35] D. G. Luenberger, *Optimization by Vector Space Methods*. Hoboken, NJ, USA: Wiley, 1997.

[36] D. M. Pelt *et al.*, "Integration of tomopy and the astra toolbox for advanced processing and reconstruction of tomographic synchrotron data," *J. Synchrotron Radiat.*, vol. 23, no. 3, pp. 842–849, 2016.

[37] M. Frigo and S. G. Johnson, "The design and implementation of FFTW3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.

[38] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, Mar. 1965.

[39] J. Kaiser and R. Schafer, "On the use of the I0-sinh window for spectrum analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 1, pp. 105–107, Feb. 1980.

[40] D. Gürsoy, F. De Carlo, X. Xiao, and C. Jacobsen, "TomoPy: A framework for the analysis of synchrotron tomographic data," *J. Synchrotron Radiat.*, vol. 21, no. 5, pp. 1188–1193, 2014.

[41] H. Zhao and A. J. Reader, "Fast ray-tracing technique to calculate line integral paths in voxel arrays," in *Proc. Nucl. Sci. Symp. Conf. Rec.*, 2003, pp. 2808–2812.

[42] R. Hass and A. Faridani, "Regions of backprojection and comet tail artifacts for π-line reconstruction formulas in tomography," *SIAM J. Imag. Sci.*, vol. 5, no. 4, pp. 1159–1184, 2012.